MATH 3320-Student Projects and Feedback

Eddy Kwessi

Department of Mathematics

December 15, 2016.

Contents

MATH 3320

Projects Guidelines

▶ The goal of these projects is for students to demonstrate their ability to use appropriate probability and statistical tools and techniques to solve engineering problems in group projects

The following is the outline that will be used for this assignment. You will participate, present your research, and turn in your accompanying report for full credit in this course. You will be paired up with at most two course mates for your project. Your will earn an individual maximum grade of 200 divided into 100 for your report and 100 for your presentation

1. Project Guidelines

1.1 Structure of your report

▶ Your report should be typeset, no more than 10 pages long. Failure to do so will result in grade of 0 for the entire report.
▶ You can use any software you want for your simulations/calculations: Matlab, Mathematica, Maple, Python, Java, R, Minitab, etc. Just make sure to mention which one was used in your report.
▶ Your report should have the following structure to be acceptable. No departure from this format will be accepted.

1. Introduction (15 pts) : This is the fist section of your report. Briefly describe your motivation for choosing the project and provide the significance of the problem based on a literature review and/or your best knowledge.

2. Statement of the problem (25 pts): This is the second section of your report.

(a) Briefly describe the problem, or the specific research question to be answered or the hypothesis to be tested, etc.

(b) Clearly state what you want to achieve.

3. Methodology (25pts): This is the third section of your report.

   (a) Clearly demonstrate the connection of this project to Probability and Statistics.

   (b) List the procedures. (A detailed description of what you did and step-by step)

   (c) Describe data processing and analysis. (How will you analyze the data and why?)

      i. If you don't have data, make it up or ask the instructor.

      ii. Are the tables and graphs appropriately used?

4. Results (25 pts): This is the fourth section of your report. Report the results you obtained.

   (a) Explain your results clearly and concisely.

   (b) Discuss the limitations and challenges of the methodology you used.

   (c) Briefly discuss what you think can be done to improve your results.

5. Bibliography (10 pts): This is fifth and final section of your report. You should list in alphabetic order all the references you used for your project. use any acceptable scientific format for articles, books, websites etc.

1.2 Project's Presentation

▶ Each project will have to be presented.
▶ The presentation should be prepared in a CLT room, videotaped, and posted on the project's forum on TLearn by the date suggested on Benchmark 4 below.

▶ **All group members** should provide feedback on at least one other suggested group and failure to do so will result in a lost of 20 points.
▶ Below are the necessary items for an acceptable presentation.

1. Be properly attired (5 pts)

2. Comments on other presentations (15 pts)

3. Each group member should comment on another suggested project (20 pts)

4. Have a good flow (chemistry) among the group members (20 pts)

5. Submit your presentation on time and it should be about your actual project (20 pts)

6. Pass all the 4 Milestone (20 pts)

1.3 Project's Benchmarks

▶ Each project will need to meet at least two of four benchmarks to be accepted. Less than two benchmarks achieved by the end of the semester will result in a grade of 0 on the entire project!

▶ Each progress report must be submitted on TLearn by the date mentioned below with the subject: **MATH 3320-F16-Group number: Milestone i**, where **i** could be 1, 2, 3 or 4.

Progress report	**Description**	**Date**
Milestone 1	Choosing your project	September 13
Milestone 2	Layout of your research goals and objectives	September 29
Milestone 3	Implementation of your goals	November 3
Milestone 4	Final Presentation and Report Submission	December 1

2. Projects

▶ You can choose a project under the co-supervision of a Faculty (See the list below. Moreover, you are highly encouraged to discuss this project with the faculty) or choose your

project based on the topics provided below.

► Do not choose a project outside of these topics.

► Should you find the need to consult with me on a project, please do so during office hours or by appointment only.

2.1 Faculty proposed projects

1. Radio Station Design: Design a radio station to broadcast a signal with a certain probability distribution. This requires finding the information content of the signal and design the station accordingly. (**Dr. Aminian**)

2. Hard Drive Design Design a hard drive to store data with certain probability distribution (Gaussian, random, etc.) at a certain rate (Mb/s). In this project, students need to design an appropriate A-to-D converter using the signal distribution and find the channel capacity for the connection to the motherboard, e.g. SATA. (**Dr. Aminian**)

3. Simulation of electron motion in semiconductors (Monte Carlo simulation): This project considers motion of an electron in a semiconductor under influence of an electric field. The electron goes through free flight for a time which can be selected from a normal distribution. During the motion, the final velocity of the electron when free flight is over would be v2=v1+(e*E/m)*t where t is the time of flight, E is the applied electric field and e and m are electron's charge and mass. After free flight is collision with an atom (knows as electron-phonon collision). during this process, electron can absorb a phone or eject a phone thus gaining or losing energy. There could be several types of collisions which can be randomly selected from a pool of 2 to 4. A simulation lasting 1M free flights can give an indication of what is happening to the electron in steady state. (**Dr. Aminian**)

4. Monte Carlo Simulation Generating random numbers from a distribution function which cannot be integrated analytically and apply it to electron motion in semiconductors (**Dr. Aminian**)

5. Robot Localization 1: Very similar to (3), except that sensor readings are used to re-weight each particle and the ones with greater weight are more probable to be selected in the next iteration. Known as a "particle filter" in robotics. (**Dr. Nickels**)

6. Robot Localization 2: The idea is to make a geo-referenced library of image (features) then dynamically take and image and compute offsets to the nearest N images (probably 2-4), and use this to compute the location from which the new image was taken. I could see a probabilistic or a statistical formulation of this problem (either find a PDF of the robot location, or find the likelihood of a false match). (**Dr. Nickels**)

7. Mapping: The most probabilistic formulation would be to give/develop a probabilistic model of a sensor and have a grid of locations, each element of the grid containing p(o), the probability that this grid cell is occupied by an obstacle. Then, given a set of sensor readings and robot locations, each grid cell is updated. (**Dr. Nickels**)

8. Kalman Filtering: In this formulation, a robot's location is modeled as a gaussian. A (set of) sensor readings is also modeled as a gaussian. The readings are then used to update the robot's location. If the uncertainty profile of the sensor readings is constant, this devolves to a Weiner Filter. I've also seen this used in radar-style tracking, and in partially observable robot arms. (**Dr. Nickels**)

9. Odometry Motion Model: Making some (probabilistic) assumptions about wheel slip, motor control, etc. model the motion of the robot in the world as an evolving distribution function. (**Dr. Nickels**)

2.2 Projects Topics

1. Information Theory

2. Mathematics of Shuffling

3. Probability in Stock Markets

    - Is Warren Buffet an investment genius or incredibility lucky?
    - Does the stock market remember yesterday?

4. Spam Filters

5. Visualizing Probability

6. Gambling

7. Probability and Sports

    - Diving/flopping culture in sports.
    - Is women tennis unpredictable?
    - Predicting a team's winning percentage in the NBA/NFL.

8. Digital Image or Speech processing

9. Probability and Weather Forecast

10. Least Square Modeling in Digital Communication

11. Data Analysis of Electric Cars Versus Gas Cars

12. Data analysis for prediction of presidential election.

13. Modeling weather effects on road casualty statistics in the U.S.

MATH 3320

Group 1: Dakota Grusak, Chris Nkinthorn, Gregory Wassom

Project's Report Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Introduction	15	15	Well written.
Statement of the Problem	25	20	That was missing and because it was embedded in the introduction.
Methodology	25	25	Well explained. You cover all the key points in your statistical analysis.
Results	25	25	Well explained as well.
Bibliography	10	10	Well done.
Other comments			I like the depth of your analysis. I like your findings.
Total	100	**95**	You made a comment about the two distributions being reconcilable. If you overlay the two graphs, you will realize that the theoretical model covers more possibilities, including the Billion + simulations that you did.

Project's Presentation Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Be properly attired	5	5	Well done
Comments of another presentation	35	35	Well done
Chemistry among group members	20	15	Not too bad but not well coordinated
Timely submission of the presentation	20	10	You did not submit the report on time
Pass all 4 Milestones	20	20	Well done
Other comments			Your video editing was not well done
Total	100	**85**	

MATH 3320: Probability and Statistics for Scientists and Engineers

Probabilistic Model for the Location of a Two-Wheeled Robot

By Dakota Grusak, Chris Nkinthorn, and Gregory Wassom
Group 1
December 1, 2016
Prepared for Dr. Eddy Kwessi

Introduction:

Predicting and controlling the actions of a robot is an important aspect of robotics. However, controlling a robot is difficult because the definite actions of a robot are affected by a world that is inherently *uncertain*. This uncertainty can come from a variety of error sources: environmental, computational, sensor, etc. This uncertainty can be accounted for through the use of mathematical models that probabilistically predict where the robot will be. However, the purpose of this experiment is to evaluate one and only one source in particular: error of missing an encoder count.

An encoder is a mechanism often found in a wheeled robot that is used to record how much a wheel has rotated. When the wheel has moved through an associated arc distance the encoder count value increases. If encoder count associated with one complete rotation of the wheel is 1024, then when the wheel is rotated 1/1024th of a complete rotation when the encoder count value increases by one. However, like any mechanism, the encoder is subject to error. When the robot moves, there is chance that the robot misses a change in arc distance.

This probability can be modeled using a Bernoulli because missing the changing the value of of an encoder is a binary process: either it correctly reads the changes or it does not. There is a certain probability with the changing being successful or not and each trial is independent. If an array of encoder values is created that describes the motion of a robot following an arbitrary path, then the Bernoulli can be applied to the true list of encoder values. By applying the Bernoulli to the true list of encoder values repeatedly, a distribution of final locations can be created. These final locations are purely the result of encoder error and no other source of error because the original list describes the path perfectly.

Methods:

Basis of Simulation

We define the following. First, the location of the robot is based on the center of the wheelbase. The motion of each wheel is captured on an encoder. Each encoder relates to the angular displacement of each wheel. As the radius of each wheel is known and that only one wheel moves at a time, then the new position of the moved wheel can be derived using an equation for a circle.

With the new wheel locations known, the new robot position can be derived and a displacement calculated. Noting which wheel moves is captured on an encoder associated with each wheel. On an actual robot, the encoder count would continuously increase but for the purposes of this simulation, we will denote changes in encoder values with a 1 or a 0. Ones represent an input into the wheel and a zero represents no change in the robot's wheel.

However, the encoder has some error: it may or may not pick up a change in input. The likelihood that the robot will pick up the change in input is either true or false and can found experimentally. This means that the probability in capturing a change in the encoder value can be modeled using a Bernoulli distribution.

By modeling the motion of the robot and calculating the values of encoder with each increment in distance, we can recreate the motion of the robot as it travels along the path. We can also treat the changes in input of encoder values with a Bernoulli distribution and calculate the possible paths taken by the robot. These final locations can be plot with relation to the set final location and a distribution of locations can be created.

Parameters of the Simulation.

1. The chance of missing a change of encoder is independent of velocity. Though practically, the likelihood missing the change of the encoder value would be dependent on the velocity, this assumption is justifiable if it moved at a relatively constant speed.
2. There is no slipping between each wheel and the ground so that we can analyze purely the effect of error in the encoder.
3. Each wheel has the same encoder so that the likelihood of missing the change in encoder count is the same for each wheel.
4. The robot is facing in the direction of the path and only moves forward.
5. The path must have a minimum radius of curvature along the bounds of the curve greater than the wheelbase of the robot.
6. The path of the differentiable and the second derivative of the slope must be nonzero.

Implementation and MATLAB code steps

User Input

The user defines the parameters of the model, and then models the motion of the robot. The following have to be defined beforehand and remain within the parameters of the model.

- User defines a radius of the wheel, r, the wheelbase distance, l, and a encoder count value per revolution, and an initial direction.
- User will define an input path as a parametric equation within the xy-plane in terms of t where $t \in [a, b]$.

Initialize Program:

To initialize the program, the user is prompted to define parameters of the robot including the number of counts per full revolution, radius of each wheel and distance between wheels. Then, a parametric function in terms of 't' within the bounds, a and b, must be inputted to provide a path for the robot to follow. The initial direction is also input into the script. From these values, the absolute position of the robot is derived.

Using the initialized values, we can derive some important values. First is the unit arc distance per change of encoder value. Given the assumption that the robot traces out the arc of a circle, we can define the arc distance by dividing by the circumference.

Then, we define the wheel locations from the first position of the parametric path. For the wheels, we use sine and cosine functions to define the x and y positions from the initial position and the initial direction. The left and right wheels are 180 degrees out of phase with each other. Note that the robot position is calculated as the mean of the x and y position of the left and right wheels.

- Define the initial robot wheel locations.

a. whe_lx = posx(0) + l/2*cos(pi/2 + init_direc)
b. whe_rx = posx(0) + l/2*cos(3*pi/2 + init_direc)
c. whe_ly = posy(0) + l/2*sin(pi/2 + init_direc)
d. whe_ry = posy(0) + l/2*sin(3*pi/2 + init_direc)

- Initial position of Robot is then calculated.
  a. ropo_x = (whe_lx + whe_rx)/2
  b. ropo_y = (whe_ly + whe_ry)/2;

Path Calculation

To assess the expected versus actual position of the robot at a given value of **t**, the length of the defined path must be calculated. Assuming a non-linear path within [a,b], we can calculate the arc length of the path so that we can reference the distance the the robot has traveled against it. We first calculate the arc length of the curve, which can be defined as the arc within the bounds **a, b** of the parametric function. We do this by adding the parametric equations in quadrature, before taking a line integral.

- f = sqrt(diff(x)^2 + diff(y)^2)
- L = int(f,t,a,b)

Another parameter that we want to find is the slope of the path as a function of t by dividing the derivative of y by the derivative of x. This is done using a MATLAB function so that it is general to any function.

Robot Following Path

The position of the robot will be defined using the two wheel locations. Here we use the absolute position of the robot in the x and y directions (ropo_x and ropo_y) and the following parameters were derived in vector form:

- ropo = [ropo_x ropo_y 0]
- roor = [(whe_rx - whe_lx) (whe_ry-whe_ly) 0]

Although parameters to define the location of the robot have been defined, they are not yet in terms of **t** and as a result, cannot be plotted using the input parametric paths. To find this **t** value, we have an expression for the displacement between the expected and actual path of the robot. This is done by minimizing the distance between the parametric equations and the position of the robot which allows us to solve for the given time which that path would occur.

- dist_eqn = (x - ropo_x)*dx + (y - ropo_y)*dy
- ts = abs(solve(dist_eqn ,t))

Once we have the desired t value, we can then calculate the angle between our desired position and actual position which gives knowledge of how the robot should make its next move. The robot follows the path based on the sign of the angle found in the previous step. Once the robot moves, we then compute a binary equivalent of its motion based on wheel count. If the robot moved the left wheel and the right wheel was stationary, then the encoder array would have the values [1 0] added to its length. If the reverse was true and it was the right wheel that moved and the left wheel stationary, then [0 1] would have been added.

From the count of the distance traveled by wheel rotation, we then calculate the new position of the robot, and compute the distance traveled in that loop iteration. This process repeats until the accumulated distance is equal to that of the true path. With the loop's termination, an array of encoder count values is created. This array of encoder counts describes the motion of a robot purely in encoder values.

The robot position vector is then updated based on the new position before going through the next loop iteration. To then get a visual representation of the position of the robot as a function of the path, we plotted the locations of the robot with each change in encoder value along with the intended path.

Adding a Bernoulli to Encoder Values

A Bernoulli trial is a situation with two outcomes: "success" or "failure". Associated with these outcomes is a probability that either event will occur in a given trial. Take for example, a fair die. If the die is randomly rolled, the probability that it will come up a 6 is ⅙. We use this idea to simulate error in the encoder. Any time there is a change in count in the encoder from one step to the next, there is a probability that the count will be missed. The probability that a count will be missed can be set based on the parameters of a given robot. We use a probability of 1/10 for our simulation meaning that a count will be missed 10% of the time. To implement this in our script, we split the encoder matrix into two vectors for "left wheel" and "right wheel". For each one of these vectors, we calculated the difference in count from one position to the next. The code works such that, if the count is different from one position to the next, there is a probability that the change in count will be unseen by the encoder. In this case, we check the Bernoulli to see whether it is true (change in count), or false (missed count) in each case. If the Bernoulli is true, the change is observed by the encoder, and if it is false, the change is not observed. We did this for many loop iterations to generate "experimental" encoder data.

Converting Bernoulli Encoders to Position

After the compilation of experimental encoder positions, we needed to convert the binary encoder values back into position to see how they deviated from the expected path. We did this by sending the encoder values through a process that is similar, but opposite to the process used to change position to encoder values above. Once we had the position vectors, we plotted the final positions of all Bernoulli trials to see the distribution of our data.

Results

We find that for a semicircular path followed by a two wheeled robot, the distribution of locations due to error with an encoder does not match the distribution that we used as reference. Note that in the figure following, from Thrun's "Probabilistic Robotics" for a velocity motion model, that the shape of the probable locations are dissimilar. Though they have a similar shape on the robot's right side or outside the arc of the circle, with the location distribution arcing downwards, they are not the same on the left. Nor are they similar in the region around the true final location. The encoder error distribution congregates around that final location whereas the reference shows a cloud.

Discussion

Comparing the location distributions, we find that the error due to an encoder is unlike the referenced motion model. First, we must discuss the limitations of our model. Namely, in the processes of calculating the final locations due to encoder error, the final position is determined by the total distance traveled: wherever the robot was when it reached the same distance L, the path length, became the recorded final position. This may explain the clustering around the true final location: to be just ahead or just behind the true final location would require a difference in the total distance traveled.

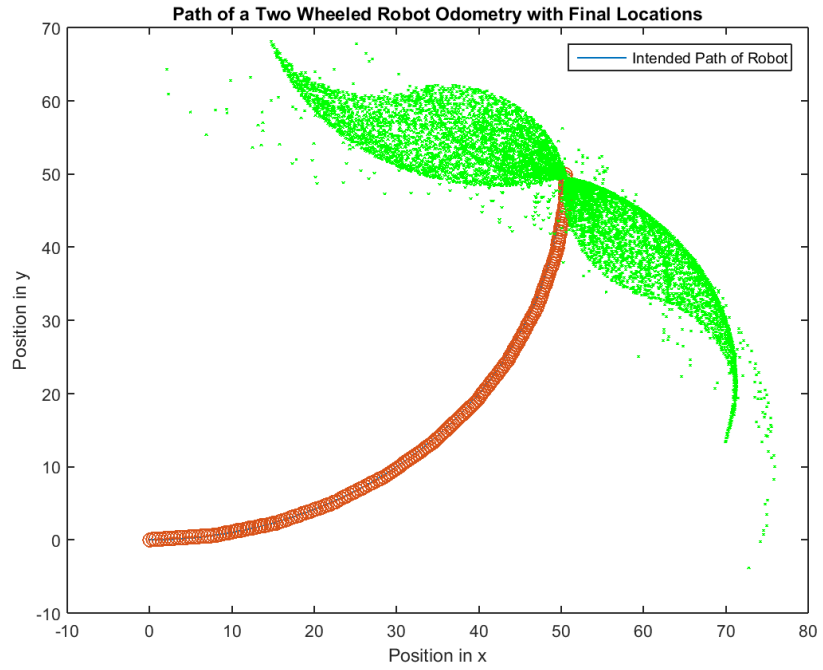


Figure 1: Location distribution of a robot tracing out the arc of a circle based on encoder error.

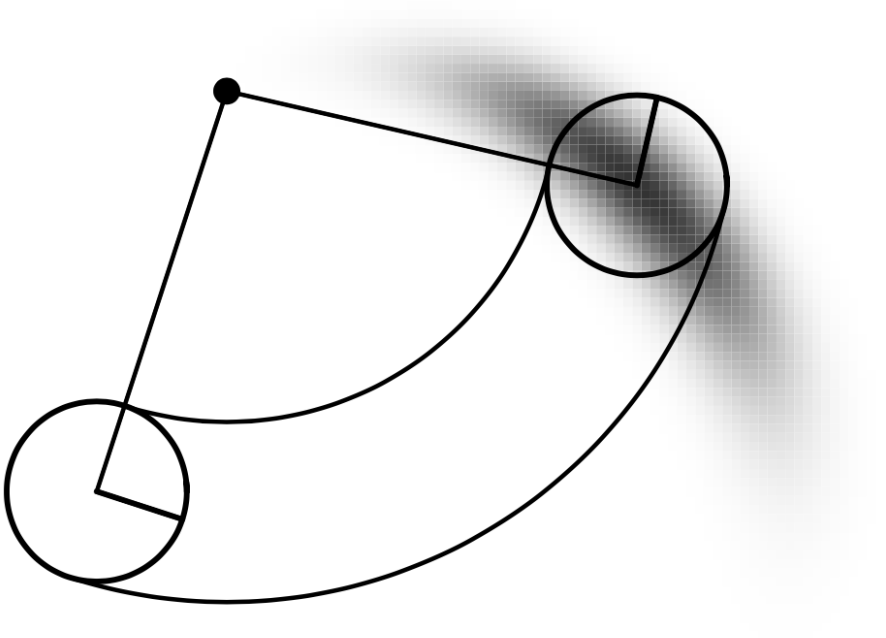


Figure 2: Velocity Motion Model for a robot tracing out an arc of a circle.

Note that another conclusion that can be pulled for the resulting graphs is that for encoder error, that the final location distribution is highly clustered with distinct regions that the robot will be in. Outside of those regions, it is much less likely for the robot to be there. These regions can be seen in the figure above. Though the two regions, shaped like palm fronds, are where the vast majority of the 100 billion final locations are found, there are some that stand out. Note that for the right side that, the location distribution seems to taper to a point, whereas on the left, the final locations seem to fan out. In a similar manner, the right fan seems to be smaller than the the one on the left.

It should be noted that the parameters for the robot were an encoder count per revolution of 4, a wheel radius of 3, and a wheelbase of 5. The likelihood of a success in the Bernoulli or missing an encoder count value of 0.001. These parameters were chosen so that the scope of the location distribution would be similar to that found in Thrun's as well as the decrease computation time. A higher encoder count, or a larger wheelbase would have meant that for each iteration, that the distance traveled would have diminished.

Though the two distributions have different shapes, it is not that the two models cannot be reconciled. The motion model found in Thrun is one for the general error of the motion of a robot whereas the one presented in this report is for purely encoder error. More than likely, the error due to a missed encoder count would be accounted for in the velocity motion model. Note that the probability of a success, a missed

encoder count, is actually far lower than what was used to exaggerate the effect. Encoders are very unlikely to fail. Given these aspects of these two models, it is totally possible that they are both valid.

Conclusions

In conclusion, we were able to generate a MATLAB script which simulated both perfect and experimental motion of a two-wheeled robot in two dimensions. This script is flexible in that it can use an arbitrary parametric plot and calculate the motion of a robot based on changing count values. We used this script for a path that traced out an arc of a circle and allowed it to plot the final locations if there were encoder error using a Bernoulli distribution where a success was missing an encoder count value. This Bernoulli application was repeated for 100 billion times so that a distribution of locations based on encoder error could be generated.

We then compared the final location distribution based on encoder error to the motion model presented by Thrun as reference. We find that the two distributions are dissimilar and that encoder error is not representative of total error. This finding can be used to show that error due to missed encoder affects the location of a two wheeled robot differently than what would have been expected of error due to other situational error sources.

Citations

Thrun, Sebastian et al. *Probabilistic Robotics*. 1st ed., Cambridge, Mass., MIT Press, 2005.

```
%% Stats Project: Odometric Model Code

% MATH 3320 - Probability and Statistics for Engineers and Scientists
% Group: Dakota Grusak, Chris Nkinthorn, Gregory Wassom
% Instructor: Eddy Kwessi, Ph.D.


%% Initialize Program

% Clear Workspace and Window
clc
clear

% Robot Parameters
n = 4;      % Count Value for Full Revolution
r = 3;      % Wheel Radius
l = 5;      % Wheelbase

% Path Parameters
syms t
x = 50*sin(t);                      % Function for x-position
y = -(50*cos(t) - 50) ;             % Function for y-position
posx = matlabFunction(x);           % Convert x Symbolic to Anonymous
Function
posy = matlabFunction(y);           % Convert y Symbolic to Anonymous
Function
dx = diff(x);                       % Function for x-derivative
dy = diff(y);                       % Function for x-derivative
delt = 10^(-5);                     % Differential step in t

% Boundaries of Path
a = 0;     % Lower Bound of t
b = pi/2;     % Bpper Bound of t

% Length of Unit Increase of Wheel
d = (2*pi*r)/(n);               % Distance Increase
arctheta = atan(d/(2*pi*l));    % Arc with Distance Increase

% Initial Direction
init_direc = 0;

% Robot Wheel Location (Initial)
whe_lx = posx(0) + l/2*cos(pi/2 + init_direc);      % Left Wheel Position
whe_ly = posy(0) + l/2*sin(pi/2 + init_direc);

whe_rx = posx(0) + l/2*cos(3*pi/2 + init_direc);    % Right Wheel Position
whe_ry = posy(0) + l/2*sin(3*pi/2 + init_direc);

% Robot Position (Initial)
ropo_x = (whe_lx + whe_rx)/2;   % Robot Position in the x
ropo_y = (whe_ly + whe_ry)/2;   % Robot Position in the y

% Initialize Values
dist = 0;
```

```
position_vec = [ropo_x ropo_y];
countl = 0;
countr = 0;
count = [countl countr];
rotheta = -pi/2;

% Probablility Parameters
pr = 0.001;                          % Probability of Missing an Encoder Count
iter = 500; iter = 1:iter;          % Iterations of Bernoulli Applications
val = 10^3;


%% Path Calculation

% Explanation:
% This portion of uses the parametric path input and calculates the length
% of the path the robot will follow. This is used to end the simulation
% when the robot has passed reached the end of the path.

% Length of Parametric Function
f = sqrt(diff(x)^2 + diff(y)^2);
L = int(f,t,a,b);
Length = vpa(L,6);
Length = double(Length);
L = Length;
clearvars Length;

% Path Slope Functions
diffx = diff(x);
diffy = diff(y);
diffdiffx = diff(diff(x));
diffdiffy = diff(diff(y));

% Position Function
pos(t) = [x y 0];
pos = matlabFunction(pos);


%% Minimum Radius of Curvaure Condition

% Explanation:
% This portion of code calculates the smallest radius of curvature
% associated with the path calculation and compares it to the length of the
% wheelbase. If the minimum radius of curvature is less than the wheelbase
% the code ends. This is because neither wheel can go backwards; the
% center of the robot would not follow that path.

rhocurve = ((diffx^2 + diffy^2)^(3/2))/(abs(diffx*diffdiffy -
diffy*diffdiffx));
rhocurve = matlabFunction(rhocurve);
tr = fminbnd(rhocurve, a, b);
minrhocurve = rhocurve(tr);

if minrhocurve < l
```

```
    fprintf('Path cannot be processed. Wheelbase is larger than the minimum
radius of curvature of the path.\n');
    fprintf('The path must be changed for reasonable count array values.\n');
    return
end


%% Robot Following Path

% Explanation:
% This is the body of the code. This loop calculates the distance the
% robot as traveled. If the total length the obot has travled. If it less
% than the calculated path length, then the loop continues. The script
% calcuates the position of the robot and then estimates the value 't'
% that describles the point on the path, nearest to the robot. This value
% 't' is then used to descrive the slope of the path. This slope is compared
% to the orientation of the robot. From this, one of the wheel positions is
% changed. The distance traveled is calculated and the loop repeats.

% Path Loop Calculation

%while dist < L                     % One of these loops is used for        %
comment out
                                    % computation or testing.               %
comment out
for i = 1:10

    % Robot Pose Parameters
    ropo_x = (whe_lx + whe_rx)/2;   % Robot Position in the x
    ropo_y = (whe_ly + whe_ry)/2;   % Robot Position in the y
    ropo = [ropo_x ropo_y 0];       % Robot Position Vector
    roor = [(whe_rx - whe_lx) (whe_ry-whe_ly) 0]; % Vector from left to right
wheel

    % Solving for 't' Value Nearest
    dist_eqn = (x - ropo_x)*dx + (y - ropo_y)*dy;
    ts = abs(solve(dist_eqn ,t));
    ts_temp = vpa(ts,6);
    ts_temp = double(ts_temp);
    ts = ts_temp;
    clearvars ts_temp;

    % Loopback Error
    pos_error = [(ropo_x - posx(ts)) (ropo_y - posy(ts))];
    if norm(pos_error) > l/4
        pos_error_l = [(whe_lx- posx(ts)) (whe_ly-posy(ts))];
        pos_error_r = [(whe_rx- posx(ts)) (whe_ry-posy(ts))];
        if norm(pos_error_l) < norm(pos_error_r)
            whe_rx = whe_lx + l*cos(rotheta + 3*arctheta);
            whe_ry = whe_ly + l*sin(rotheta + 3*arctheta);
            count = [count; 0 1; 0 1; 0 1];
            whe_lx = whe_rx + l*cos(pi + rotheta - arctheta);
            whe_ly = whe_ry + l*sin(pi + rotheta - arctheta);
            count = [count; 1 0];
        else
```

```
            whe_lx = whe_rx + l*cos(pi + rotheta + 3*arctheta);
            whe_ly = whe_ry + l*sin(pi + rotheta + 3*arctheta);
            count = [count; 1 0; 1 0; 1 0];
            whe_rx = whe_lx + l*cos(rotheta - arctheta);
            whe_ry = whe_ly + l*sin(rotheta - arctheta);
            count = [count; 0 1];
        end

    else
        direc = pos(delt+ts) - pos(ts);
        theta = atan2(norm(cross(direc,roor)), dot(direc,roor));
        rotheta = atan2(roor(2),roor(1));
        sgn = sign(theta-pi/2);

        if sgn > 0
            whe_rx = whe_lx + l*cos(rotheta + arctheta);
            whe_ry = whe_ly + l*sin(rotheta + arctheta);
            count = [count; 0 1];
        else
            whe_lx = whe_rx + l*cos(pi + rotheta - arctheta);
            whe_ly = whe_ry + l*sin(pi + rotheta - arctheta);
            count = [count; 1 0];
        end
    end

    ropo_x2 = (whe_lx + whe_rx)/2;   % Robot Position in the x
    ropo_y2 = (whe_ly + whe_ry)/2;   % Robot Position in the y

    dist = dist + sqrt((ropo_x - ropo_x2)^2 + (ropo_y - ropo_y2)^2);    %
Calculate distance

    ropo_x = ropo_x2; ropo_y = ropo_y2; % Calculate new location

    position_vec = [position_vec; ropo_x ropo_y];

end


%% Plot Function

% Explanation:
% The purpose of this portion or code is to plot the parametric path and
% the path of the robot based on encoder values. It then saves itself as
% a figure.

t = a:(b-a)/1000:b;

xp = posx(t);
yp = posy(t);

% Create and Change to Data Folder

Folder = pwd;
[PathStr,FolderName] = fileparts(Folder);
```

```
plotpathname = [FolderName ' Path Print'];

if ~exist('data_plots', 'dir')
  mkdir('data_plots');
end

cd('data_plots')

% Path Figure

figure(1)
plot(xp,yp,position_vec(:,1),position_vec(:,2), 'o','MarkerSize',8)

leg = legend('Intended Path of Robot');
title('Path of a Two Wheeled Robot Odometry')
xlabel('Position in x')
ylabel('Position in y')

saveas(1, plotpathname, 'png');
save('variables.mat')

for run = 1:val

%%  Bernoulli Application

% Explanation:
% The purpose of this portion of code is to use the array of encoder count
% values and apply the Bernoulli to each encoder vector separately. This
% is done for many iterations, as specified by the user.

count_mod = zeros(max(size(count)), min(size(count)), max(size(iter)));

for k = 1:max(size(iter)) % For all iterations
    for i = 1:length(count) - 1 % Length of the count array
        for j = 1:2 % For each wheel encoder
            if count(i + 1, j) ~= count(i, j) && binornd(1, pr) == 1
                if count(i, j) == 0
                    count_mod(i, j, k) = 1;
                else
                    count_mod(i, j, k) = 0;
                end
            else
                count_mod(i, j, k) = count(i, j);
            end
        end
    end
end

%% Generating Final Points

ropo_mod_final_pos = zeros(max(size(iter)), 2);

parfor i = 1:max(size(iter))
```

```
% Mod Wheel Left Position (Initial)

whe_lx_mod = posx(0) + l / 2 * cos(pi / 2 + init_direc);
whe_ly_mod = posy(0) + l / 2 * sin(pi / 2 + init_direc);

% Mod Wheel Right Position (Initial)

whe_rx_mod = posx(0) + l / 2 * cos(3 * pi / 2 + init_direc);
whe_ry_mod = posy(0) + l / 2 * sin(3 * pi / 2 + init_direc);

for j = 2:max(size(count))

    % Mod Robot Position
    ropo_x_mod = (whe_lx_mod + whe_rx_mod) / 2;
    ropo_y_mod = (whe_ly_mod + whe_ry_mod) / 2;

    roor_mod = [(whe_rx_mod - whe_lx_mod) (whe_ry_mod - whe_ly_mod) 0];
    rotheta_mod = atan2(roor_mod(2), roor_mod(1));

    if count_mod(j, 1, i) == 0
        whe_rx_mod = whe_lx_mod + l * cos(rotheta_mod + arctheta);
        whe_ry_mod = whe_ly_mod + l * sin(rotheta_mod + arctheta);
    else
        whe_lx_mod = whe_rx_mod + l * cos(pi + rotheta_mod - arctheta);
        whe_ly_mod = whe_ry_mod + l * sin(pi + rotheta_mod - arctheta);
    end

end

ropo_mod_final_pos(i, :) = [ropo_x_mod ropo_y_mod];

end

% Plot Final Points

final_pos(:, :, run) = ropo_mod_final_pos;

save('final_pos')

end

figure(2)
plot(xp, yp, position_vec(:, 1), position_vec(:, 2), 'o', 'MarkerSize', 8)

hold on
for run = 1:val
    plot(final_pos(:, 1, run), final_pos(:, 2, run), 'x', 'MarkerSize', 2,
'Color', 'Green')
end

leg = legend('Intended Path of Robot');
title('Path of a Two Wheeled Robot Odometry with Final Locations')
xlabel('Position in x')
```

```
ylabel('Position in y')

plotlocationname = [FolderName ' Location Print'];
saveas(2, plotlocationname, 'png');

cd('..')
```

MATH 3320

Group 2: Nic Rodriguez, Brandon Donnermeyer, Evan McDowell

Project's Report Feedback

Required Sections	Maximum Grade	Your grade	My comments
Introduction	15	15	Well done
Statement of the Problem	25	20	It was not well written at the end
Methodology	25	20	You did not explain how you got the OE equation. If you came up with the equation, what was the motivation. if you did not, provide a citation
Results	25	25	Except from a couple of typos, the section was well written and limitations of the method was given
Bibliography	10	10	Well done
Other comments			A title is missing for your project
Total	100	**90**	

Project's Presentation Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Be properly attired	5	5	Well done. Adequate for the subject chosen
Comments of another presentation	35	30	Well done
Chemistry among group members	20	20	Well coordinated
Timely submission of the presentation	20	20	You submitted your milestones on time
Pass all 4 Milestones	20	20	Well done
Other comments			Your video editing was very well done.
Total	100	**95**	

TRINITY UNIVERSITY

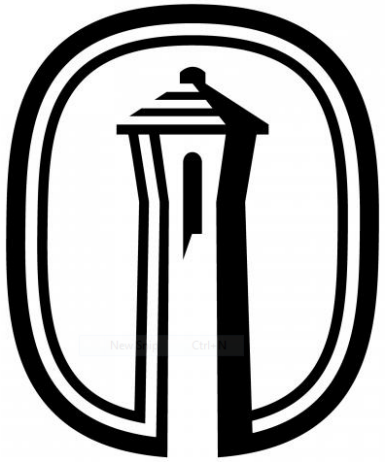
Pledged,

Group 2 Final Report

Math 3320

Nic Rodriguez Brandon Donnermeyer Evan McDowell

12/1/16



Introduction:

Sports in the United States give people unbridled excitement and entertainment, and can sometimes act as an escape from everyday life. The competitive nature of professional sports in tandem with the obsession over statistical record keeping and scrutiny of records have captivated the sports enthusiasts of the world. Because of this, and because our group closely relates to athletics and competitive sports in general, our group chose the project prompt that coupled professional sports and probability. We chose to use the United States and its most popular sport, American Football as our specific area of interest due to its highly competitive environment, rules, and system the NFL abides by are very familiar to all members of our group. Using this collective understanding to our advantage, we sought to seek an answer to a question that dwells on every NFL fan's mind prior to the beginning of each season: "How many games will we win this season?" Being the topic of many pre-season debates, we wanted to create a method that could be used to progressively predict the number of wins a team could earn during the regular season using the measurable statistics from that week's contest.

Problem Statement:

We're want to determine if the number of wins a franchise in the NFL earns in a season can be correlated to the measurable statistics provided by a team's weekly performance. By experimenting with several of these measurable statistics such as turnover ratio, yards gained, yards lost, and average point differential, our group's goal is to be able to prove or disprove the experimental hypothesis of whether the weekly measurable statistics of any NFL team can be used to accurately predict the number of wins they will be able to achieve in a single season. Specifically, we are going to consider a success any prediction that is within a full game of the actual number of wins a team earns during the regular season.

Methodology:

To create a model for predicting the number of wins in a season we needed to obtain sample data. To do this we used "sports-reference.com" and downloaded statistical data from the last 15 years of NFL football starting from 2002 and ending at the current 2016 season. Since the 2016 season was still ongoing we updated the 2016 data with at end of the last regular season game of each week. Our strategy for creating a mathematical model was rather simple. We'd look at one stat at a time and make a scatter plot in excel of the specific stat vs the number of wins per team per season. If the stat looked promising, we would conduct a simple linear regression analysis using excel to determine if we would proceed further. After doing a quick analysis of many different of the different stats available to us we decided to go with point differential which is the difference between points scored and points allowed each game and a stat we developed called offensive efficiency. Offensive efficiency (OE) considers average run yards per game, average run yards allowed per game, average pass yards, average pass yards allowed per game, turnovers, and opponents turnovers.

$$OE = \frac{\frac{Run\ yards}{Run\ yards\ allowed} * \frac{pass\ yards}{pass\ yards\ allowed}}{\frac{Run\ yards}{Run\ yards\ allowed} * \frac{pass\ yards}{pass\ yards\ allowed} + \frac{turnovers}{opponents\ turnovers}} \quad (1)$$

Scatter plots for both can be seen below.

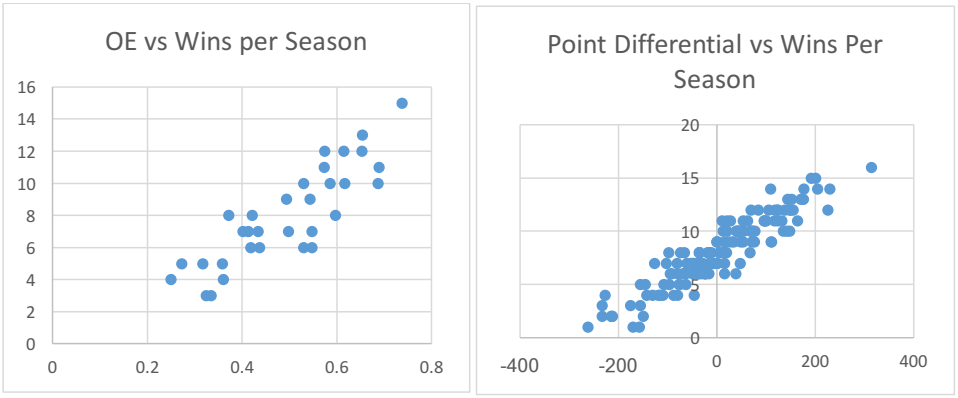


Figure 1: (left) OE for the 2015 season. (right) Point difference for the 2015 and 2014 season.

We uploaded the stats to Matlab for further analysis. Since both at first glance look like they could take on a linear model we did a linear regression analysis on both stats and concluded due to the complexity of the OE model, the time it took to collect the data necessary for that particular model, and the fact that the linear regression analysis showed more of a linear correlation with the point differential model we decided to focus all our efforts on building a mathematical model to predict the number of wins per season using the point difference stat.

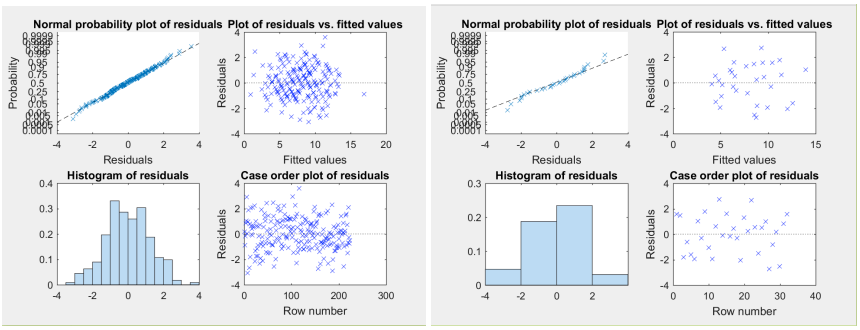


Figure 2: Linear regression analysis of point difference vs wins (left). Linear regression analysis of OE vs wins (right).

To build the model we used Matlab to run a 1000 simulations using a random 70% of the data every time. With each simulation Matlab gave us different coefficients for our linear model so we summed each coefficient and divided the sum by the number of simulations to obtain an average.

$$Wins = x1 + PointDifference * x2 \quad (2)$$

The model can be seen in equation 2 with x1 being the intercept coefficient and x2 being the slope coefficient. Once we obtained the average coefficients predicting the number of wins per season for each team was the easy part. We simply had Matlab calculate the number of projected wins by using equation 2 and inputting each team's respective point difference. Before we went and predicted the outcome of the 2016 season we needed to verify that the model held up outside the range of data used to build it so we repeated the process used to build the first model but we split the in half and used half to build the model and half to test the model. Once the we verified that it held up we went and obtained predictions for the 2016 NFL season using the data accumulated through the first 12 weeks of the 2016 season. However, to obtain the predictions we had to slightly alter equation 2 to account for the fact that not all 16 regular season games had been played in the 2016 season. Instead of just using straight point difference we used the average point difference per game which didn't change the overall outcome of previous models it simply changed the coefficients of the equation as seen in equation 3.

Results:

After we narrowed our focus down to two different stats, OE and point difference, we performed a linear fit of the data using Matlab. The linear fit of the OE data produced an R-squared vale of 0.76 and a root mean squared error of 1.52. The point difference data produced an R-squared value of 0.849 and a root mean squared error of 1.2, which implies a much better correlation than the OE data. Since both data sets seemed promising we did a linear regression analysis for both OE and point difference which can be seen in figure 2 and concluded that while both sets of data followed a linear trend, point difference could be used to build a better model than the OE data. We than simulated all 448 collective NFL seasons from 2002-2015 one thousand times using a random 70% test data for each simulation and produced the equation seen below.

$$Wins = 7.9903 + 0.4462 * \left(\frac{PointDifference}{\#\ of\ games\ played}\right) \quad (3)$$

To test that this was an accurate model of the given data we plugged in the point difference for each team and produced the data seen in appendix A-1. Using this model, we obtained an overall average difference of 0.95 games between predicted wins and actual wins for all individual 448 NFL seasons since 2002. Before we could predict the end of the 2016 season we needed to verify that our model could hold true outside of the given range of data so we split our data in half and used the 2002-2008 seasons to build a model and used the built model to predict the 2009 – 2015 seasons. In doing so we obtained an average game difference of 0.9129 games which was better than the previous model using the complete set of data. We than used the original model that contains all 448 season to predict the end of the 2016 season.

'New England Patriots*'	11.8
'Green Bay Packers*'	7.4
'St. Louis Rams'	5.4
'Miami Dolphins'	8.3
'Kansas City Chiefs*'	9.5
'Buffalo Bills'	9.8
'Carolina Panthers*'	7.8
'New York Jets'	5.2
'Arizona Cardinals'	8.7
'San Diego Chargers'	8.9
'Tennessee Titans'	8.4
'Cincinnati Bengals*'	6.7
'Denver Broncos*'	9.9
'Dallas Cowboys'	12.1
'Atlanta Falcons'	10.2
'San Francisco 49ers*'	3.4
'Pittsburgh Steelers'	9.7
'Chicago Bears'	4.6
'Seattle Seahawks'	9.5
'Washington Redskins'	8.6
'Houston Texans*'	6.3
'Indianapolis Colts*'	6.8
'Jacksonville Jaguars'	4.8
'Cleveland Browns'	2.3
'New Orleans Saints*'	9.1
'Detroit Lions'	8.3
'Tampa Bay Buccaneers'	7.4
'Oakland Raiders'	9.3
'Baltimore Ravens*'	8.7
'New York Giants*'	8.7
'Philadelphia Eagles'	9.6
'Minnesota Vikings*'	9

Figure 3: 2016 predictions using the average point difference per game for each team through week 12.

Some limitations with this model is that close games and blowout wins effect the predictions a lot. Take the Oakland Raiders for example, they have a low point differential per game but and are predicted to win only 9 games. However, as of week 12 they've already won 9 games so the model is saying that they are most likely going to lose the rest of the season. The model also doesn't consider the strength of each team's schedule or the quality of teams they are yet to play so if we were to add those factors into the mix I feel like we could have an extremely accurate model. With the limited amount of time we had, this model has excelled our expectations.

Appendix A-1: tables for 2002-2015 predictions (PWins = Predicted Wins)

Allteams2015	AllWins2015	AllPWins2015
'New England Patriots*'	12	12.1
'Green Bay Packers*'	10	9.2
'St. Louis Rams'	7	6.6
'Miami Dolphins'	6	5.8
'Kansas City Chiefs*'	11	11.2
'Buffalo Bills'	8	8.5
'Carolina Panthers*'	15	13.2
'New York Jets'	10	10
'Arizona Cardinals*'	13	12.8
'San Diego Chargers'	4	5.9
'Tennessee Titans'	3	4.6
'Cincinnati Bengals*'	12	11.8
'Denver Broncos*'	12	9.6
'Dallas Cowboys'	4	5.3
'Atlanta Falcons'	8	7.8
'San Francisco 49ers'	5	3.9
'Pittsburgh Steelers*'	10	10.8
'Chicago Bears'	6	6.3
'Seattle Seahawks*'	10	12
'Washington Redskins*'	9	8.2
'Houston Texans*'	9	8.7
'Indianapolis Colts'	8	5.9
'Jacksonville Jaguars'	5	6
'Cleveland Browns'	3	3.8
'New Orleans Saints'	7	6.1
'Detroit Lions'	7	6.8
'Tampa Bay Buccaneers'	6	5.9
'Oakland Raiders'	7	6.9
'Baltimore Ravens'	5	6
'New York Giants'	6	7.4
'Philadelphia Eagles'	7	6.5
'Minnesota Vikings*'	11	9.7

AverageGameDifference2015 =

0.8281

Allteams2014	AllWins2014	AllPWins2014
'New England Patriots*'	12	12.2
'Green Bay Packers*'	12	11.8
'St. Louis Rams'	6	7.2
'Miami Dolphins'	8	8.4
'Kansas City Chiefs'	9	10
'Buffalo Bills'	9	9.5
'Carolina Panthers*'	7	7
'New York Jets'	4	4.8
'Arizona Cardinals*'	11	8.3
'San Diego Chargers'	9	8
'Tennessee Titans'	2	3
'Cincinnati Bengals*'	10	8.6
'Denver Broncos*'	12	11.5
'Dallas Cowboys*'	12	11.1
'Atlanta Falcons'	6	7
'San Francisco 49ers'	8	7.1
'Pittsburgh Steelers*'	11	9.8
'Chicago Bears'	5	4.6
'Seattle Seahawks*'	12	11.8
'Washington Redskins'	4	4.2
'Houston Texans'	9	9.8
'Indianapolis Colts*'	11	10.4
'Jacksonville Jaguars'	3	3.5
'Cleveland Browns'	7	6.9
'New Orleans Saints'	7	7.4
'Detroit Lions*'	11	9.1
'Tampa Bay Buccaneers'	2	4.4
'Oakland Raiders'	3	2.5
'Baltimore Ravens*'	10	10.9
'New York Giants'	6	7.4
'Philadelphia Eagles'	10	10
'Minnesota Vikings'	7	7.5

AverageGameDifference2014 =

0.8031

Allteams2013	AllWins2013	AllPWins2013
'New England Patriots*'	12	10.9
'Green Bay Packers*'	8	7.7
'St. Louis Rams'	7	7.5
'Miami Dolphins'	8	7.5
'Kansas City Chiefs*'	11	11.4
'Buffalo Bills'	6	6.6
'Carolina Panthers*'	12	11.4
'New York Jets'	8	5.3
'Arizona Cardinals'	10	9.5
'San Diego Chargers*'	9	9.3
'Tennessee Titans'	7	7.5
'Cincinnati Bengals*'	11	11.4
'Denver Broncos*'	13	13.6
'Dallas Cowboys'	8	8.2
'Atlanta Falcons'	4	5.5
'San Francisco 49ers*'	12	11.6
'Pittsburgh Steelers'	8	8.2
'Chicago Bears'	8	7.1
'Seattle Seahawks*'	13	13.1
'Washington Redskins'	3	4.1
'Houston Texans'	2	3.8
'Indianapolis Colts*'	11	9.5
'Jacksonville Jaguars'	4	2.5
'Cleveland Browns'	4	5.3
'New Orleans Saints*'	11	11
'Detroit Lions'	7	8.5
'Tampa Bay Buccaneers'	4	5.2
'Oakland Raiders'	4	4.4
'Baltimore Ravens'	8	7.1
'New York Giants'	7	5.6
'Philadelphia Eagles*'	10	9.6
'Minnesota Vikings'	5	5.6

AverageGameDifference2013 =

0.8094

Allteams2012	AllWins2012	AllPWins2012
'New England Patriots*'	12	14.2
'Green Bay Packers*'	11	10.6
'St. Louis Rams'	7	6.6
'Miami Dolphins'	7	7.2
'Kansas City Chiefs*'	2	2.1
'Buffalo Bills'	6	5.5
'Carolina Panthers*'	7	7.8
'New York Jets'	6	5.4
'Arizona Cardinals'	5	5.1
'San Diego Chargers*'	7	8
'Tennessee Titans'	6	4.1
'Cincinnati Bengals*'	10	9.9
'Denver Broncos*'	13	13.2
'Dallas Cowboys'	8	7.3
'Atlanta Falcons'	13	11.3
'San Francisco 49ers*'	11	11.4
'Pittsburgh Steelers'	8	8.6
'Chicago Bears'	10	10.7
'Seattle Seahawks*'	11	12.5
'Washington Redskins'	10	9.3
'Houston Texans'	12	10.3
'Indianapolis Colts*'	11	7.2
'Jacksonville Jaguars'	2	2.8
'Cleveland Browns'	5	6.2
'New Orleans Saints*'	7	8.2
'Detroit Lions'	4	6.2
'Tampa Bay Buccaneers'	7	7.8
'Oakland Raiders'	4	3.8
'Baltimore Ravens'	10	9.5
'New York Giants'	9	10.3
'Philadelphia Eagles*'	4	3.5
'Minnesota Vikings'	10	8.8

AverageGameDifference2012 =

0.9437

Allteams2011	AllWins2011	AllPWins2011
'New England Patriots*'	13	12.7
'Green Bay Packers*'	15	13.5
'St. Louis Rams'	2	2.1
'Miami Dolphins'	6	8.4
'Kansas City Chiefs*'	7	4.5
'Buffalo Bills'	6	6.3
'Carolina Panthers*'	6	7.4
'New York Jets'	8	8.4
'Arizona Cardinals'	8	7
'San Diego Chargers'	8	8.8
'Tennessee Titans'	9	8.2
'Cincinnati Bengals*'	9	8.6
'Denver Broncos*'	8	5.8
'Dallas Cowboys'	8	8.6
'Atlanta Falcons'	10	9.4
'San Francisco 49ers*'	13	12.1
'Pittsburgh Steelers'	12	10.7
'Chicago Bears'	8	8.3
'Seattle Seahawks*'	7	8.1
'Washington Redskins*'	5	5.8
'Houston Texans*'	10	10.8
'Indianapolis Colts*'	2	2.9
'Jacksonville Jaguars'	5	5.6
'Cleveland Browns'	4	5.6
'New Orleans Saints*'	13	13.7
'Detroit Lions'	10	10.4
'Tampa Bay Buccaneers'	4	2.3
'Oakland Raiders'	8	6
'Baltimore Ravens*'	12	11
'New York Giants'	9	7.8
'Philadelphia Eagles'	8	9.8
'Minnesota Vikings*'	3	5

AverageGameDifference2011 =

1.0750

Allteams2010	AllWins2010	AllPWins2010
'New England Patriots*'	14	13.6
'Green Bay Packers*'	10	12
'St. Louis Rams'	7	6.9
'Miami Dolphins'	7	6.3
'Kansas City Chiefs*'	10	9.1
'Buffalo Bills'	4	4.1
'Carolina Panthers*'	2	2.2
'New York Jets'	11	9.7
'Arizona Cardinals'	5	4
'San Diego Chargers'	9	11.2
'Tennessee Titans'	6	8.4
'Cincinnati Bengals*'	4	6
'Denver Broncos*'	4	4.5
'Dallas Cowboys'	6	6.8
'Atlanta Falcons'	13	11.4
'San Francisco 49ers*'	6	6.9
'Pittsburgh Steelers'	12	11.9
'Chicago Bears'	11	9.3
'Seattle Seahawks'	7	5.3
'Washington Redskins'	6	5.9
'Houston Texans*'	6	7
'Indianapolis Colts*'	10	9.3
'Jacksonville Jaguars'	8	6.2
'Cleveland Browns'	5	6.3
'New Orleans Saints*'	11	10.1
'Detroit Lions'	6	7.8
'Tampa Bay Buccaneers'	10	8.6
'Oakland Raiders'	8	9.1
'Baltimore Ravens*'	12	10.4
'New York Giants*'	10	9.3
'Philadelphia Eagles'	10	9.7
'Minnesota Vikings*'	6	6.2

AverageGameDifference2010 =

1.0469

Allteams2009	AllWins2009	AllPWins2009
'New England Patriots*'	10	11.9
'Green Bay Packers*'	11	12.5
'St. Louis Rams'	1	0.9
'Miami Dolphins'	7	7.2
'Kansas City Chiefs*'	4	4.4
'Buffalo Bills'	6	6.1
'Carolina Panthers*'	8	8.2
'New York Jets'	9	11
'Arizona Cardinals'	10	9.4
'San Diego Chargers'	13	11.6
'Tennessee Titans'	8	6.7
'Cincinnati Bengals*'	10	8.4
'Denver Broncos*'	8	8
'Dallas Cowboys'	11	11
'Atlanta Falcons'	9	9
'San Francisco 49ers*'	8	9.3
'Pittsburgh Steelers'	9	9.2
'Chicago Bears'	7	6.7
'Seattle Seahawks'	5	5
'Washington Redskins'	4	6.1
'Houston Texans*'	9	9.5
'Indianapolis Colts*'	14	11
'Jacksonville Jaguars'	7	5.5
'Cleveland Browns'	5	4.4
'New Orleans Saints*'	13	12.6
'Detroit Lions'	2	1.6
'Tampa Bay Buccaneers'	3	3.7
'Oakland Raiders'	5	3
'Baltimore Ravens*'	9	11.5
'New York Giants*'	8	7.3
'Philadelphia Eagles'	11	10.5
'Minnesota Vikings*'	12	12.3

AverageGameDifference2009 =

0.8844

Allteams2008	AllWins2008	AllPWins2008
'New England Patriots*'	11	10.7
'Green Bay Packers*'	6	9.1
'St. Louis Rams'	2	1.6
'Miami Dolphins'	11	8.7
'Kansas City Chiefs*'	2	3.9
'Buffalo Bills'	7	7.8
'Carolina Panthers*'	12	10.3
'New York Jets'	9	9.3
'Arizona Cardinals'	9	8
'San Diego Chargers'	8	10.5
'Tennessee Titans'	13	11.8
'Cincinnati Bengals*'	4	3.6
'Denver Broncos*'	8	5.9
'Dallas Cowboys'	9	7.9
'Atlanta Falcons'	11	9.8
'San Francisco 49ers*'	7	6.8
'Pittsburgh Steelers'	12	11.4
'Chicago Bears'	9	8.7
'Seattle Seahawks'	4	5.3
'Washington Redskins'	8	7.1
'Houston Texans*'	8	7.2
'Indianapolis Colts*'	12	10.1
'Jacksonville Jaguars'	5	6.2
'Cleveland Browns'	4	4.8
'New Orleans Saints*'	8	9.9
'Detroit Lions'	0	1.2
'Tampa Bay Buccaneers'	9	9
'Oakland Raiders'	5	4.6
'Baltimore Ravens*'	11	11.8
'New York Giants*'	12	11.6
'Philadelphia Eagles'	9	11.5
'Minnesota Vikings*'	10	9.2

AverageGameDifference2008 =

1.1344

Allteams2007	AllWins2007	AllPWins2007
'New England Patriots*'	16	16.6
'Green Bay Packers*'	13	11.9
'St. Louis Rams'	3	3.2
'Miami Dolphins'	1	3.3
'Kansas City Chiefs*'	4	5
'Buffalo Bills'	7	5.2
'Carolina Panthers*'	7	5.8
'New York Jets'	4	5.6
'Arizona Cardinals'	8	8.1
'San Diego Chargers'	11	11.5
'Tennessee Titans'	10	8.1
'Cincinnati Bengals*'	7	7.8
'Denver Broncos*'	7	5.6
'Dallas Cowboys'	13	11.5
'Atlanta Falcons'	4	3.8
'San Francisco 49ers*'	5	4
'Pittsburgh Steelers'	10	11.4
'Chicago Bears'	7	7.6
'Seattle Seahawks'	10	10.8
'Washington Redskins'	9	8.6
'Houston Texans*'	8	7.8
'Indianapolis Colts*'	13	13.1
'Jacksonville Jaguars'	11	10.9
'Cleveland Browns'	10	8.5
'New Orleans Saints*'	7	7.7
'Detroit Lions'	7	5.3
'Tampa Bay Buccaneers'	9	9.7
'Oakland Raiders'	4	4.8
'Baltimore Ravens*'	5	5
'New York Giants*'	10	8.6
'Philadelphia Eagles'	8	9
'Minnesota Vikings*'	8	9.5

AverageGameDifference2007 =

0.9406

Allteams2006	AllWins2006	AllPWins2006
'New England Patriots*'	12	12
'Green Bay Packers*'	8	6.2
'St. Louis Rams'	8	7.6
'Miami Dolphins'	6	7.4
'Kansas City Chiefs*'	9	8.4
'Buffalo Bills'	7	7.7
'Carolina Panthers*'	8	7
'New York Jets'	10	8.6
'Arizona Cardinals'	5	5.9
'San Diego Chargers'	14	13.1
'Tennessee Titans'	8	5.9
'Cincinnati Bengals*'	8	9.1
'Denver Broncos*'	9	8.4
'Dallas Cowboys'	9	10
'Atlanta Falcons'	7	7
'San Francisco 49ers*'	7	4.9
'Pittsburgh Steelers'	8	9
'Chicago Bears'	13	12.7
'Seattle Seahawks'	9	7.8
'Washington Redskins'	5	6.1
'Houston Texans*'	6	5.3
'Indianapolis Colts*'	12	9.8
'Jacksonville Jaguars'	8	10.6
'Cleveland Browns'	4	4.8
'New Orleans Saints*'	10	10.5
'Detroit Lions'	3	5.4
'Tampa Bay Buccaneers'	4	4.1
'Oakland Raiders'	2	3.5
'Baltimore Ravens*'	13	12.1
'New York Giants*'	8	7.8
'Philadelphia Eagles'	10	9.9
'Minnesota Vikings*'	6	6.8

AverageGameDifference2006 =

1.0125

Allteams2005	AllWins2005	AllPWins2005
'New England Patriots*'	10	9.1
'Green Bay Packers*'	4	6.7
'St. Louis Rams'	6	6.2
'Miami Dolphins'	9	8
'Kansas City Chiefs*'	10	10.1
'Buffalo Bills'	5	5.4
'Carolina Panthers*'	11	11.6
'New York Jets'	4	4.8
'Arizona Cardinals'	5	5.9
'San Diego Chargers'	9	10.9
'Tennessee Titans'	4	4.7
'Cincinnati Bengals*'	11	9.9
'Denver Broncos*'	13	11.7
'Dallas Cowboys'	9	8.4
'Atlanta Falcons'	8	8.3
'San Francisco 49ers*'	4	2.8
'Pittsburgh Steelers'	11	11.6
'Chicago Bears'	11	9.6
'Seattle Seahawks'	13	12.9
'Washington Redskins'	10	9.8
'Houston Texans*'	2	3.3
'Indianapolis Colts*'	14	13.2
'Jacksonville Jaguars'	12	10.5
'Cleveland Browns'	6	6.1
'New Orleans Saints*'	3	3.5
'Detroit Lions'	5	5.5
'Tampa Bay Buccaneers'	11	8.7
'Oakland Raiders'	4	5.4
'Baltimore Ravens*'	6	7.1
'New York Giants*'	11	10.9
'Philadelphia Eagles'	6	5.9
'Minnesota Vikings*'	9	6.9

AverageGameDifference2005 =

0.9000

Allteams2004	AllWins2004	AllPWins2004
'New England Patriots*'	14	12.8
'Green Bay Packers*'	10	9.2
'St. Louis Rams'	8	6
'Miami Dolphins'	4	5.8
'Kansas City Chiefs*'	7	9.3
'Buffalo Bills'	9	11
'Carolina Panthers*'	7	8.4
'New York Jets'	10	10
'Arizona Cardinals'	6	6.9
'San Diego Chargers'	12	11.6
'Tennessee Titans'	5	5.4
'Cincinnati Bengals*'	8	8
'Denver Broncos*'	10	10.1
'Dallas Cowboys'	6	4.9
'Atlanta Falcons'	11	8.1
'San Francisco 49ers*'	2	2.7
'Pittsburgh Steelers'	15	11.3
'Chicago Bears'	5	5.3
'Seattle Seahawks'	9	7.9
'Washington Redskins'	6	7.3
'Houston Texans*'	7	7.2
'Indianapolis Colts*'	12	12.7
'Jacksonville Jaguars'	9	7.5
'Cleveland Browns'	4	4.9
'New Orleans Saints*'	8	6.4
'Detroit Lions'	6	6.5
'Tampa Bay Buccaneers'	5	7.9
'Oakland Raiders'	5	4.7
'Baltimore Ravens*'	9	9.3
'New York Giants*'	6	6.8
'Philadelphia Eagles'	13	11.4
'Minnesota Vikings*'	8	8.3

AverageGameDifference2004 =

1.1250

Allteams2003	AllWins2003	AllPWins2003
'New England Patriots*'	14	11
'Green Bay Packers*'	10	11.7
'St. Louis Rams'	12	11.2
'Miami Dolphins'	10	9.4
'Kansas City Chiefs*'	13	12.1
'Buffalo Bills'	6	7
'Carolina Panthers*'	11	8.6
'New York Jets'	6	7.5
'Arizona Cardinals'	4	1.8
'San Diego Chargers'	4	4.5
'Tennessee Titans'	12	11
'Cincinnati Bengals*'	8	6.9
'Denver Broncos*'	10	10.2
'Dallas Cowboys'	10	8.8
'Atlanta Falcons'	5	4.6
'San Francisco 49ers*'	7	9.3
'Pittsburgh Steelers'	6	7.2
'Chicago Bears'	7	6.3
'Seattle Seahawks'	10	10.1
'Washington Redskins'	5	5.7
'Houston Texans*'	5	4.6
'Indianapolis Colts*'	12	11
'Jacksonville Jaguars'	5	6.5
'Cleveland Browns'	5	6.1
'New Orleans Saints*'	8	8.4
'Detroit Lions'	5	5
'Tampa Bay Buccaneers'	7	9
'Oakland Raiders'	4	5
'Baltimore Ravens*'	10	11
'New York Giants*'	4	4.1
'Philadelphia Eagles'	12	10.4
'Minnesota Vikings*'	9	9.7

AverageGameDifference2003 =

1.0719

Allteams2002	AllWins2002	AllPWins2002
'New England Patriots*'	9	8.9
'Green Bay Packers*'	12	9.9
'St. Louis Rams'	7	6.5
'Miami Dolphins'	9	10.1
'Kansas City Chiefs*'	8	9.8
'Buffalo Bills'	8	7.5
'Carolina Panthers*'	7	6.8
'New York Jets'	9	8.6
'Arizona Cardinals'	5	3.8
'San Diego Chargers'	8	7.1
'Tennessee Titans'	11	9.2
'Cincinnati Bengals*'	2	3.1
'Denver Broncos*'	9	9.3
'Dallas Cowboys'	5	4.9
'Atlanta Falcons'	9	10.4
'San Francisco 49ers*'	10	8.4
'Pittsburgh Steelers'	10	9.2
'Chicago Bears'	4	5.3
'Seattle Seahawks'	7	7.6
'Washington Redskins'	7	6.4
'Houston Texans*'	4	4.1
'Indianapolis Colts*'	10	9
'Jacksonville Jaguars'	6	8.3
'Cleveland Browns'	9	8.6
'New Orleans Saints*'	9	9.2
'Detroit Lions'	3	4
'Tampa Bay Buccaneers'	12	12.1
'Oakland Raiders'	11	12
'Baltimore Ravens*'	7	6.9
'New York Giants*'	10	9.1
'Philadelphia Eagles'	12	12.7
'Minnesota Vikings*'	6	6.6

AverageGameDifference2002 =

0.8375

Bibliography:

Mathworks Support Page. (n.d.). Retrieved November 10, 2016, from https://www.mathworks.com/support/?s_tid=gn_supp

@. (n.d.). Pro Football Statistics and History | Pro-Football-Reference.com. Retrieved November 10, 2016, from http://www.pro-football-reference.com/

H. (n.d.). The Complete History Of The NFL. Retrieved November 10, 2016, from http://projects.fivethirtyeight.com/complete-history-of-the-nfl/

MATH 3320

Group 3: Mattheus Righes, Rolando Morales, Vivek Poovathoor

Project's Report Feedback

Required Sections	Maximum Grade	Your grade	My comments
Introduction	15	10	You added a paragraph that should have been in the conclusion instead
Statement of the Problem	25	20	This was missing. It was not clear reading your project to clearly know what you wanted to accomplish
Methodology	25	25	This was well explained. It shows that you had a good understanding of the project.
Results	25	25	Well explained as well. Those results are meaningful
Bibliography	10	10	Well done
Other comments			
Total	100	**90**	

Project's Presentation Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Be properly attired	5	5	Well done. Adequate for the subject chosen
Comments of another presentation	35	30	Well done
Chemistry among group members	20	10	You need an improvement there
Timely submission of the presentation	20	20	You submitted your milestones on time
Pass all 4 Milestones	20	20	Well done
Other comments			Your video was a bit blurry.
Total	100	**85**	

Sonar Sensor Modeling

By: Matheus Righes, Rolando Morales, and Vivek Poovathoor
Engineering Statistics and Probability – Dr. Kwessi
December 1st, 2016

Introduction:

In the field of robotics, sensors are used to localize an automaton within its environment and then update the current grid based on the sensor readings. Robots must have knowledge of their pose within their environment in order to accomplish specific tasks, especially if they are placed in new environments. A common sensor used to detect obstacles is a sonar sensor. This sensor operates by sending sound waves, or "pings", and then returning the time taken for the pulse to travel forth and back. From that signal, using an Arduino Uno, we calculated the distance of the detected object using half of the time it took for the ping to return and the speed of sound (in either centimeters or inches). Specifically, we used the HC-SR04 sonar sensor which is equipped with four pins: VCC, GND, ECHO, and TRIG. We hoped to give a probabilistic model of this sonar sensor as it moved through an arbitrary grid with objects randomly placed throughout the map; we desired to use the varying distances and angles at which the sensor detected an object in order to develop a representative, probabilistic model. From the collected data of readings of randomly placed objects we determined that no distinguishable distribution of the sonar sensor could be determined from a surface and histogram plot of our dataset. Furthermore, we observed that the lack of shape to the distribution can be attributed to the uncertainties of the sensor's measurement method which was able to pick up small protrusions in the grid environment in addition to the edges of objects that were placed at greater distances.

Without introducing much room for human error, we decided to derive our data set from a sonar sensor that is fixed within a hemispherical grid and is able to rotate between 0 and 180 degrees. Within the grid were three randomly placed cylindrical objects that were intended to be detected by the HC-SR04 sonar sensor. Also, because the grid was limited to certain dimensions, any measurement reading that was beyond the confines of the custom map was defined as an empty space (or zero). The data was then organized using and plotted as histogram in Matlab and plotted as a bivariate probability distribution using Minitab.

Our ultimate takeaway from this experiment is that due to the sensitivity of the sonar sensor, as it detects an object, and the uncertainties of the measurements, especially at greater distances, a definitive probabilistic model of the sensor could not be attributed to a distinguishable distribution and the sensor has is not accurate in differentiating between objects.

Methodologies:

We first devised a map/grid (which was constructed out of large paper) that we would place the sensor on. It is assumed that each grid had independent probabilities and the locations of the objects in the grid were all random. The hemispherical grid was drawn in a polar coordinate fashion, with radii separated 5 degrees from each other to a maximum distance of 70 cm (drawn from 0 to 180 degrees). Three cylindrical objects were then placed randomly on the grid, all of which were to be intended to be detected by the sonar sensor; we neglect the shape of the grid when detecting objects using the sensor because we are only seeking to determine the probabilistic locations of objects. A model of this method and the locations of the objects is shown in **Figure A-1.**

Next we designed a circuit that would allow us to record the distance readings using the HC-SR04, as shown in **Figure A-5.** Using an Arduino Uno microcontroller and basic circuit components such as switches, resistors, pushbuttons, etc. we built the circuit for the sonar sensor on a breadboard platform. The circuit would work by requiring the user to depress a pushbutton

which would signal the sonar sensor to make 100 distance readings. An Arduino script (which is a variant of C) was written in order to send commands to the sonar sensor to send trigger signals and to listen for echoes from any object. In order to store and save the raw data from the sensor for analysis, a Python script was written that would take in the serial data readings from the microcontroller and write the readings to a file of any name. The execution of the script would finish as soon the sonar sensor made 150 readings.

Once the sensor (including the rest of the circuit), programs, and grid were all interfaced we conducted our data collection method. We positioned the sensor first to 0 degrees and collected distance readings, which were stored to a file named "0" using the Python script. Then the sensor was moved to 5 degrees, in which the process was repeated, then 10 degrees, then 15 degrees, and so on until 180 degrees was finally reached. In each angular position the sonar sensor collected distance data which would indicate the presence of an object.

Once all the data was collected from the grid, we utilized Python and Matlab scripts (and commands) to first convert the polar coordinate data into rectangular coordinates and organize the data from the files into a tabular format to calculate the frequency for each grid location. A Matlab script was utilized to create the bin widths for a histogram; we determined to use 50 bins for the histogram plot. The 3-D histogram was then plotted in Matlab using the *hist3()* function. Then, using same frequency (z) data, along with the x and y data, we created smooth curve plots (probability plots) of the data using Minitab. We decided to use the graphing capabilities of Matlab and Minitab to see how well the sensor was able to collect the objects' data and to see what sort of distribution would fit the probability graph of the sonar sensor data. A 3-D histogram was used because we varied both the angular position and the radial distance of the randomly placed objects. Certain constraints were placed to take into account pulses that never return to the sonar sensor or send erroneous results. Therefore, a maximum limit of 70 cm was set (the radius of our grid) such that no object(s) beyond 70 cm would be registered. Any distance greater than this limit was arbitrarily registered to be an empty space, i.e. a distance of 0 cm; and all the objects placed on the grid were within this upper limit.

Results:

The raw data of the sonar sensor reading was plotted first in a 3-D histogram against varying radial distances and sweeping-angles of the sensor. This graph allowed us to determine the likely location of the objects purely based off of sensory data. From this histogram, it can be seen that the sensor does not exhibit a noticeable nor unique probability distribution. Using Minitab, we fitted a smooth curve which yielded the same shape as the histogram, which also showed no distinguishable distribution to the probability of the sensor. The sensor also picked up some disturbances (bumps on the surface of grid) which were depicted in both the histogram and the probability curve. These unexpected distance readings occurred twice, one very close to the sensor and one towards the father, radial end of the hemisphere. These errors also contributed to the lack of shape to the probability curve of the sonar sensor readings.

From the 5500 samples that were collected 3337 of these were pings from the sonar sensor indicating that some sort of object has been detected. The events of object 1, object 2, and object 3 are mutually exclusive and independent because each object was situated in non-overlapping locations on the grid and the detection of one of the objects does not affect the chances of the other two objects being detected. The probability that our sensor detected one of the randomly placed cylindrical objects is calculated as follows:

$$Let\ t\ be\ the\ total\ number\ of\ pings$$
$$Let\ o1\ be\ the\ number\ of\ pings\ due\ to\ object\ 1$$
$$Let\ o2\ be\ the\ number\ of\ pings\ due\ to\ object\ 2$$
$$Let\ o3\ be\ the\ number\ of\ pings\ due\ to\ object\ 3$$

$$Let\ p\ be\ the\ number\ of\ pings\ that\ correctly\ detect\ either\ object$$

$$We\ want\ \Pr\ (either\ object\ 1, object\ 2\ or\ object\ 3\ were\ detected)$$

$$o1 = 300$$
$$o2 = 450$$
$$o3 = 463$$
$$t = 3337$$

$$\mathrm{P}(o1, o2, or\ o3) = \mathrm{P}(\mathrm{o1}) + P(o2) + P(o3)$$
$$= \frac{300}{3337} + \frac{450}{3337} + \frac{463}{3337}$$

$$\mathrm{P}(\mathrm{o1}, \mathrm{o2}, \mathrm{or\ o3}) = 0.3635$$

Discussion:

The probability surface plots of the sonar sensor readings do not indicate a readily identifiable distribution. The two-dimensional bins show the locations where the sensor detected some sort of object, irrespective of the shape or size as shown in **Figure A-2** and **Figure A-3**. In the first attempt to detect the obstacles in our grid we used angular divisions of 10 degrees instead of 5 degrees; and in our Matlab analysis the bin widths were too narrow to capture the variability of the distance readings of the sonar sensor. Therefore, we increased the divisions to 5 degrees. The height of the bins of the histogram depict the presence of consistent detections of obstacles. Although the histogram suggests that there are two large objects in the grid a closer look at it will show otherwise. Firstly, it can be observed that there is a region of a supposedly large object around 135 degrees that has lower probability, as suggested by **Figure A-4**, which show lower probability-heights at these locations. In other words, it separates two regions with higher probabilities. Those two regions represent the two objects that were situated close to each other. The reason why the separation isn't obvious is because a sonar wave increases in radius as it moves further from its source, thus bouncing back from objects that may not be in the same "line of sight". Therefore, at the angles between the objects, the sonar will detect those objects even though they do not lie at the respective angle. However, the probabilities of those readings are not greater than as if the objects were placed at such an angle, thus we can notice that there are three objects in the whole grid, as expected. For all of those reasons, although it can be seen from the probability density distribution that there is a probability of an object being in a range of angles that is larger than the object itself, the regions where we actually had the objects are very close to the regions where we had the highest probabilities of an object being there.

The sensor was able to identify vacant grid cells, but the presence of small bins in the 2-D distribution show how sensitive the sensor is to small disturbances in our semicircular grid. These small locations were due to the slight change in elevation of the paper that was used to construct the map. Thus, the bins of noticeably smaller heights are also depicted that indicate the presence of small "ridges". These locations have lower frequencies because the sensor did not consistently detect the presence of these obstacles as consistently as the larger cylindrical objects that were randomly placed and meant to be detected by the sonar sensor.

The probability of either object 1, objects 2, or object 3 to be detected indicate the accuracy of the sensor. The number of pings due to either of objects were collected by pinpointing the location of each object and which grid cells each of the occupied. From there we collected the number of pings that the sensor sent to that grid cell, which was yielded as we converted our semicircular, polar grid to a two-dimensional Cartesian coordinate grid. A high proportion would imply that of the total number of pings detected that a sufficient amount would be attributable to the presence of the actual objects, however as shown, only 36.35% of the total pings sent can be attributed to either object 1, object 2, or object 3. From this we can say that the sensor that was used is quite sensitive but not accurate.

The primary area of improvement would include constructing a grid that minimizes the amount of disturbances as well as use more objects to further test the accuracy of the sensor. A larger grid would provide more samples to analyze and determine if an identifiable distribution can be fit to the dataset, however as of now the data does not appear to fit a commonly applicable distribution. In addition to a more strategic construction of our grid, a method that would control angular adjustments to the sensor would yield in more accurate data. The current method employed is manual adjustments which employ human error.

Conclusion:

In this project the task was to develop a probabilistic distribution of an HC-SR04 sonar sensor by analyzing data from the sensor as it was swept through a semicircular grid that contained three randomly situated objects. Based on the results from computer analysis, using Python, Matlab and Minitab, the distribution did not appear to fit any commonly used distributions. The accuracy of the sensor was quantified by determining the probability that the times that the correct objects were identified out of the total number of times the sensor detected some sort of object. This proportion was 0.3635, indicating a low accuracy of the sensor due to its sensitivity. The method could be improved by creating a more robust map as well as add more randomly placed objects throughout the grid to verify the accuracy, or lack therof, of HC-SR04 sensor.

References:

[1] ***Probabilistic Robotics*****. Sebastian Thrun, Wolfram Burgard, and Dieter Fox. (2005, MIT Press.) 647 pages**

[2] ***Exploring Arduino.*** **Jeremy Blum (2013, John Wiley & Sons, Inc.) 384 pages**

Appendix:

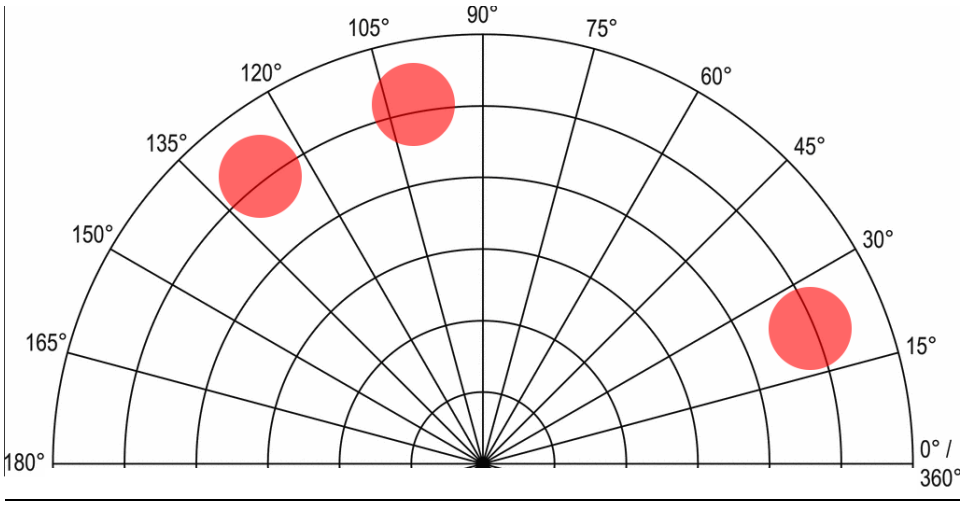


Figure A-1: Map of grid with surface areas of three randomly placed objects.

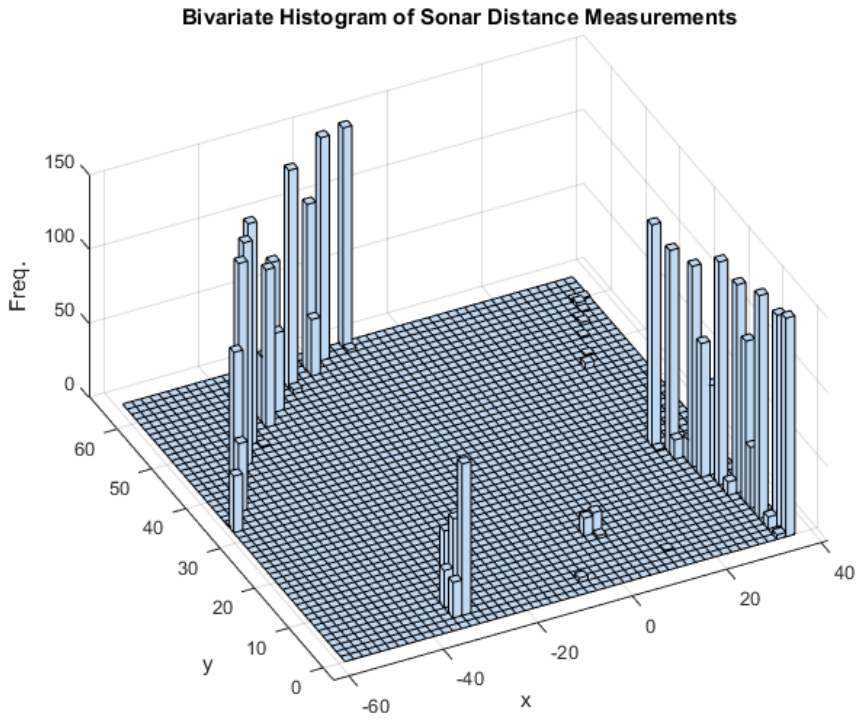


Figure A-2: Histogram generated by Matlab of sonar distance measurements in semicircular grid.

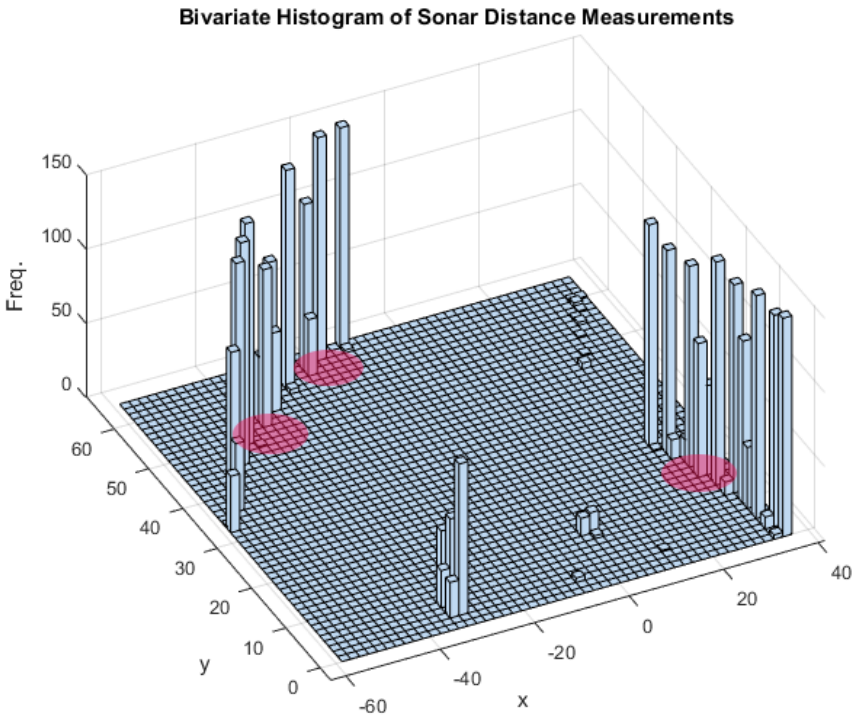


Figure A-3: Histogram of sonar distance with actual locations of the three objects.

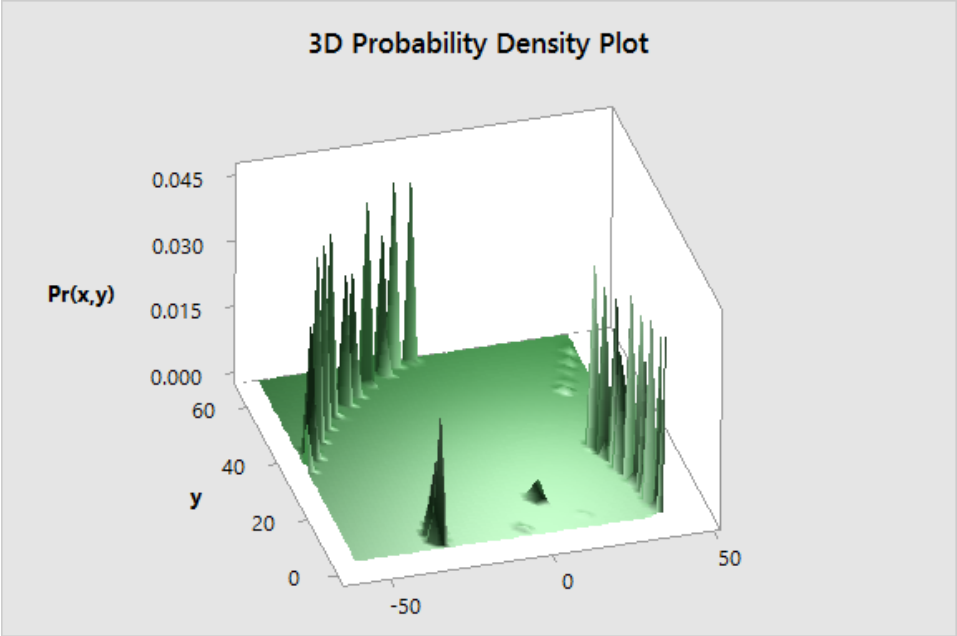


Figure A-4: Probability distribution plot of object detections generated by Minitab.

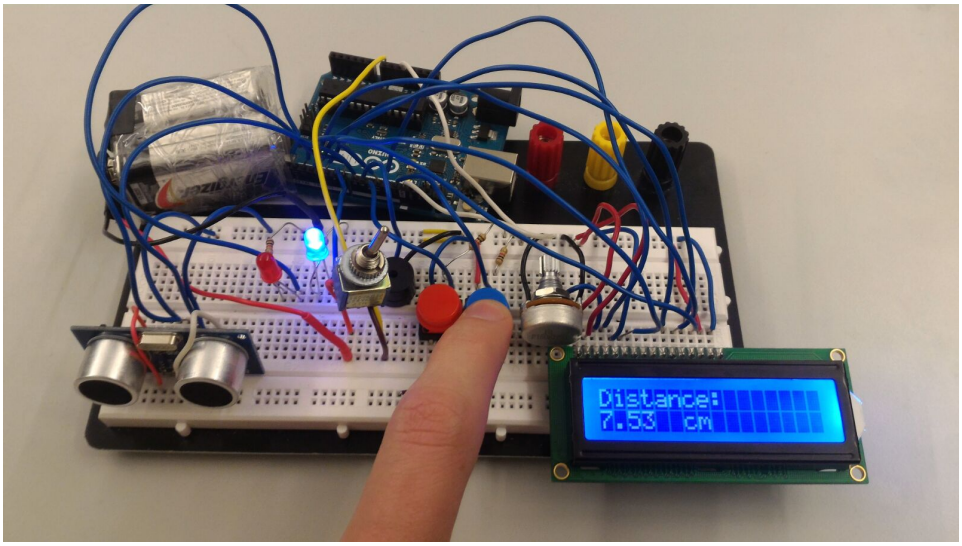


Figure A-5: Custom circuit used for data acquisition using HC-SRO4 and Arduino Uno.

MATH 3320

Group 4: Anna Kroll, Anirudth Tibrewal, Nicholas LoBue

Project's Report Feedback

Required Sections	Maximum Grade	Your grade	My comments
Introduction	15	10	It could have been easily been the statement of your problem
Statement of the Problem	25	20	Same comment as above.
Methodology	25	25	Well explained except for some confusion in the meaning of Q-Q in Q- Q Plot. It means Quantile Quantile plot, not Quartile-Quartile. Quantile refers to quarters only whereas quantile refers to percent.
Results	25	25	Well explained as well. You had a good understanding of the limitation of your own method
Bibliography	10	10	Well done
Other comments			
Total	100	**85**	

Project's Presentation Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Be properly attired	5	5	Well done. Adequate for the subject chosen
Comments of another presentation	35	30	Well done
Chemistry among group members	20	10	You need an improvement there
Timely submission of the presentation	20	20	You submitted your milestones on time
Pass all 4 Milestones	20	20	Well done
Other comments			Your video was a bit blurry.
Total	100	**85**	

Trinity University

Probability and Statistics for Engineers & Scientists
Semester Project Report
Group #5

Group Members: Davi Dias and Joao Marques
Faculty Advisor: Dr. Eddy Kwessi

Abstract

The purpose of this project was to give students a better understanding on the concepts of probability and statistics inside Texas Hold'em Poker. The students first gathered analytical data and studied the probabilities of several hands inside the game. A presentation accompanying this report demonstrated the results while providing some mathematical laws to prove the data collected was indeed satisfactory. The principle of multiplication, the binomial coefficient and combinatorics were the main tools utilized by our group to analyze the data. Although the collected results showed accuracy, our group did not take into account some other issues inside the game such as bluffing and personal behavior of each player on the board.

Table of contents

1. Initiation

1.1 Background

"Poker Texas Hold'em is a variation of the card game of poker. Two cards, known as the hole cards or hold cards, are dealt face down to each player, and then five are dealt face up in three stages. The stages consist of a series of three cards ("the flop"), later an additional single card ("the turn" or "fourth street") and a final card ("the river" or "fifth street"). Each player seeks the best five card poker hand from the combination of the community cards and their own hole cards. If a player's best five card poker hand consists only of the five community cards and none of the player's hole cards, it is called "playing the board". Players have betting options to check, call, raise or fold. Rounds of betting take place before the flop is dealt, and after each subsequent deal." [2]

1.2 Introduction

The Texas Hold'em Poker is not just about pure luck. This project goes deeper into the probabilities inside the game to show how important is for the players to know which hands are more common than the others. The more the player knows, the easier it is to develop a successful decision-making and strategy. To determine which hands are more common, we show different techniques to predict the probabilities in certain parts of the game.

There are two different ways to calculate probabilities in a poker game. The first, that should be only applied when the game has first started, is dividing the number of outcomes that satisfy the condition being evaluated, by the total number of cards in the game. This calculation will not be valid when the deck has no longer 52 cards in it. On the other hand, the second approach is more applicable to the game, because it involves conditional probability, and the calculations are updated at every game move.

This project will examine the likelihood to win a hand analyzing various stages of the game. Our group first approached starting hands, breaking it down into single and dominated hands. We focused on pocket pairs, which are the most desired hand combinations to start the rounds. For this section, we based ourselves on the odds of obtaining each different card combination, without taking into account the probability to win the hand. Then, we looked into what changes when the flop, the turn and the river are played in the game. With this analysis, we were able to indicate the chances of a successful outcome in the game, considering all game stages. We used three mathematical models that explain the probabilities: the binomial coefficient, the principle of multiplication and the Monte Carlo algorithm.

2. Methodologies

Our group used the website www.cardplayer.com in order to analyze a specific given set of data regarding the different combinations of the deck of cards used in the *Texas Hold'em Poker* game. Our data collection includes an analysis of three different scenarios: the starting hands, the flop and the after flop. Within the starting hands conditions, we analyzed both single hands and dominated hands with pocket pair scenarios.

2.1 Single Hands w/ Pocket Pair

In order to identify how many possible starting hands a single player can have in one round of the game, we explicitly calculated its probability using the binomial coefficient rule. If there are 52 different cards in the deck, and a player draws two out of the 52, the probability can be calculated by doing 52 nCr 2**.** We have found it to be 1,326 different starting hands possibilities. Then, this number was divided into subcategories and placed in a table (see Table1 in the *Results* section) for better understanding. As

previously discussed in the *Introduction*, we analyzed the pocket pair condition for the starting hands. By applying the binomial coefficient rule, we discovered the number of possibilities for each hand having a suit combination; we calculated it to be 6 since 4 nCr 2 yields this number. Because a deck of cards has 13 different ranks from A to K (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K), we observed that there are only 13 pair possibilities. We calculated the amount of possible combinations by applying the multiplication principle; this value yielded 78, since 6 x 13 = 78.

2.2 Dominated Hands w/ Pocket pair

We continued the analysis of the game with the next different condition. A dominated hand is a hand that is beaten by another hand and is unlikely to win against it. For the pocket pair case, we observed that a dominated hand only occurs when a pocket pair has a higher rank. One can notice that the only way a lower rank pocket pair can win against a higher rank pocket pair is if, after the flop, the lower pocket pair becomes a three of a kind. For more information on this topic, see the *After Flop* methodology section. In order to find if another player on the table has a higher rank pocket pair, we considered a scenario with only two players. The probability that a single opponent has a higher pair can be stated as the probability that the first card dealt to the opponent is a higher rank than the pocket pair and the second card is the same rank as the first. By subtracting two cards from the deck (the two cards that form the pocket pair for the player), the number of cards in the deck decreases to 50. Once the opponent is dealt with its first card, there are 49 cards left in the deck, from which 3 have the same rank as the first opponent's card. Observing this pattern, we could develop a formula to find out if a single opponent has a higher rank pocket pair. The formula is given below, where R is the rank of the pocket pair:

Equation 1:

$$P = \frac{(14-R)\,x\,4}{50}\ x\ \frac{3}{49}$$

To verify our expectations and to prove the expressed formula would give the expected results, we used the previously discussed website simulator.

2.3 The Flop

The flop is the moment of the game when three cards are dealt with the face up to the board. In order to calculate the number of possible combinations at the flop, we used the binomial coefficient rule. After the player is given two cards from the deck, there are 50 remaining left cards in the deck to be played. When the flop occurs, the number of possible combinations are given by applying the binomial coefficient 50 nCr 3, which yields to 19,600. This number means that there are 19,600 possible combinations after the three cards are dealt with the face-up to the board.

Our group then used the principle of multiplication to find out the probability of specific flops in the game. In other words, we simply multiply together the probabilities of each of those cards being dealt. For example, suppose we want to find out the probability of flopping three A's. We multiply 4 aces over the 50 remaining cards, then we have only 3 aces left, divided by 49 remaining cards and so on, as seen bellow:

$$(4/50) * (3/49) * (2/48) = 0.02048\%.$$

2.4 The After-Flop (Turn and River)

Similarly, we applied the binomial coefficient rule to find out the number of possible combinations for the turn and the river. The turn happens when a fourth card from the deck is dealt with the face-up to the board. Similarly, the river occurs when a fifth card is faced up to the board. Those numbers were calculated to be 230,300 (50 nCr 4) and 2,118,760 (50 nCr 5), respectively. As one may notice, as the number of cards on the board increases, the chances of making a game also increases, due to the fact that more combinations can yield from five cards on the table comparing to four.

2.5 Monte-Carlo Algorithm

Our group used the Monte-Carlo algorithm to simulate/ plot simple rounds of the game not taking into consideration the suits and for a single deal of five cards with no draw. We only simulate a game for four different kinds of outcome combinations: a pair, three-of-a-kind and four-of-a-kind. The code works in the following way: it first generates the whole deck of cards with the 52 cards. Then it create a matrix to store the probability of the three different combinations previously mentioned; it randomizes the card vector (shuffles the deck) and deals a set of 5 cards to 4 players (in the algorithm, we don't consider the effects of the flop and randomly gives 5 cards to each player in the beginning of the game). The algorithm then counts the cards in each hand to look for pairs, three-of-a-kind and four-of-a-kind. Finally, it plots the results of the simulation.

The results for the simulation can be seen in the *Results* section and the MATLAB code can be seen in the Appendix.

3. Results

3.1 Single Hand

Table 1: Single Hand probability

Single Hand	Number of hands	Combination of suits	Total possible combinations	Probability
Pocket Pair	13	6	78	5.8%

As shown in Table 1, there are 13 different types of pocket pairs (AA, KK, QQ, JJ, 10s, 9s, 8s, 7s, 6s, 4s, 3s and 2s). With all suits (spades, hearts, diamonds and clubs), we have a combination 13 x 4nCr2, yielding 78 different combinations. Then, dividing 78 by 1326, which is the total possible starting hand combinations (52 nCr 2), we have a 5.8% probability to start off the game with a pair.

3.2 Dominated Hands

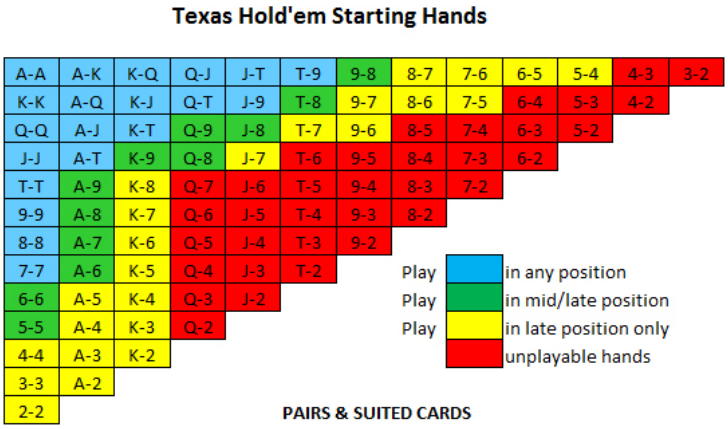


Figure 1: Chart showing how likely it is to win the hand with each combination

Exploring the formula shown in section 2.2, imagine a situation where 2 players face each other in the game: the first player has a Q-Q, but do not know that the second player has a 6-6. According to formula 1, we substitute r = 12 for the queens and r = 6 for the pair of 6. The probability that player 1 will face a larger pair will be 0.98%, while player 2 will face a 3.92%. Now, we see that the first player has a more playable hand, as confirmed in Figure 1.

3.3 The flop

Analyzing the results, we have Table 2 below with some examples that show how the hands will change after the flop:

Table 2: Probabilities to combine certain cards, given a 10-J Hearts:

Hand	Probability
Royal Street Flush	(1/50)*(1/49)*(1/48)x100% = 0.00085%
Four of a kind	(6/50)*(2/49)*(1/48) x100%= 0.01%
Flush	(12/50)*(11/49)*(10/48) x100%= 1.12%
Full House (assuming a pocket pair)	(4/50)*(3/49)*(2/48) x100%= 0.02%
Higher Simple Sequence	(4/50)*(4/49)*(4/48)x100% = 0.054%

Of course these probabilities only show what can be done with the first three open cards on the table, but this is only a demonstration to show how to calculate the probabilities, and to mention that it is curious that it is more likely to obtain a flush than a sequence in this situation, given the flush is more valuable.

3.4 Turn and River

Similarly, the after-flop probability will follow the same procedure done in section 3.3. However, there are a lot of different ways to calculate the chances, since we necessarily have to take into account what was done in the flop. With that in mind, there are thousands of hundreds of combinations after what happened when the first three table cards are opened.

3.5 Monte-Carlo Algorithm

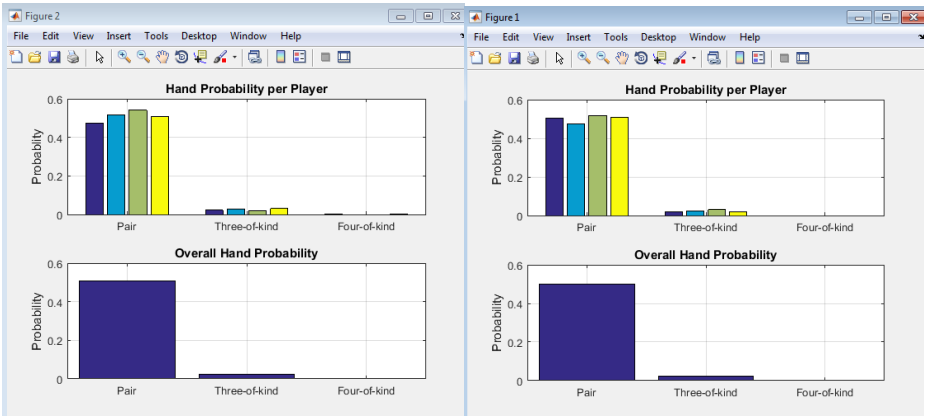


Figure 2 and 3: Monte-Carlo simulation on Matlab showing two different random hands.

As shown above, the Monte-Carlo algorithm simulated two different random hands. From both figures, we see that the probability of getting a Pair is around 50% overall, less than 5% for a Three-of-kind, and insignificant chances for a Four-of-kind.

4. Discussion

Throughout the whole project, our group verified the importance of probability and statistics in *Texas Hold'em Poker.* The binomial coefficient rule provided an overall efficient analysis of the entire game, from exploring the probability of the starting hands to the possibilities in the flop, turn and river.

It was not surprising to see that according to the Monte-Carlo Algorithm, the chances of getting a pair are way higher than a four-of-a-kind. Our group observed that after running several simulations in MATLAB, the scenarios above were those that occurred more often. We also can see that for a random

shuffled deck of cards, a player can have higher chances of winning a game than the others, which makes the game more interesting and fascinating.

Our research does have problems, because the Texas Hold'em Poker is an extremely complicated game that has a lot more variances than those we approached. For example, our group only assumed the probabilities based on the card that were not utilized by a single player, and that is not the complete truth, since there are cards with other players and discarded cards. However, we reasonably based our assumptions on how the player has to see the game, taking into account that any card can be discarded or be in someone else's hand.

Another problem seen in the project was that we did not consider human's interactions that influence the game. Our calculations were strictly mathematics, and we could not foresee how players bluffing would affect the emotional of each player, compromising their likelihood to win.

Lastly, we did not use any Binomial Distributions, as the group stated on the preliminary report. However, we did use the binomial coefficient to show all calculations regarding the probabilities as shown in the methodologies section. In addition to this mathematical model, we observed that the Monte-Carlo algorithm fails to calculate a lot of different kinds of hands. The code used on MATLAB is only capable of demonstrating probabilities for pairs, three and four of a kind.

5. Conclusion

We observed that, even though all the probabilities found are very low, it is possible to develop a relatively strategy based on them. Since Poker is a card game, and it is not designed to yield "successes" all the time, knowing the odds can be helpful to situate the player in the game, making players who are unfamiliar with the subject to have an educated guess on whether or not they should call. If the bet is large, they may feel that it is too expensive to try and catch the right card, but if the bet is small they will be more inclined to call. The key to improve decision-making and strategy is to have in mind what are the chances to form combinations with what the player has in hands. So, we concluded that the most important part of the game is the starting hand, since it will determine the course of the game.

The results showed that starting a hand with a pocket pair is probably the best way to start the game, because more valuable types of combinations can be easier formed from it. For example, it is easier to form a full house starting with a pocket pair than starting with two different random cards. It also showed that a flush can be easier achieved if the player starts with a pocket pair of suits.

In the end, we agreed that doing this kind of detailed analysis is hard to do while in the middle of a hand. However, doing it later, away from the table, helps clarify the likely reality of what was going on in the game.

6. References

Linus (view profile). (n.d.). Retrieved October 28, 2016, from https://www.mathworks.com/matlabcentral/answers/89788-poker-bar-graph-probability

Poker probability (Texas hold 'em). (n.d.). Retrieved October 28, 2016, from https://en.wikipedia.org/wiki/Poker_probability_(Texas_hold_'em)

Texas Hold'em Poker Odds Calculator. (n.d.). Retrieved October 28, 2016, from http://www.cardplayer.com/poker-tools/odds-calculator/texas-holdem

Appendix

```
% Monte Carlo Generation of Poker Hand Probability
% Simple example with no suits, and a single deal of five cards with no
% draw. Only looking at pairs, three-of-a-kinds and four-of-a-kinds.


% Generate all cards in the deck
oneSuit  = 1:13;
deck     = repmat(oneSuit,1,4);
numCards = numel(deck);

% Run Monte Carlo Simulations
% Effectively play large number of hands to determine probabilities
numSimulations = 1000;
numPlayers     = 4;
numCardsDealt  = 5;

% Create matrix to store pairs, etc. for each player
% Three rows as this code only counts pairs, three-of-a-kinds,
% and four-of-a-kinds.
handsTotal     = zeros(3,numPlayers);

for i = 1:numSimulations,

% Randomize card vector, "shuffle the deck"
  shuffleIndex = randperm(numCards);
  deckShuffled = deck(shuffleIndex);

% Deal hands (5x cards a player)
  dealCards = deckShuffled(1:numPlayers*numCardsDealt);
  dealCards = reshape(dealCards,numPlayers,numCardsDealt);

% Count the cards in each hand
  cardCount = histc(dealCards',oneSuit);

% Count singletons, pairs, etc. Eliminate singletons and no card counts
  handCount = histc(cardCount,0:4);
  handCount = handCount(3:end,:);
```

```
% Add counts from this deal to overall count
  handsTotal = handsTotal + handCount;
end

% Plot results
figure;
subplot(2,1,1)
bar(handsTotal./numSimulations);
grid on;
title('Hand Probability per Player');
ylabel('Probablity')
set(gca,'XTick',1:3,'XTickLabel',{'Pair','Three-of-kind',...
    'Four-of-kind'});


subplot(2,1,2)
bar(sum(handsTotal,2)./(numSimulations*numPlayers));
grid on;
title('Overall Hand Probability');
ylabel('Probability')
set(gca,'XTick',1:3,'XTickLabel',{'Pair','Three-of-kind',...
    'Four-of-kind'});
```

MATH 3320

Group 5: Davi Dias, Joao Marques

Project's Report Feedback

Required Sections	Maximum Grade	Your grade	My comments
Introduction	15	13	Well written except for some obvious typos
Statement of the Problem	25	20	This was embedded in the introduction and should have been separated from it
Methodology	25	25	Well explained except for some confusion in the meaning of Q-Q in Q- Q Plot. It means Quantile Quantile plot, not Quartile-Quartile. Quantile refers to quarters only whereas quantile refers to percent.
Results	25	25	Well explained as well. You had a good understanding of the limitation of your own method
Bibliography	10	10	Well done
Other comments			A title for your project is missing
Total	100	**93**	

Project's Presentation Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Be properly attired	5	3	Davi was not properly attired
Comments of another presentation	35	30	Well done
Chemistry among group members	20	10	You need an improvement there.
Timely submission of the presentation	20	20	You submitted your milestones on time
Pass all 4 Milestones	20	20	Well done
Other comments			Your video was a bit blurry and sketchy
Total	100	**83**	

Trinity University

Probability and Statistics for Engineers & Scientists
Semester Project Report
Group #5

Group Members: Davi Dias and Joao Marques
Faculty Advisor: Dr. Eddy Kwessi

Abstract

The purpose of this project was to give students a better understanding on the concepts of probability and statistics inside Texas Hold'em Poker. The students first gathered analytical data and studied the probabilities of several hands inside the game. A presentation accompanying this report demonstrated the results while providing some mathematical laws to prove the data collected was indeed satisfactory. The principle of multiplication, the binomial coefficient and combinatorics were the main tools utilized by our group to analyze the data. Although the collected results showed accuracy, our group did not take into account some other issues inside the game such as bluffing and personal behavior of each player on the board.

Table of contents

1. Initiation

1.1 Background

"Poker Texas Hold'em is a variation of the card game of poker. Two cards, known as the hole cards or hold cards, are dealt face down to each player, and then five are dealt face up in three stages. The stages consist of a series of three cards ("the flop"), later an additional single card ("the turn" or "fourth street") and a final card ("the river" or "fifth street"). Each player seeks the best five card poker hand from the combination of the community cards and their own hole cards. If a player's best five card poker hand consists only of the five community cards and none of the player's hole cards, it is called "playing the board". Players have betting options to check, call, raise or fold. Rounds of betting take place before the flop is dealt, and after each subsequent deal." [2]

1.2 Introduction

The Texas Hold'em Poker is not just about pure luck. This project goes deeper into the probabilities inside the game to show how important is for the players to know which hands are more common than the others. The more the player knows, the easier it is to develop a successful decision-making and strategy. To determine which hands are more common, we show different techniques to predict the probabilities in certain parts of the game.

There are two different ways to calculate probabilities in a poker game. The first, that should be only applied when the game has first started, is dividing the number of outcomes that satisfy the condition being evaluated, by the total number of cards in the game. This calculation will not be valid when the deck has no longer 52 cards in it. On the other hand, the second approach is more applicable to the game, because it involves conditional probability, and the calculations are updated at every game move.

This project will examine the likelihood to win a hand analyzing various stages of the game. Our group first approached starting hands, breaking it down into single and dominated hands. We focused on pocket pairs, which are the most desired hand combinations to start the rounds. For this section, we based ourselves on the odds of obtaining each different card combination, without taking into account the probability to win the hand. Then, we looked into what changes when the flop, the turn and the river are played in the game. With this analysis, we were able to indicate the chances of a successful outcome in the game, considering all game stages. We used three mathematical models that explain the probabilities: the binomial coefficient, the principle of multiplication and the Monte Carlo algorithm.

2. Methodologies

Our group used the website www.cardplayer.com in order to analyze a specific given set of data regarding the different combinations of the deck of cards used in the *Texas Hold'em Poker* game. Our data collection includes an analysis of three different scenarios: the starting hands, the flop and the after flop. Within the starting hands conditions, we analyzed both single hands and dominated hands with pocket pair scenarios.

2.1 Single Hands w/ Pocket Pair

In order to identify how many possible starting hands a single player can have in one round of the game, we explicitly calculated its probability using the binomial coefficient rule. If there are 52 different cards in the deck, and a player draws two out of the 52, the probability can be calculated by doing 52 nCr 2**.** We have found it to be 1,326 different starting hands possibilities. Then, this number was divided into subcategories and placed in a table (see Table1 in the *Results* section) for better understanding. As

previously discussed in the *Introduction*, we analyzed the pocket pair condition for the starting hands. By applying the binomial coefficient rule, we discovered the number of possibilities for each hand having a suit combination; we calculated it to be 6 since 4 nCr 2 yields this number. Because a deck of cards has 13 different ranks from A to K (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K), we observed that there are only 13 pair possibilities. We calculated the amount of possible combinations by applying the multiplication principle; this value yielded 78, since 6 x 13 = 78.

2.2 Dominated Hands w/ Pocket pair

We continued the analysis of the game with the next different condition. A dominated hand is a hand that is beaten by another hand and is unlikely to win against it. For the pocket pair case, we observed that a dominated hand only occurs when a pocket pair has a higher rank. One can notice that the only way a lower rank pocket pair can win against a higher rank pocket pair is if, after the flop, the lower pocket pair becomes a three of a kind. For more information on this topic, see the *After Flop* methodology section. In order to find if another player on the table has a higher rank pocket pair, we considered a scenario with only two players. The probability that a single opponent has a higher pair can be stated as the probability that the first card dealt to the opponent is a higher rank than the pocket pair and the second card is the same rank as the first. By subtracting two cards from the deck (the two cards that form the pocket pair for the player), the number of cards in the deck decreases to 50. Once the opponent is dealt with its first card, there are 49 cards left in the deck, from which 3 have the same rank as the first opponent's card. Observing this pattern, we could develop a formula to find out if a single opponent has a higher rank pocket pair. The formula is given below, where R is the rank of the pocket pair:

Equation 1:

$$P = \frac{(14-R)\,x\,4}{50}\ x\ \frac{3}{49}$$

To verify our expectations and to prove the expressed formula would give the expected results, we used the previously discussed website simulator.

2.3 The Flop

The flop is the moment of the game when three cards are dealt with the face up to the board. In order to calculate the number of possible combinations at the flop, we used the binomial coefficient rule. After the player is given two cards from the deck, there are 50 remaining left cards in the deck to be played. When the flop occurs, the number of possible combinations are given by applying the binomial coefficient 50 nCr 3, which yields to 19,600. This number means that there are 19,600 possible combinations after the three cards are dealt with the face-up to the board.

Our group then used the principle of multiplication to find out the probability of specific flops in the game. In other words, we simply multiply together the probabilities of each of those cards being dealt. For example, suppose we want to find out the probability of flopping three A's. We multiply 4 aces over the 50 remaining cards, then we have only 3 aces left, divided by 49 remaining cards and so on, as seen bellow:

$$(4/50) * (3/49) * (2/48) = 0.02048\%.$$

2.4 The After-Flop (Turn and River)

Similarly, we applied the binomial coefficient rule to find out the number of possible combinations for the turn and the river. The turn happens when a fourth card from the deck is dealt with the face-up to the board. Similarly, the river occurs when a fifth card is faced up to the board. Those numbers were calculated to be 230,300 (50 nCr 4) and 2,118,760 (50 nCr 5), respectively. As one may notice, as the number of cards on the board increases, the chances of making a game also increases, due to the fact that more combinations can yield from five cards on the table comparing to four.

2.5 Monte-Carlo Algorithm

Our group used the Monte-Carlo algorithm to simulate/ plot simple rounds of the game not taking into consideration the suits and for a single deal of five cards with no draw. We only simulate a game for four different kinds of outcome combinations: a pair, three-of-a-kind and four-of-a-kind. The code works in the following way: it first generates the whole deck of cards with the 52 cards. Then it create a matrix to store the probability of the three different combinations previously mentioned; it randomizes the card vector (shuffles the deck) and deals a set of 5 cards to 4 players (in the algorithm, we don't consider the effects of the flop and randomly gives 5 cards to each player in the beginning of the game). The algorithm then counts the cards in each hand to look for pairs, three-of-a-kind and four-of-a-kind. Finally, it plots the results of the simulation.

The results for the simulation can be seen in the *Results* section and the MATLAB code can be seen in the Appendix.

3. Results

3.1 Single Hand

Table 1: Single Hand probability

Single Hand	Number of hands	Combination of suits	Total possible combinations	Probability
Pocket Pair	13	6	78	5.8%

As shown in Table 1, there are 13 different types of pocket pairs (AA, KK, QQ, JJ, 10s, 9s, 8s, 7s, 6s, 4s, 3s and 2s). With all suits (spades, hearts, diamonds and clubs), we have a combination 13 x 4nCr2, yielding 78 different combinations. Then, dividing 78 by 1326, which is the total possible starting hand combinations (52 nCr 2), we have a 5.8% probability to start off the game with a pair.

3.2 Dominated Hands

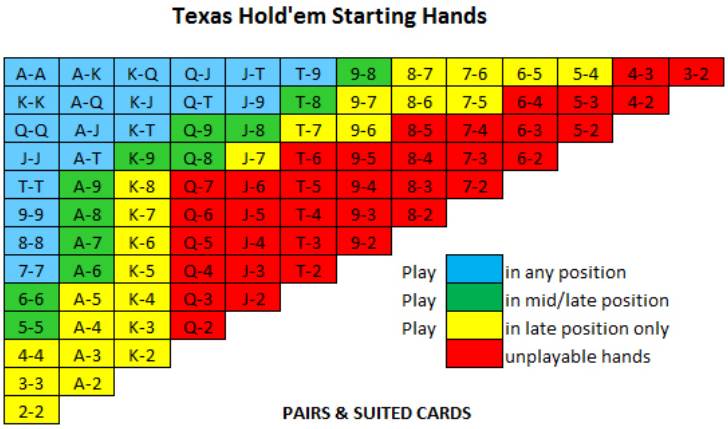


Figure 1: Chart showing how likely it is to win the hand with each combination

Exploring the formula shown in section 2.2, imagine a situation where 2 players face each other in the game: the first player has a Q-Q, but do not know that the second player has a 6-6. According to formula 1, we substitute r = 12 for the queens and r = 6 for the pair of 6. The probability that player 1 will face a larger pair will be 0.98%, while player 2 will face a 3.92%. Now, we see that the first player has a more playable hand, as confirmed in Figure 1.

3.3 The flop

Analyzing the results, we have Table 2 below with some examples that show how the hands will change after the flop:

Table 2: Probabilities to combine certain cards, given a 10-J Hearts:

Hand	Probability
Royal Street Flush	(1/50)*(1/49)*(1/48)x100% = 0.00085%
Four of a kind	(6/50)*(2/49)*(1/48) x100%= 0.01%
Flush	(12/50)*(11/49)*(10/48) x100%= 1.12%
Full House (assuming a pocket pair)	(4/50)*(3/49)*(2/48) x100%= 0.02%
Higher Simple Sequence	(4/50)*(4/49)*(4/48)x100% = 0.054%

Of course these probabilities only show what can be done with the first three open cards on the table, but this is only a demonstration to show how to calculate the probabilities, and to mention that it is curious that it is more likely to obtain a flush than a sequence in this situation, given the flush is more valuable.

3.4 Turn and River

Similarly, the after-flop probability will follow the same procedure done in section 3.3. However, there are a lot of different ways to calculate the chances, since we necessarily have to take into account what was done in the flop. With that in mind, there are thousands of hundreds of combinations after what happened when the first three table cards are opened.

3.5 Monte-Carlo Algorithm

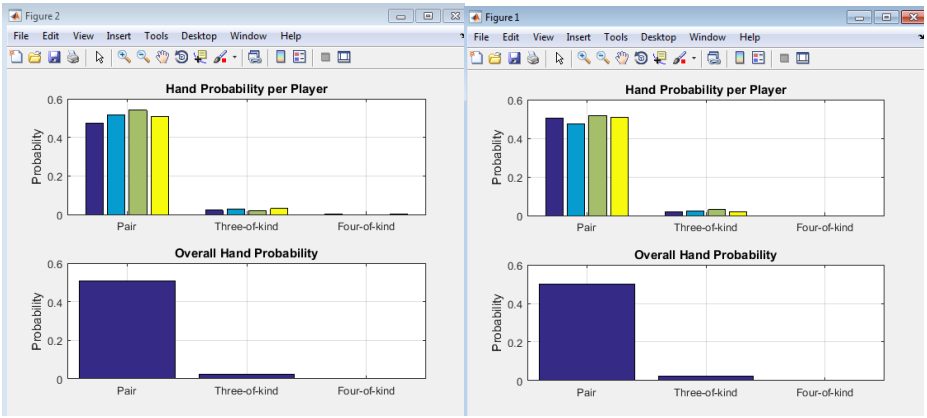


Figure 2 and 3: Monte-Carlo simulation on Matlab showing two different random hands.

As shown above, the Monte-Carlo algorithm simulated two different random hands. From both figures, we see that the probability of getting a Pair is around 50% overall, less than 5% for a Three-of-kind, and insignificant chances for a Four-of-kind.

4. Discussion

Throughout the whole project, our group verified the importance of probability and statistics in *Texas Hold'em Poker.* The binomial coefficient rule provided an overall efficient analysis of the entire game, from exploring the probability of the starting hands to the possibilities in the flop, turn and river.

It was not surprising to see that according to the Monte-Carlo Algorithm, the chances of getting a pair are way higher than a four-of-a-kind. Our group observed that after running several simulations in MATLAB, the scenarios above were those that occurred more often. We also can see that for a random

shuffled deck of cards, a player can have higher chances of winning a game than the others, which makes the game more interesting and fascinating.

Our research does have problems, because the Texas Hold'em Poker is an extremely complicated game that has a lot more variances than those we approached. For example, our group only assumed the probabilities based on the card that were not utilized by a single player, and that is not the complete truth, since there are cards with other players and discarded cards. However, we reasonably based our assumptions on how the player has to see the game, taking into account that any card can be discarded or be in someone else's hand.

Another problem seen in the project was that we did not consider human's interactions that influence the game. Our calculations were strictly mathematics, and we could not foresee how players bluffing would affect the emotional of each player, compromising their likelihood to win.

Lastly, we did not use any Binomial Distributions, as the group stated on the preliminary report. However, we did use the binomial coefficient to show all calculations regarding the probabilities as shown in the methodologies section. In addition to this mathematical model, we observed that the Monte-Carlo algorithm fails to calculate a lot of different kinds of hands. The code used on MATLAB is only capable of demonstrating probabilities for pairs, three and four of a kind.

5. Conclusion

We observed that, even though all the probabilities found are very low, it is possible to develop a relatively strategy based on them. Since Poker is a card game, and it is not designed to yield "successes" all the time, knowing the odds can be helpful to situate the player in the game, making players who are unfamiliar with the subject to have an educated guess on whether or not they should call. If the bet is large, they may feel that it is too expensive to try and catch the right card, but if the bet is small they will be more inclined to call. The key to improve decision-making and strategy is to have in mind what are the chances to form combinations with what the player has in hands. So, we concluded that the most important part of the game is the starting hand, since it will determine the course of the game.

The results showed that starting a hand with a pocket pair is probably the best way to start the game, because more valuable types of combinations can be easier formed from it. For example, it is easier to form a full house starting with a pocket pair than starting with two different random cards. It also showed that a flush can be easier achieved if the player starts with a pocket pair of suits.

In the end, we agreed that doing this kind of detailed analysis is hard to do while in the middle of a hand. However, doing it later, away from the table, helps clarify the likely reality of what was going on in the game.

6. References

Linus (view profile). (n.d.). Retrieved October 28, 2016, from https://www.mathworks.com/matlabcentral/answers/89788-poker-bar-graph-probability

Poker probability (Texas hold 'em). (n.d.). Retrieved October 28, 2016, from https://en.wikipedia.org/wiki/Poker_probability_(Texas_hold_'em)

Texas Hold'em Poker Odds Calculator. (n.d.). Retrieved October 28, 2016, from http://www.cardplayer.com/poker-tools/odds-calculator/texas-holdem

Appendix

```
% Monte Carlo Generation of Poker Hand Probability
% Simple example with no suits, and a single deal of five cards with no
% draw. Only looking at pairs, three-of-a-kinds and four-of-a-kinds.


% Generate all cards in the deck
oneSuit  = 1:13;
deck     = repmat(oneSuit,1,4);
numCards = numel(deck);

% Run Monte Carlo Simulations
% Effectively play large number of hands to determine probabilities
numSimulations = 1000;
numPlayers     = 4;
numCardsDealt  = 5;

% Create matrix to store pairs, etc. for each player
% Three rows as this code only counts pairs, three-of-a-kinds,
% and four-of-a-kinds.
handsTotal     = zeros(3,numPlayers);

for i = 1:numSimulations,

% Randomize card vector, "shuffle the deck"
  shuffleIndex = randperm(numCards);
  deckShuffled = deck(shuffleIndex);

% Deal hands (5x cards a player)
  dealCards = deckShuffled(1:numPlayers*numCardsDealt);
  dealCards = reshape(dealCards,numPlayers,numCardsDealt);

% Count the cards in each hand
  cardCount = histc(dealCards',oneSuit);

% Count singletons, pairs, etc. Eliminate singletons and no card counts
  handCount = histc(cardCount,0:4);
  handCount = handCount(3:end,:);
```

```
% Add counts from this deal to overall count
  handsTotal = handsTotal + handCount;
end

% Plot results
figure;
subplot(2,1,1)
bar(handsTotal./numSimulations);
grid on;
title('Hand Probability per Player');
ylabel('Probablity')
set(gca,'XTick',1:3,'XTickLabel',{'Pair','Three-of-kind',...
    'Four-of-kind'});


subplot(2,1,2)
bar(sum(handsTotal,2)./(numSimulations*numPlayers));
grid on;
title('Overall Hand Probability');
ylabel('Probability')
set(gca,'XTick',1:3,'XTickLabel',{'Pair','Three-of-kind',...
    'Four-of-kind'});
```

MATH 3320

Group 6: Tristan Ashton, Kirsten Endresen, Regis Noubiap

Project's Report Feedback

Required Sections	Maximum Grade	Your grade	My comments
Introduction	15	15	Well written
Statement of the Problem	25	20	This was embedded in the introduction and should have been separated from it
Methodology	25	23	Well explained. There was a clear lack of consistency of notation from distributions to distributions.
Results	25	25	Well explained as well. You had a good understanding of the limitation of your own method
Bibliography	10	10	Well done
Other comments			I liked the explanation and the application provided. I might use your presentation as a motivation for the Central Limit Theorem in the coming years!
Total	100	**93**	

Project's Presentation Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Be properly attired	5	5	Well done.
Comments of another presentation	35	30	Well done
Chemistry among group members	20	15	You did a good job
Timely submission of the presentation	20	20	You submitted your milestones on time
Pass all 4 Milestones	20	20	Well done
Other comments			Your video would have been perfect were it not for the lack of editing
Total	100	**90**	

Trinity University

Probability and Statistics for Engineers and Scientists

Noise Reduction in Digital Images

Group 6
Tristan Ashton
Kirsten Endresen
Regis Noubiap

Professor: Dr. Eddy Kwessi

12/01/2016

Table of Contents

1. Introduction

In the era of digital information, image acquisition and transfer has become a ubiquitous necessity. From the millions of photos uploaded every day to social media, as well as professional photos taken by scientists such as biologists, geologists, and astronomers, photographic images have become increasingly indispensable. The subject matter is widely disparate and varied. The mobile phone industry continually promotes the clarity and pixel density of their cameras to their consumer base; professional photographers spend years learning various softwares to enhance their photos or ensure fidelity to the subject; astronomers give a lot of importance to image quality in their study of astral bodies. Issues of fidelity all stem from noise, and understanding its physical origins allows the development of techniques for minimization.

The goal of this project is, as stated, the minimization of noise rather than its total elimination. Complete elimination encounters the problem of diminishing returns: although it is possible to completely remove noise from an image, it would require a much larger sample size than is strictly necessary to clean up an image to a reasonable degree. Noise of all kinds arises from statistical fluctuations of various origins, and although statistics is a powerful tool it cannot predict individual outcomes of fundamentally random processes. We see several examples of this limitation in physics: the statistical methods underlying quantum mechanics cannot predict what a single measurement will yield. Statistical mechanics cannot predict the motion of a single particle in a gas. Despite problems of randomness, we can measure emergent trends once the sample size grows large, and we can begin to isolate the physical origins of data fluctuations.

The scope of this project includes only four types of noise despite the large catalog of potential statistical errors. Our analysis focused on uniform, gaussian, poisson, and spike (colloquially known as “salt-and-pepper”) noise. These four were chosen because they are the most common types, and their mathematical underpinnings and physical sources are well understood. With this in mind, we aim to examine the efficacy of averaging out different sources of noise in an image by applying noise according to known distributions, summing over combinations of 1, 2, 3, up to 100 images, and analyzing the speed at which we return to the original image.

1.1 Uniform Noise

Uniform noise arises due to transcriptions of essentially continuous elements (information flow from the real world) onto a discrete receptor (the CCD of a camera or the pixels on a screen). Overlap of border elements falls victim to attempted smoothing, and can be seen prominently in low-resolution cameras. Its mathematical description follows a uniform distribution, which is a continuous function.

The uniform distribution has a constant probability density function within a specified range. This distribution is given by

$$P(x) = \begin{cases} \frac{1}{b-a} & a < x < b \\ 0 & otherwise \end{cases}$$

where b is the upper bound of the specified range and a is the lower bound. The variable x can be any real number since the distribution is continuous over the interval from a to b.

Obviously, the mean μ is the average of a and b, calculated by:

$$\mu = \frac{a+b}{2}$$

The variance of the standard deviation is also straightforward to calculate, and is given by

$$Var(x) = \frac{1}{12}(b-a)^2$$

Therefore, the standard deviation is

$$\sigma = \sqrt{\frac{1}{12}(b-a)^2}$$

1.2 Gaussian Noise

Gaussian noise, from the Gaussian or Normal distribution, is commonly seen in digital images. Contributors to Gaussian noise primarily include thermal vibrations, since the atomic or molecular constituents of thermal vibrations will at any time be moving randomly and interfering with the motion of electrons seeking to display an image. We observe Gaussian noise arising from thermal vibrations in the atoms of conductors, blackbody radiation from any source of heat, and in the large-scale limit of shot noise. Shot noise, however, primarily follows a Poisson distribution, which we will consider separately.

The Gaussian distribution has a probability density function described by

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)/(2\sigma^2)}$$

where μ is the center mean, and σ is the standard deviation, which quantifies the spread of the distribution. The variable x can be any real number since the distribution is continuous and extends to positive and negative infinity. The mean can also be any real number, but the standard deviation must have a positive value. Note that the mean and standard deviations are defined explicitly in the probability density function for the Gaussian distribution, so unlike the uniform distribution, additional calculations are not required to customize these parameters. The graph of the Gaussian distribution is the bell curve, which is shown below.

1.3 Poisson Noise

Poisson noise is a fairly common discrete distribution. As mentioned above, its primary physical origin is fluctuations in the number of photons sensed by the receptor at any time: although the numbers are massive, the detector will never register a fraction of a photon, yielding a discrete distribution. Poisson noise is thus often called shot noise, which also encompasses measures of current (which is comprised of discrete electrons). The distribution is always positive, since measuring negative photons (or electrons) is impossible, and each measured event is independent of all the others. It begins to approximate a Gaussian at high intensity levels, so the Poisson is often considered the positive low-mean limit Gaussian.

The mathematical formulation of the Poisson distribution is given by

$$P(X) = \frac{e^{-\mu}\mu^{x}}{x!}$$

where μ is the mean and e is the base of the natural logarithm. The positive and discrete nature is extracted from the factorial term in the denominator, since the factorial of a negative number is undefined, and fractional factorials cannot be done without the aid of the gamma function. In keeping with the requirement that valid probability distributions must go to zero at infinity, we note that the factorial term increases much faster than μ^{x}. Additionally, the distribution fulfills the requirement that the enclosed area must equal 1. In the necessary summation from x = 0 to infinity, $\Sigma\mu^{x}/x!$ is simply a definition of e^{μ}, which then multiplies with $e^{-\mu}$ to yield 1.

The Poisson function arises from the binomial distribution:

$$P_p(n \mid N) = \frac{N!}{n!\,(N-n)!}\, p^n\, (1-p)^{N-n}.$$

where N is the number of trials and p is the probability of a successful trial. By taking the limit of this function in the case of very large N and utilizing Stirling's approximation for large factorial arguments, the formulation of the Poisson distribution emerges.

Since the exponential contains a negative argument, the curve slopes down after passing the peak value at the mean. When plotted, the curve resembles a Gaussian squished against the y-axis. For a small mean, the peak is very close to the y-axis and the curve begins sloping down almost immediately. As the mean increases the peak moves away from the y-axis, creating a small tail and approaching a Gaussian form when the mean becomes large. Also of note is the peak value: as the mean increases, the peak decreases as more nonzero values become enclosed under the right-shifting curve.

It is trivial to obtain the variance since it is simply equal to the mean. Likewise, since the standard deviation is the square root of the variance, in this case it simply equals the square root of the mean. The interconnectedness of the parameters demonstrates that only a single value is required to characterize the curve completely and determine the probability of events that correspond to a Poisson distribution.

1.4 Spike Noise

Salt-and-pepper noise gets its name from the scattering of saturated and empty pixels on an image. Typically one will observe saturated pixels in dark areas and empty pixels in bright areas, as if tossing pinches of salt and pepper across the image. Spike noise is caused by bit transmission errors and analog-to-digital conversion errors. Its mathematics follow a probability tree instead of a statistical distribution. Pixels will become saturated or empty with a certain probability, and remain as measured otherwise. The mathematical analysis of this type of noise is discussed in the procedure section.

1.5 Sampling Distributions and the Central Limit Theorem

This project analyzes the effect of sampling many times from four known probability distributions. When averaging many noisy images, each pixel basically takes the mean noise

sampled from each distribution and adds it to the original pixel value. The composite image, being a grid of many pixels, therefore constitutes a sampling distribution of the mean.
The mean of this sampling distribution remains the same as the population mean, regardless of the sample size or the population distribution, which means

$$\mu_{\bar{x}} = \mu$$

where μ is the population mean and $\mu_{\bar{x}}$ is the mean of the sample distribution. The effective standard deviation of the sample population is given by

$$\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$$

where σ is the standard deviation of the population and n is the sample size. This shows that as we average many images, the total noise of the composite will converge to the population mean.

While we examine each noise type separately, after many iterations of averaging, the sampling distributions begin to behave similarly. Despite the seeming oddity of this convergence, it can be explained by a striking result from the field of statistics: the Central Limit Theorem. This theorem states that distributions of sample averages will converge to fit a normal distribution as the sample size increases, regardless of the distribution of the original population.

2. Methodology

For each of these distributions, we used Matlab to generate random noise and added the noise to a clear image, shown in Figure 1 and Figure 2. We varied the number of noisy images and averaged over the set of these to create a composite image. By varying the size of the set, we studied how quickly each type of noise could be removed by this process.

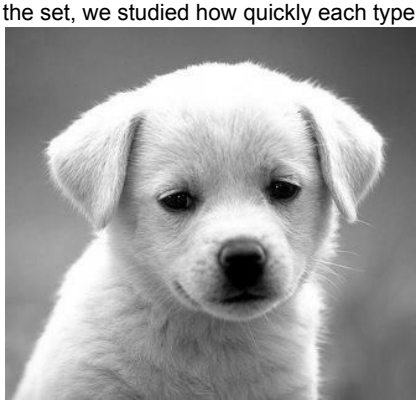


Figure 1. *First clean image*

Figure 2. *Second clean image*

First, the image was loaded into Matlab as an array of integer pixel values between 0, which corresponds to a black pixel, and 255, which corresponds to a white pixel. We can represent the original image as a matrix $f(x,y)$ where x and y are the coordinates of a pixel. From our probability distribution of choice (Gaussian, Uniform, or Poisson), we create an array of zero-mean random noise, the same size as our clean image. We denote this array as $\eta_i(x,y)$. Each noisy image is simply the sum of the noise and the original image, or

$$g_i(x,y) = f(x,y) + \eta_i(x,y)$$

When generating Gaussian noise, we used the Matlab function normrnd to select random numbers from a normal distribution centered at zero with a standard distribution of 50. When generating uniform noise, we used the Matlab function rand to select a random double between 0 and 1 from a uniform distribution, multiplied this value by 173.2 to achieve the same variance that we used for the Gaussian distribution, and subtracted 86.6 to maintain a mean noise value of zero. To generate Poisson noise, we used the Matlab function poissrnd with a lambda value of 2500, which is 50^2 because it gave us a standard deviation of 50, also matching our choice for Gaussian noise. We subtracted 2500, again to maintain a mean noise value of zero.

The spike noise is generated differently. For each pixel, we took a random value $r(x,y)$ between 0 and 1 from a uniform distribution, and the noised pixel was determined based on a probability tree, as follows:

$$g_i(x',y') = \begin{cases} 0 & \text{if } r(x',y') < \varepsilon \\ 255 & \text{if } r(x',y') > 1-\varepsilon \\ f(x',y') & \text{otherwise} \end{cases}$$

where ε was manually selected in order to generate our individual images with a noise standard deviation of 50 to match the other distributions. We found that the spike noise was most comparable to the noise from the other distributions at $\varepsilon = 0.065$, so we chose this value for analysis.

We generated 100 individual images with noise generated from each of the four distributions. Then, we began averaging multiple images together by computing

$$\bar{g}(x,y) = \frac{1}{n}\sum_{i=1}^{n} g_i(x,y)$$

where n is the number of noisy images that contributed to each average. We varied the number of images between 2 to 100 and captured the composite figures.

We analyzed each of these images first by comparing them to the original clear image, to determine qualitatively whether averaging was an effective way to minimize noise. Next, we generated histograms to show the sampling distributions of the mean for each value of n. This allowed us to visually examine whether our results were congruent with the Central Limit Theorem, as we hypothesized. Finally, we computed the standard deviations of the composite noise for our various sample sizes and compared them to theoretical values. The composite noise was calculated by

$$\bar{\eta}(x,y) = \frac{1}{n}\sum_{i=1}^{n} \eta_i(x,y)$$

or equivalently,

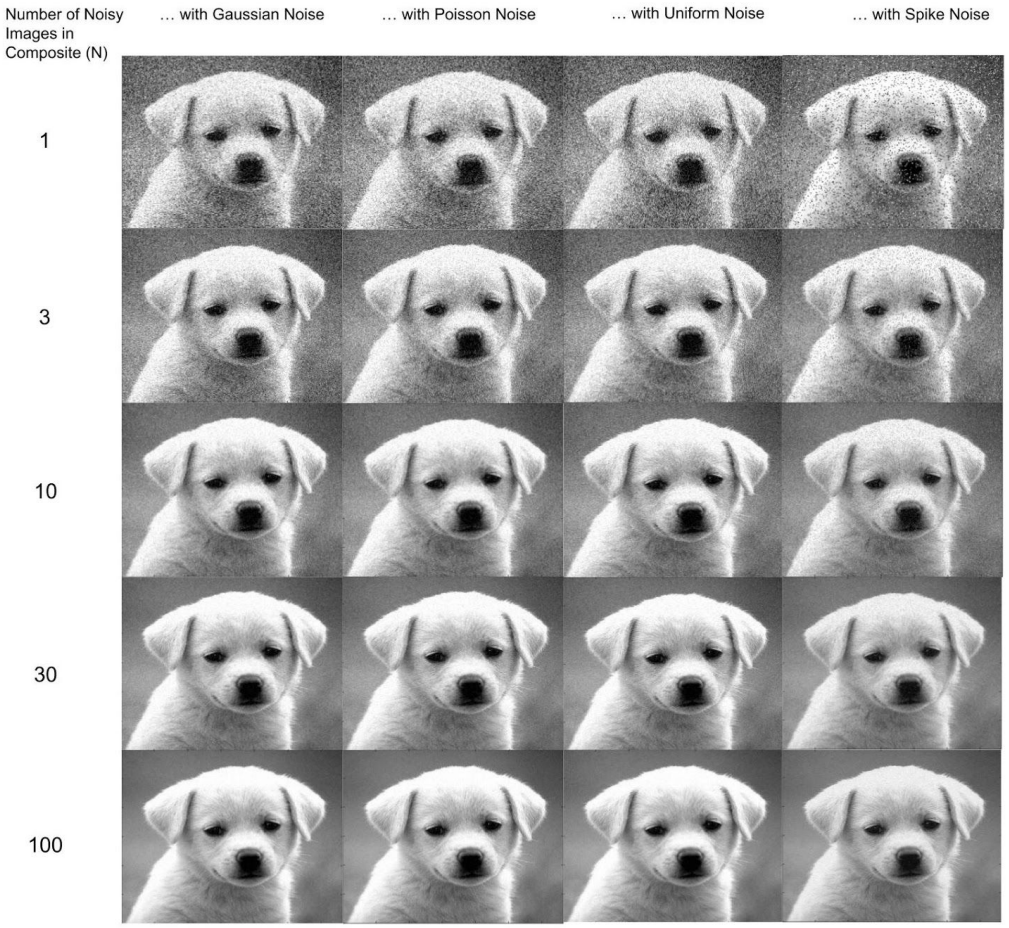
$$\bar{\eta}(x,y) = \bar{g}(x,y) - f(x,y)$$

The second form is necessary to compute the effective noise from the Spike noise composites. The standard deviation of every pixel's noise was determined, and compared to the expected

value of $\frac{\sigma}{\sqrt{n}}$. We do this in order to quantify the clarity of the composite images, and so that we can also relate what we see to the behavior that we expect from sampling distributions.

3. Results and Discussion

3.1 Noise Reduction in Composite Images



8

Figure 3. *The image above shows the effect of the 4 types of noises discussed and the effect of averaging 3, 10, 30, and 100 of such noisy images.*

We notice here that as the sample size averaging increases, the original noisy image of each distribution becomes cleaner than their individual noisy image. Upon the 100 sample size average of each distribution, the Gaussian, Poisson and uniform noises result in an image with a quite similar degree of clarity compared to that of the Spike noise. This is because unlike the other three distributions which have a mean of zero, Spike noise does not necessarily result in a noise distribution with a mean of zero. For this distribution, the magnitude of added noise depends on the brightness or darkness of the original clean image.

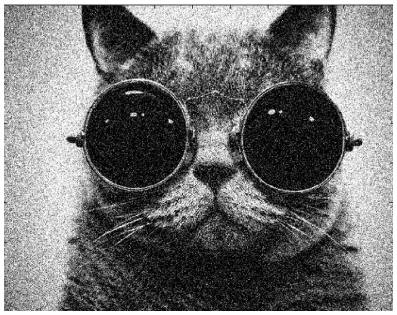


Figure 4. *Gaussian noise added to clean image (N=1).*

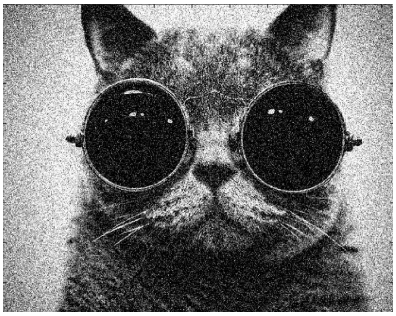


Figure 5. *Poisson noise added to clean image (N=1).*

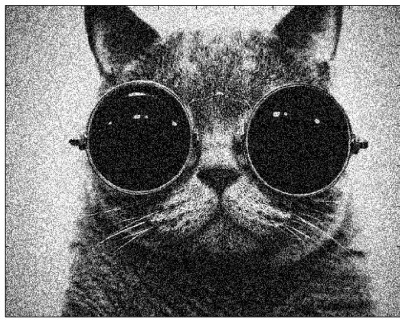


Figure 6. *Uniform noise added to image (N=1).*

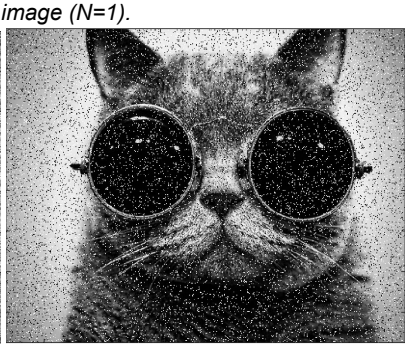


Figure 7. *Spike noise added to clean image (N=1).*

Figure 8. *Composite image (N=100) with Gaussian noise.*

Figure 9. *Composite image (N=100) with Poisson noise.*

Figure 10. *Composite image (N=100) with Uniform noise.*

Figure 11. *Composite image (N=100) with Spike noise.*

When we added noise to the second clean image, we found that averaging still reduced the noise in the same way we expected. For the Gaussian, Poisson, and Uniform noise, the composite image is indistinguishable from the original clean image. One noticeable feature in this image is that it is darker than the first image overall. This allows us to more easily observe the effect of spike noise. The spike noise makes the dark areas brighter here, which can be seen in the cat's sunglasses in Figure 11. Averaging this type of noise actually decreased the contrast of the entire image for this reason.

3.2 Statistical Analysis

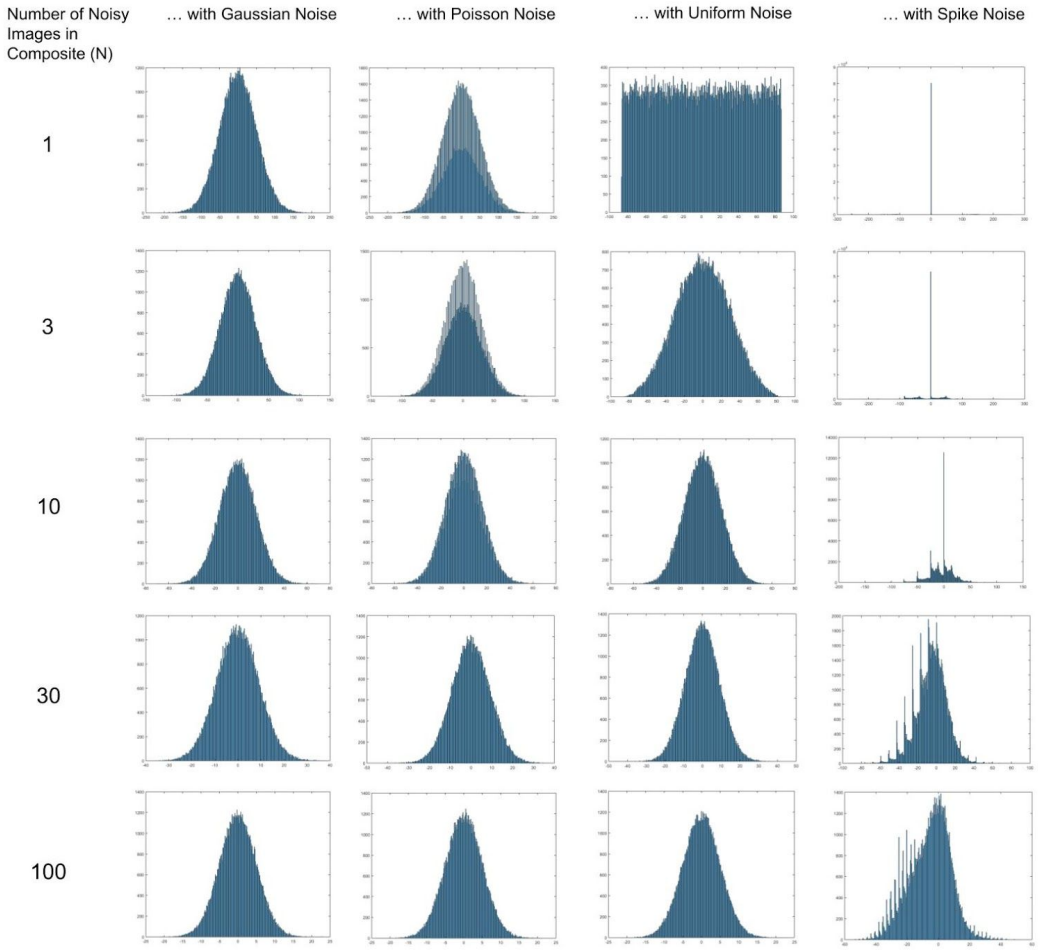


Figure 12: *Histogram plots of each distribution added to the image along with averaged graphs as more images are overlaid. A distinct overall trend emerges: as more histograms are averaged, each distribution tends towards a Gaussian profile.*

The Gaussian distribution, by default, does not converge to anything except itself. Poisson noise also approaches a Gaussian very quickly, because its distribution is inherently similar to the normal. Its convergence is sped up by our condition of zero-mean noise: although the Poisson distribution is always positive, by shifting the entire graph to the left by an amount

equal to the mean, and thus allowing negative values, we ensure that the noise averages itself out. Uniform noise starts flat, as expected, but quickly transforms into a bell curve and by N = 10 is virtually indistinguishable from the previous two distributions. With spike noise, although its histogram is generated via recording probabilities per pixel, it too becomes a bell curve by N = 30, strikingly supporting the Central Limit Theorem. However, there are pseudo-peaks rising from the curve below the mean, which are due to issues of pre-saturation. Many pixels throughout the image initially have values of either 0 or 255, and will not change if the probability code requires that they take on the minimum or maximum value, respectively. Since our initial image has many more white pixels than black, we will primarily observe black pixels in the noise distribution since the converted white pixels will get washed out. This will skew the resultant histogram to the form that we obtained.

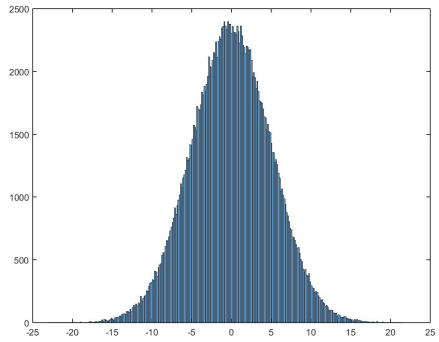


Figure 13. *Sampling distribution of the mean for the cat image with Gaussian noise (N=100).*

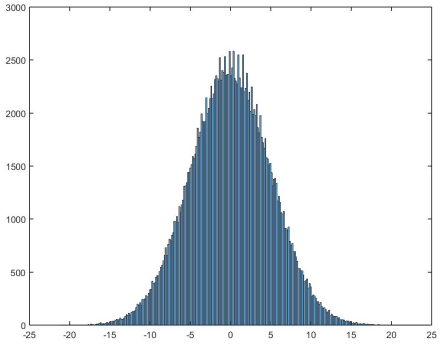


Figure 14. *Sampling distribution of the mean for the cat image with Poisson noise (N=100).*

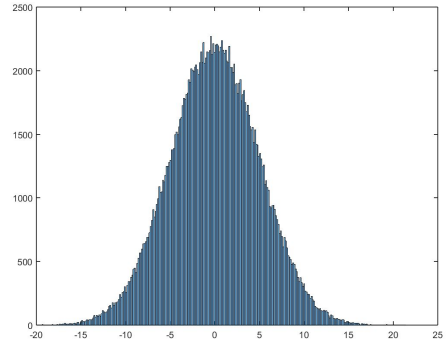


Figure 15. *Sampling distribution of the mean for the cat image with Uniform noise (N=100).*

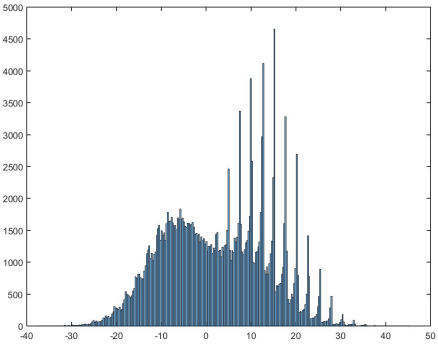


Figure 16. *Sampling distribution of the mean for the cat image with Spike noise (N=100).*

From these charts, we noticed several things. The overall pattern remains the same as with the first image--that with more images contributing to the average, the noise in the composite image is minimized. With Gaussian, Uniform, and Poisson distributions, the noise still tends toward zero. However, the spike noise composite image (at N=100) does not closely

resemble the distribution that we saw from the dog image. The reason for this is related to the different way that we generated the noisy image, that is without explicitly defining $\eta_i(x,y)$. The noise therefore depends a lot on the image. As we can see, the dog image is very light and the cat image is very dark. This explains why the sampling distributions are skewed in opposite directions. The spike noise pulls the lighter image toward lower pixel values, and it pulls the darker image toward higher pixel values. This affects both the mean and the standard deviation of the distributions shown in Figure 12 and Figure 16.

Finally, we compare the observed standard deviations of the sampling distributions of the mean noise, as described in the procedure. We obtained the following results:

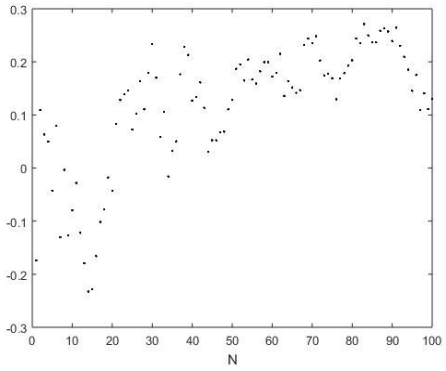


Figure 17. *This plot shows the percent difference between the theoretical standard deviations and the observed standard deviations of the mean noises, from the Gaussian distribution, for varying values of N.*

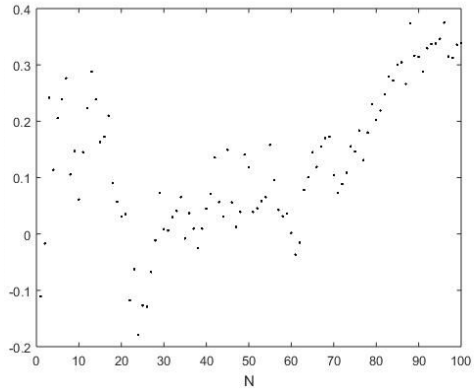


Figure 18. *This plot shows the percent difference between the theoretical standard deviations and the observed standard deviations of the mean noises, from the Uniform distribution, for varying values of N.*

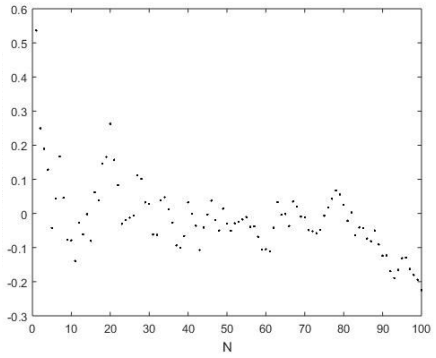


Figure 19. *This plot shows the percent difference between the theoretical standard deviations and the observed standard deviations of the mean noises, from the Poisson distribution, for varying values of N.*

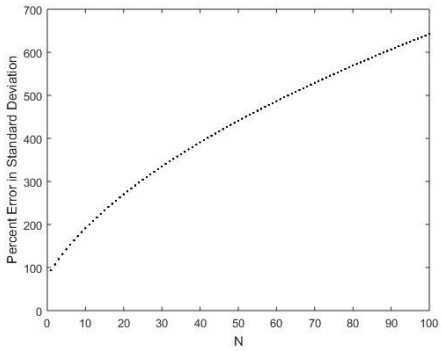


Figure 20. *This plot shows the percent difference between the theoretical standard deviations and the observed standard deviations of the mean noises, from the Spike noise distribution, for varying values of N.*

From these plots, we can extract a few conclusions about the error (essentially the standard deviation) in the mean noises from each of our distributions. In the first three plots, we compare the theoretical standard deviations with the observed deviations in Gaussian, uniform, and Poisson noise. For these three distributions the percent differences between the observed and expected standard deviations are less than ∓1% for all values of N. In other words, they follow the expected trend of $\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$. However, for the spike noise, the error drastically increases to over 600%. This effect supports what we have seen through all of our analysis of spike noise, that this type of noise is entirely different. However, it is interesting that the spike noise can still be reduced by averaging, despite its differences from the other noise distributions.

4. Camera Application

Given the four generated noise distributions we just discussed, digital images produced by our everyday cameras contain all noise distributions. Principal sources of Gaussian noise in digital images arise during acquisition e.g. sensor noise caused by poor illumination and/or high temperature, and/or transmission. Spike noise originates from analog-to-digital converter errors as well as bit errors in transmission. The poisson noise however merely comes from the discrete nature of electric charge.

Since noise, as shown in figure 2 above reduces upon averaging, we experimented such averaging with images from a Samsung SM-N900 back camera. Since these images contain at least the 4 types of noise distributions we studied, we expected a better quality image upon averaging on Matlab. Consider the Figure 21 below;

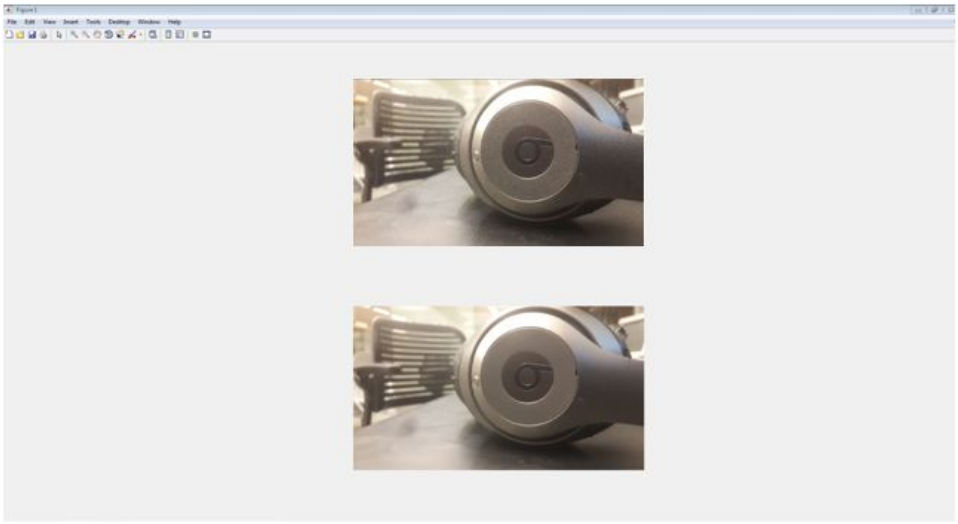


Figure 21. *Image showing one of the 7 camera images at the top, and the averaged image at the bottom*

We can see that the bottom image is clearer than the top image. Averaging did reduce the noise. However, we still do have some noise in the averaged image. This could further be reduced by taking more identical images to be averaged on either Matlab or Adobe Photoshop. In the real world, the noise sources that we have studied cannot be separated. They exist simultaneously, but we can see from this experiment that the quality of the image can still be improved by averaging.

5. Conclusions and Future Work

From these analyses, we concluded that noise, upon averaging, decreases depending on the sample size of the averaged noisy images. Firstly, we observe heavy evidence and confirmation of the Central Limit Theorem. As mentioned above, the theorem posits that no matter what population distributions we deal with, they will converge to a normal distribution. This is an important phenomenon found throughout the field of statistics, and simplifies a great deal of analysis since it eliminates the need for information about the population distribution and reduces the sample distribution to a simple bell curve once the sample size rises beyond thirty. This precisely matches what we found in our study: the averaged histograms of each of the four distributions converge to Gaussian form by the time N increases past 30. This even applies to spike noise, despite its markedly different physical origin and initial form, and is the clearest demonstration of the power of the Central Limit Theorem.

The second key theme, and the one that we initially set out to investigate, is that averaging can be an effective way to reduce noise in digital images. However, this mainly applies to situations where one has many images at their disposal. The mean noise value, which for three of our four cases was zero, remains the same no matter what sample size we use. As our sample size increases, deviation from this mean decreases: we observe this trend in the tightening of each histogram sequence, as more and more values cluster closely around the mean. In theory, the standard deviation should decrease as one over the square root of the sample size. We find that our results are consistent with this principle.

Future improvements could focus on methods to reduce and eliminate noise when only one image is available. This could be done via techniques such as a Fourier or wavelet transform which is an important tool required to decompose an image into its sine and cosine components. Additionally this project can be extended to include the elimination of noise from periodic or systematic sources.

6. References

Cattin, Dr Philippe. "Image restoration: Introduction to signal and image processing." *MIAC, University of Basel. Retrieved* 11 (2013).

Chang-Yanab, Chi, Zhang Ji-Xiana, and Liu Zheng-Juna. "Study On Methods Of Noise Reduction In A Stripped Image." *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B6b. Beijing 2008* (2008).

Devore, Jay, and Nicholas R. Farnum. *Applied Statistics For Engineers And Scientists*. United Kingdom: Duxbury Press, 1999. *Book Review Digest Plus (H.W. Wilson)*. Web. 29 Nov. 2016.

Farooque, Mohd Awais, and Jayant S. Rohankar. "Survey on various noises and techniques for denoising the color image." *International Journal of Application or Innovation in Engineering & Management (IJAIEM)* 2.11 (2013): 217-221.

7. Appendix:

Matlab Script:

```
f = imread('B2DBy.jpg');
f = double(f); % Comes as an unsigned integer, but in order to add h, needs to be a double
mn = size(f(:,:,1));
N = 100; % Number of g_i to sum over
gsum = double(zeros(mn)); % This matrix needs to be same size as our image
dir = ['Uniform_r100_n100']; %rename as needed for each noise distribution
 if(~exist(dir,'dir'))
       mkdir(dir)
   else
       rmdir(dir,'s')
       mkdir(dir)
   end
dist_flag = 1; % This is where we can choose what distribution to study.
                % 0 = Gaussian; 1 = Uniform; 2 = Poisson; 3 = Spike Noise
for i = 1:1:N
   if (dist_flag == 0)
       h(:,:,i) = normrnd(0,50,mn(1),mn(2)); % Generates Gaussian random noise for each i, average value of
0, standard dev 50.
       g(:,:,i) = f(:,:,1) + h(:,:,i);
   elseif (dist_flag == 1)
       h(:,:,i) = -86.6 + 173.2*rand(mn(1),mn(2)); % Generates Uniform random noise, average value of 0,
range of 100.
       g(:,:,i) = f(:,:,1) + h(:,:,i);
   elseif (dist_flag == 2)
       h(:,:,i) = -2500 + poissrnd(2500,mn(1),mn(2)); % Generates Poisson random noise, average value of 0,
lambda = 50.
       g(:,:,i) = f(:,:,1) + h(:,:,i);
   elseif (dist_flag == 3)
       tmp = rand(mn(1), mn(2));
       for y = 1:1:mn(2)       %For each pixel, there is a 10% chance that the pixel will be turned black
          for x = 1:1:mn(1)   % and a 10% chance that the pixel will be turned white. 80% of the time, the pixel
             if (tmp(x,y) < 0.065)
                g(x,y,i) = 0;
             elseif (tmp(x,y) < 0.935)
                g(x,y,i) = f(x,y,1);
             else
                g(x,y,i) = 255;
             end
          end
       end
   end
   gsum = gsum + g(:,:,i);
```

```
    tmp1 = gsum./i;
    tmp1(:,:,2) = tmp1(:,:,1);
    tmp1(:,:,3) = tmp1(:,:,1);
    tmp2 = uint8(tmp1);
    tmp3 = image(tmp2);
    if (i<10)
       saveas(tmp3, [dir,'/img_i_00',int2str(i),'.png'],'png')
    elseif (i<100)
       saveas(tmp3, [dir,'/img_i_0',int2str(i),'.png'],'png')
    else
       saveas(tmp3, [dir,'/img_i_',int2str(i),'.png'],'png')
    end
    %Make Histogram
    tmp4 = tmp1 - f;
    xhi = max(max(tmp4(:,:,1)))*1.5; %calculates an expansive max/min x-value
    tmp5 = histogram(tmp4(:,:,1),300);
    set(gca, 'xlim', [-xhi xhi]);  %hardcodes x-axis limits to stop the histograms jumping around
    if (i<10)
       saveas(tmp5, [dir,'/hist_i_00',int2str(i),'.png'],'png')
    elseif (i<100)
       saveas(tmp5, [dir,'/hist_i_0',int2str(i),'.png'],'png')
    else
       saveas(tmp5, [dir,'/hist_i_',int2str(i),'.png'],'png')
    end
end
```

18

MATH 3320

Group 7: Parker Pennington, Kalli Douma, Shania Bulous

Project's Report Feedback

Required Sections	Maximum Grade	Your grade	My comments
Introduction	15	15	Well written
Statement of the Problem	25	20	This was embedded in the introduction and should have been separated from it
Methodology	25	23	Well explained. There was a little confusion about the notion of outliers.
Results	25	25	Well explained as well. You had a good understanding of the limitation of your own method. Also, the concluding remarks and the recommendation were well done.
Bibliography	10	10	Well done
Other comments			I liked the use of Anderson Darling test for normality. Refrain from making statement without facts
Total	100	**93**	

Project's Presentation Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Be properly attired	5	5	Well done.
Comments of another presentation	35	30	Well done
Chemistry among group members	20	15	Not bad at all
Timely submission of the presentation	20	20	You submitted your milestones on time
Pass all 4 Milestones	20	20	Well done
Other comments			
Total	100	**90**	

Trinity University, San Antonio, TX

Electric Vehicles vs. Gas Vehicles

Pennington, Parker

Math 3320: Probability and Statistics for Engineers and Scientists

Dr. Eddy Kwessi

12/1/2016

Table of Contents

Introduction:

The economic value of a personal vehicle is directly affected by more than just the purchase cost. Gas mileage, expected maintenance cost, and longevity of the vehicle, to name a few, should all be considered when planning a purchase. The most common form of transportation is the automobile. A consumer has two choices when purchasing a car: gasoline powered (Internal Combustion Engine, ICE) or an electric vehicle (EV). In order to try to predict the most economically beneficial vehicle, we decided to analyze statistics relating to the costs of each vehicle's fuel, energy efficiency, and maintenance.

The goal of this project is to help average college students make educated choices in the type of car, gasoline or electric, they should purchase. While electric cars are initially more expensive, there is an opportunity to displace that cost through tax incentives and potentially lower energy costs. However, since ICEs are much cheaper, the initial cost of purchasing an electric car may take a lot of exclusively electric driving to make up for the higher price. This offset may be caused by the difference in fuel prices or maintenance costs**.** Our plan is to resolve these uncertainties in an attempt to explain how different factors affect the value of a car. We hope to be able to definitively answer which kind of vehicle is economically cheaper over the entire span of owning the car.

Methods:

For the average college student, the most pertinent cost after the actual price of a car is fuel cost. Fuel efficiency, miles per gallon for gas cars and mpge (the electric equivalent to miles per gallon) for electric cars, is the most important factor when attempting to identify a cost related to fuel consumption. Because a comparison is being made with college students in mind, the cars chosen were based on four affordable makes of cars: Ford, Chevy, Toyota, and Nissan.

Specifically, the comparison between electric and gasoline power involved the Chevy Volt, Toyota Prius, Nissan Leaf, and Ford Fusion Energi, versus the Chevy Malibu, Toyota Corolla, Nissan Versa, and Ford Fusion, respectively. Though three of the electric cars used to compare are actually hybrids, for the purposes of our comparison, most of the data used for these cars were based on the vehicles while they were in a solely electric engine mode.

To try to achieve the goal of trying to help future college students get their best buy, comparative distributions were created for the different factors relating to fuel consumption. Before these distributions were created, two separate sets of comparative plots were generated for the electric and gas vehicles. For gas vehicles, the three plots generated were mpg vs. make, mpg vs. tank size, and mpg vs. total range. For electric vehicles, three more plots were generated: mpge vs. make, mpge vs. total electric range, and mpge vs. battery size.

Next, we wanted to see if any of the parameters for either car were normally distributed. To do this, Minitab was used to analyze the data. Miles per gallon, mileage range, engine size, and gas tank size were analyzed for the gas vehicles. Electric mileage range and battery size were analyzed for the electric vehicles. In order to analyze normal distribution for gas and electric vehicles, we used the Anderson-Darling normality test and normal distributions. To analyze the normal distributions, we assumed the populations of our small samples were normally distributed.

The third part of our fuel cost analysis compared the cost per mile of the two categories of car. For both gas and electric vehicles, the current national average for fuel cost was used. For gasoline vehicles this was $2.06/gal and for electric vehicles this was $0.11/kWh. Then, by using the MSRP of each make and model, the difference in cost between each electric vehicle and its' gasoline powered complement was determined. However, because electric vehicles have

environmental benefits, there are tax incentives, which are paid back to the consumer. So, the prices of each electric vehicle were adjusted by reducing the cost by each respective tax rebate. The gas tank/battery capacity of each the car was taken and multiplied with the total electric mileage range, resulting with the cost per mile ($/mi). The total difference in initial cost of the vehicle divided by the difference in cost per mile in electric and gas modes, resulted in the total number of miles that would have to be driven in electric mode in order to make up for the more expensive initial cost of the electric car.

In addition to this fuel cost analysis, the percentage saved over five years due to fuel cost was calculated by first gathering data on Nissan, Chevy, Toyota, and Ford electric and gas cars of similar size. This data was gathered from Edmunds True Cost to Own Calculator (Edmunds TCO). The percentage saved over five years for each make was then calculated using (1). To compare the percentage difference, the mean and standard deviation was found using (2) and (3). The mean and standard deviation were used in (4) and (5) to find the lower and upper and boundaries for outliers; any data below the lower boundary or above the upper boundary would be an outlier. This would allow perspective buyers to see how much the percentage difference may change due to different makes and models and information about cost to own in an attempt influence the decision between an electric versus gasoline powered vehicle.

$$Percentage\ Saved\ Over\ 5\ Years = \frac{gas\ cost - electric\ cost}{gas\ cost} * 100 \qquad (1)$$

$$(Mean\ Value\ of\ Sample), \qquad \bar{x} = \sum \frac{x}{N} \qquad (2)$$

$$(Sample\ Standard\ Deviation), \qquad s = \sqrt{\sum \frac{(x - \mathrm{x})^2}{n - 1}} \qquad (3)$$

$$Lower\ boundary = \bar{x} - (s * 1.5) \quad (4)$$

$$Upper\ boundary = \bar{x} + (s * 1.5) \quad (5)$$

The comparative maintenance requirements of the two types of vehicles are also imperative to the car purchase decision. In the argument between gasoline and electric, there are two main points of contention. One side of the argument commented that most of the maintenance costs of a gas vehicle have to do with its engine, which includes replacing spark plugs, oil changes, and filter changes. None of these components are found in electric car engines, so they are assumed to be cheaper. But, electric cars need to be charged using charging stations, which may need repair as well. To create a better picture about which side of the argument held the greatest amount of credibility, the expected maintenance cost over time for both electric and gasoline powered cars was also researched.

Results:

Our group found it necessary to first provide the sets of raw data that we collected in our research for each category of car before showing the different applications and calculations that stemmed from the original data. See Tables 1 and 3 in Appendix A. As explained in the methods, the first step was to create comparisons from the raw data by creating plots of mpg vs. tank size, mpg vs. make, and mpg vs. total range. See Figures 1, 2, and 3 in Appendix B. For electric cars, three plots were generated: mpge vs. make, mpge vs. electric mile range, and mpge vs. battery size. See Figures 4, 5, and 6 in Appendix B. These plots help depict to a prospective buyer that in both cases, there is no single car that preforms best in every category.

Next, to determine if certain data were normally distributed or not, Minitab was used to analyze the data. For gas vehicles, miles per gallon, mileage range, engine size, and gas tank size

were viewed. For electric vehicles, electric mile range and battery size were viewed. For miles per gallon for the gas cars, the Anderson-Darling normality test was plotted in Figure 7 in Appendix B. Based on this plot, miles per gallon for gas cars appears to be normally distributed. Three of the four data points fall within one standard deviation of the mean; all four of the data points fall within two standard deviations of the mean. This procedure was carried out for the other five remaining categories. With only four data points to use, this procedure resulted in an approximately normal distribution for all of the six parameters, which appeared to be a reasonable assumption for all except for electric mileage range. Assuming electric mileage range has a normal distribution results in the distribution Pr(electric mileage range)~N($\mu = 50.25$, σ^2=1250.69). This resulted in a reasonable looking normality plot, but this distribution is not the true distribution of the population's electric mileage range. The mean is not that high, nor is the distribution as spread out as the standard deviation implies. Most electric vehicles or hybrids have values closer to the lowest three values in our data: 19, 22, and 53 miles. However, even with the small sample size of cars, miles per gallon, gas tank size, gas mileage range, gas engine size, and electric battery size could all easily be considered normally distributed.

It wasn't important that each distribution was found to be normal. The goal was to find the distribution that represented each parameter the best, so that when a potential buyer of a vehicle wanted to see how a certain car's mpg, for example, compared to the overall population of cars, they would be able to do that quickly. Finding the true distributions would have been made much easier had data for more cars been researched.

Another significant calculation was the number of miles it would take to offset the initial higher price of an electric vehicle. Information from Tables 2 and 4 in Appendix A were used for this calculation. Table 5 shows the number of miles needed to drive each car in electric mode

exclusively to save the extra amount of money that you spent in buying the higher priced electric car. Note that the Toyota Prius Prime's impressive mpg and poor conversion of electric power to distance driven actually made driving in gas mode more economically efficient than driving in electric mode. However, this project is based around the college student, and while the higher initial cost does not necessarily have to be displaced within the four years of a normal college career, the difference needs to be diminished within a reasonable time frame. If it is assumed that the operating yearly mileage is the same as the national average, 13,476 miles, the amount of time in years to offset the higher initial cost was calculated. See Table 6 in Appendix A. The Chevy Volt is the only car with an achievable time to own a vehicle of just over 16 years. The Nissan and the Ford have timeframes that are both unrealistic amount of times to own a car, let alone driving it in only electric mode for that amount of time.

Next, the percentage money saved over five years was calculated by first finding data on Edmunds TCO. See Tables 1 and 3 in Appendix A. The percentage change was calculated using (1). For Nissan, Chevy, Toyota, and Ford, the percentage changed over 5 years is 46.334%, 54.66%, 46.259%, and 42.437%, respectively. The sample standard deviation, calculated using (3), was found to be 5.157%. The range for outliers, calculated using (4) and (5), was found to be 39.687% for the lower boundary and 55.158% for the upper boundary. Since all the data was within this boundary, there were no outliers.

The significance of no outliers being present out of the percentages saved over five years for these cars is that no car is great at saving the driver a lot of money compared to other cars, but there is no car that is going to break the driver's bank either. The chevy certainly has the best percentage saved, but all of them save the driver money.

Over the entire lifespan of all electric and gas cars, maintenance cost for electric vehicles was about one-third the amount gas cars required.

Discussion:

The biggest selling points for the electric car are that they are better for the environment than gas cars and overall, one could save money in the very long run with cheaper maintenance and fuel costs. This was not the case for any of the four cars we researched. Only if a driver drove the Chevy Volt in electric only mode for over 216,000 miles would that driver come close to breaking even on choosing an electric vehicle rather than a gas vehicle. Not to mention, gasoline powered vehicles are very competitive to the electric vehicles in many of the chosen categories. These results were somewhat surprising as it would be reasonable to assume, with all the attention electric cars are receiving, that the electric cars could drive farther without a fuel fill up and be much more efficient than any of the gasoline cars. However, that was not found.

One of the main focuses of our project has to do with comparing the percentage difference in fuel cost over five years. While a fairly simple idea to try to compare, because there are so many variables attributed to automobiles, it made finding a way to minimize these variables imperative for quality results. The make, model, size, and year of the car all had to be normalized in order to create a baseline to properly compare similar electric and gas cars. Since we didn't get a sample that was completely random and instead chose which cars to take data from, the data gathered about the sample mean, sample standard deviation, and outliers is not completely accurate.

Collection of the data for fuel efficiency and tank range was a straight forward process. However, there are very few fully electric vehicles on the market that are close to the price range

of a normal college student. Because of this, we chose to use hybrid vehicles in the study as well, analyzing them mostly in their electric modes.

The data collected about maintenance requirements for both electric and gasoline powered vehicles, along with the amount saved for the overall operation of these vehicles, implies multiple assumptions, including amount driven, type of vehicle, driving style, geographical location, and preference in the amount of care given to the car. Assuming the an average driver, researched data had to be located all assuming the same things, which was sometimes difficult. The average driving data collected assumed they would be driving within a city (e.g. low mileage) and would be tuning up and caring for their vehicle in accordance with averages expressed in the actual research.

In the results found about displacing extra cost due to an electric car with distance driven in electric mode, we used the national average for mileage per year. It is possible that a college student could drive much less than the national average, thus it would take even longer to see an economical return due to lower fuel costs.

Another difficulty that came with trying to collect data related to electric and gasoline cars is many of the articles written are used to convince the reader one way or another about electric vehicles. When data was found on websites with this bias, the useful information had to be extracted, dismissing the persuasive information and further researching what some sources do not want the media to know.

Finally, lacking a larger amount of data for our plots and distributions revealed incomplete pictures of how some of the parameters are realistically distributed. A more complete study would include all electric only vehicles, hybrids, and gas only vehicles produced after a certain year and sold to the consumer under a certain price. A lack of time prevented this kind of

data from being collected. Therefore, our lack data leaves our conclusions vulnerable to possible criticisms. Despite our incomplete amount of data, we are confident in the conclusions.

Conclusion:

Claims have been made that the moral obligation of a cleaner car and the lower cost of electricity to gasoline would offset the consistently higher price of an electric vehicle. While the moral incentive to buy an electric car could possibly be quantified to theoretically cover some of the difference in cost, we chose to not include moral incentive as a factor in our research. The only factors took into account were those related to price: fuel efficiency, maintenance costs, and money saved over time. In our analysis of the data, with the college student in mind, the results did not align themselves with what was expected.

Our results showed that gasoline cars and electric cars preform very similarly, a highly unanticipated result. But, the best decision to save the most amount of money is to buy a low cost gasoline powered car with high gas mileage. The extra cost of purchasing an electric car will not be made up. In the future, for electric cars to ever have a reasonable chance of saving the consumer enough money to buy the initially higher priced electric car, any of the following would have to occur: the gas price would have in increase, the electricity price would have to decrease, the mpge would have to increase, the electric only mileage range would have to increase, or the price of electric cars would have to decrease. Until that happens, the gas vehicle will be more economically efficient to the driver.

Appendix A:

Table 1: Gasoline Powered Raw Data

Make	Model	Engine Size	MPG	Total Range	Number of passengers	Cost of fuel over 5 years ($)
Toyota	Corolla	1.8 L	32	422 mi	5	5,320
Nissan	Versa	1.6 L	35	378 mi	5	4,964
Ford	Fusion	1.5 L	30	498 mi	5	5,957
Chevy	Malibu	1.5L	32	416 mi	5	7,168

Table 2: Gasoline Car Data

Make	MSRP ($)	Tank Capacity
Toyota Corolla	17,300	13.2 gal
Nissan Leaf	11,990	10.8 gal
Chevy Volt	21,625	13.0 gal
Ford Fusion Energi	22,110	16.6 gal

Table 3: Electric Powered Raw Data

Make	Model	Battery Size	MPGe	Total Range	Number of passengers	Cost of fuel over 5 years ($)
Toyota	Prius Prime	8.8 kWh	120	640 mi	4	2,859
Nissan	Leaf	18.4 kWh	106	107 mi	5	2,664
Ford	Fusion Energi	7 kWh	88	420 mi	5	3,429
Chevy	Volt	30 kWh	112	550 mi	5	3,250

Table 4: Electric Car Data

Make	MSRP ($)	Tax Rebate ($)	Battery Capacity	Total Electric Range
Toyota Prius Prime	27,100	4,502	8.8 kWh	22 mi
Nissan Leaf	29,010	7,500	30 kWh	107 mi
Chevy Volt	33,170	7,500	18.4 kWh	53 mi
Ford Fusion Energi	33,900	4,007	7 kWh	19 mi

Table 5: Miles to Offset Cost

Make	Miles to Offset Cost
Toyota	N/A
Nissan	607,857 mi
Chevy	216,310 mi
Ford	568,933 mi

Table 6: Time to Offset Cost:

Make	Years to Offset Cost
Toyota	N/A
Nissan	45.11 years
Chevy	16.05 years
Ford	42.22 years

Appendix B:

Figure 1: MPG vs. Make

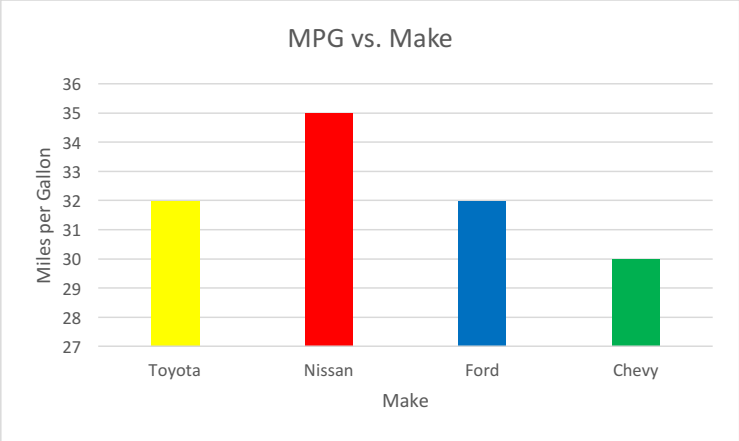


Figure 2: MPG vs Tank Size

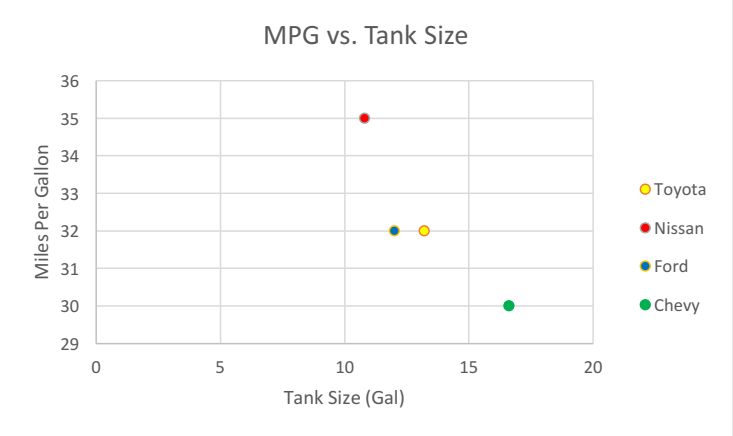


Figure 3: MPG vs. Total Range

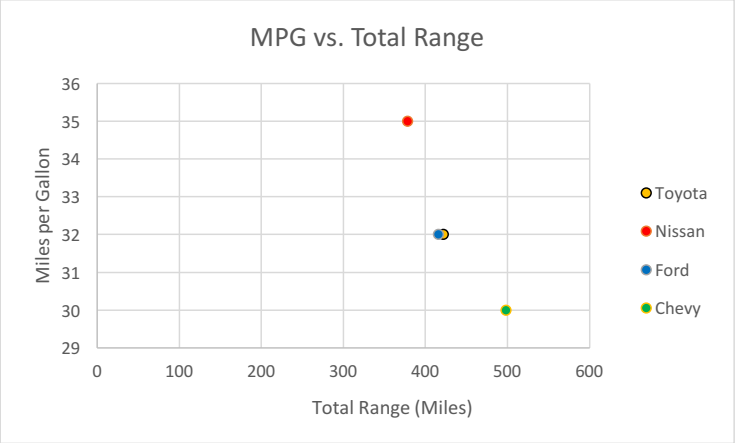


Figure 4: MPGe vs. Make

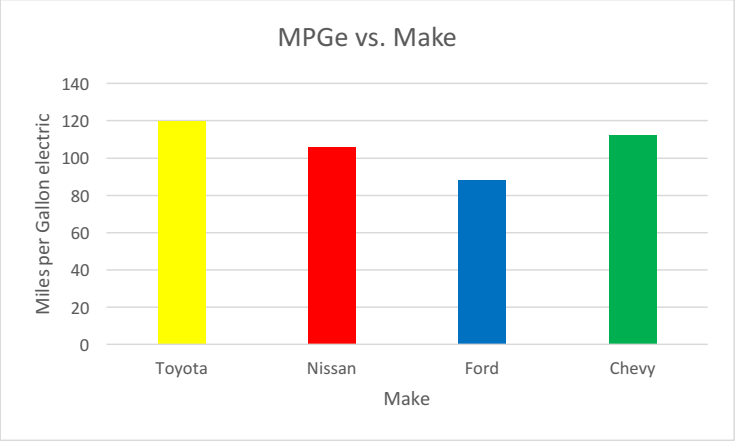


Figure 5: MPGe vs. Electric Mile Range

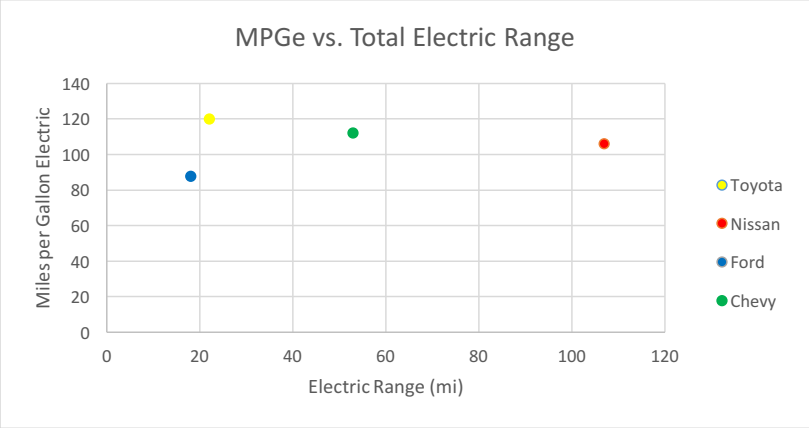


Figure 6: MPGe vs. Battery Size

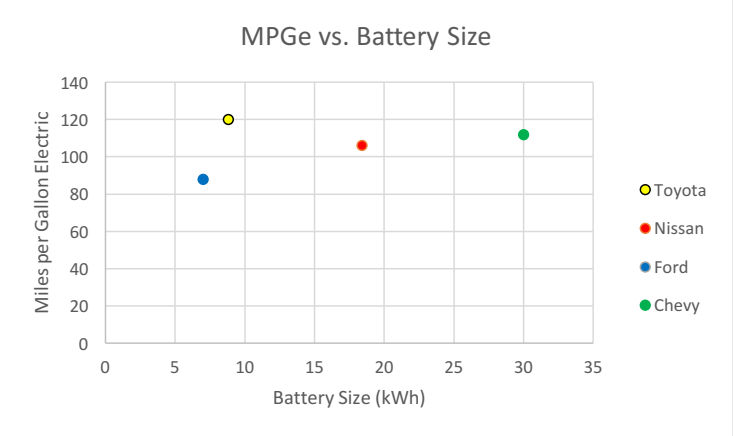
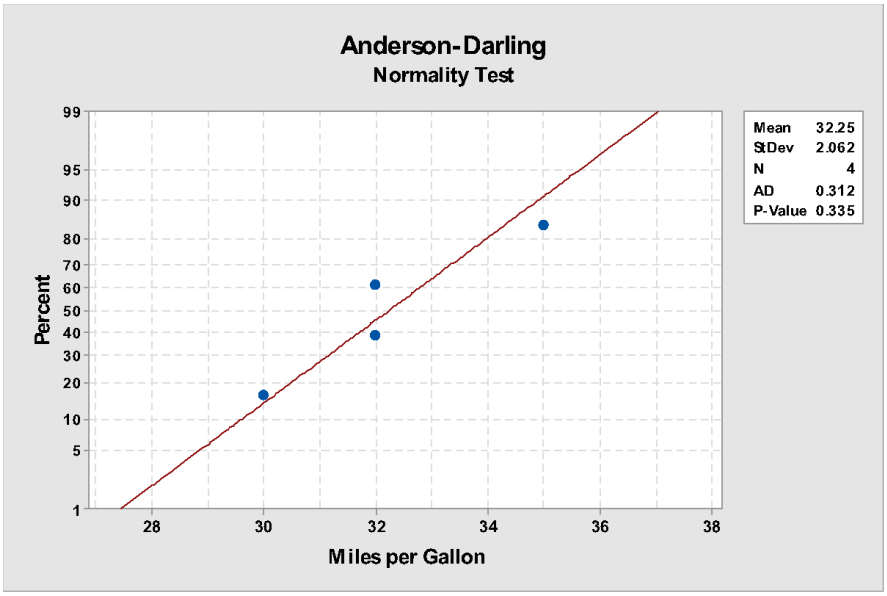


Figure 7: Normality Test Miles per Gallon



Works Cited

"Chevy Volt Review." *Plugin Cars.* http://www.plugincars.com/chevrolet-volt. Accessed 10 Nov. 2016.

"Ford Fusion Energi Review." *Plugin Cars*. http://www.plugincars.com/ford-fusion-energi. Accessed 10 Nov. 2016.

"Fueling a Better Future: The Many Benefits of Half the Oil." *Union of Concerned Scientists,* June 2013. http://www.ucsusa.org/clean-vehicles/fuel-efficiency/benefits-of-reducing-us-oil-use#.WBGo7C0rKpp. Accessed 29 Sep. 2016.

"Gasoline vs. Electric: Cost to Drive 27 Miles." *Union of Concerned Scientists,* June 2013. http://www.ucsusa.org/sites/default/files/legacy/assets/images/cv/Chart-Gasoline-vs-Electric-Fuel-Costs_Full-Size.jpg. Accessed 29 Sep. 2016.

"Nissan Leaf Review." *Plugin Cars.* http://www.plugincars.com/nissan-leaf. Accessed 10 Nov. 2016.

"Tesla model S Review." *Plugin Cars.* http://www.plugincars.com/tesla-model-s. Accessed 10 Nov. 2016.

"Toyota Prius Plug-in Hybrid (Prime) Review." *Plugin Cars.* http://www.plugincars.com/toyota-prius-plugin-hybrid. Accessed 10 Nov. 2016.

"True Cost to Own." *Edmunds,* edmunds.com/tco.htm. Accessed 10 Nov. 2016.

Berman, Brad. "Total Cost of Ownership of an Electric Car." *Plugin Cars,* 8 Nov. 2016, http://www.plugincars.com/eight-factors-determining-total-cost-ownership-electric-car-127528.html. Accessed 10 Nov. 2016.

Goreham, John. "Myth Busted: Electric Vehicles Cost More to Maintain than Gas Cars Do." *Torque News,* 31 March 2014. http://www.torquenews.com/1083/myth-busted-electric-vehicles-cost-more-maintain-gas-cars-do. Accessed 29 Sep. 2016.

Herron, David. " Electric Cars are Cheaper to Maintain." *The Long Tail Pipe,* 23 June 2015. https://longtailpipe.com/ebooks/green-transportation-guide-buying-owning-charging-plug-in-vehicles-of-all-kinds/electric-cars-arent-too-expensive-you-can-own-one-for-free/electric-car-ownership-economics/electric-cars-are-cheaper-to-maintain-no-oil-to-change-no-gaskets-to-replace-etc/.

Herron, David. "Electric Cars are Cheaper to Own." *The Long Tail Pipe,* 23 June 2016, https://longtailpipe.com/ebooks/green-transportation-guide-buying-owning-charging-plug-in-vehicles-of-all-kinds/electric-cars-arent-too-expensive-you-can-own-one-for-free/electric-car-ownership-economics/. Accessed 10 Nov. 2016.

Hovis, Mark. "EV vs ICE Maintenance the First 100000 Miles." *Inside EVs,* http://insideevs.com/ev-vs-ice-maintenance-the-first-100000-miles/. Accessed 29 Sep. 2016.

Pagerit, S. "Fuel Economy Sensitivity to Vehicle Mass for Advanced Vehicle Powertrains." *Autonomie.* January 2016, http://www.autonomie.net/docs/6%20-%20Papers/Light%20duty/fuel_econom_sensitivity.pdf. Accessed 10 Nov. 2016.

Seredynski, Paul. *Edmunds,* 7 Sept. 2016. http://www.edmunds.com/fuel-economy/decoding-electric-car-mpg.html. Accessed 10 Nov. 2016.

MATH 3320

Group 8: Kirby Smith, Laura Wilson, Reece Arawaka

Project's Report Feedback

Required Sections	Maximum Grade	Your grade	My comments
Introduction	15	15	Well written
Statement of the Problem	25	25	Well done
Methodology	25	20	Well explained. The only drawback was its lack of sophistication. You could have fitted distributions into your histograms and make your project even more interesting
Results	25	25	Well explained as well. You had a good understanding of the limitation of your own method. Also, the concluding remarks and the recommendation were well done.
Bibliography	10	10	Well done
Other comments			You lacked a cover page for you report. I acknowledge the tedious data collection's process
Total	100	**95**	

Project's Presentation Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Be properly attired	5	5	Well done. Adequate for the subject chosen
Comments of another presentation	35	30	Well done
Chemistry among group members	20	15	You need an improvement there
Timely submission of the presentation	20	20	You submitted your milestones on time
Pass all 4 Milestones	20	20	Well done
Other comments			There was lack of proper editing, but I liked the transitions
Total	100	**90**	

Kirby Smith
Laura Wilson
Reece Arakawa
MATH 3320
December 1, 2016

Diving in the Game of Soccer

Introduction

Diving in soccer is a controversial topic among players, coaches, fans, and officials. Soccer is known for players' exaggerations and entirely false collisions and injuries that happen all too frequently in some matches. Players are motivated to flop in order to regain possession of the ball, run down the clock, or even gain penalty kicks. In fact, flopping is sometimes encouraged by coaches or others since it has the potential of being a game-winning move. Who could forget the infamous dive by Arjen Robben that resulted in a penalty on Mexico and then an arguably unearned win by the Netherlands in the 2014 World Cup? Flopping has become a part of the sport that cannot be ignored and has the power to change the course of a game.

We anticipate that there will be some type of correlation between flopping culture and the guilty team's position on the field, their score, the offender's nationality, or how much time has passed in the game. We have chosen to look at European Premiere League games over a eight-year period from 2009-2016. For this report, we defined a flop as a player falling or going down when there was no contact initiated by either player or an over exaggeration of contact or an injury. In order to introduce as little bias as possible, we paid attention only to instances of flopping officially penalized by a yellow card.

The Wall Street Journal produced some elementary statistics on flopping and which teams were guiltiest. They only took into consideration diving incidents that resulted in a player being substituted. They counted total injuries and total game time "wasted" by these incidents. Also, many argue that South American, Spanish, and Italian players are the most likely players to dive and suggest that it is a cultural phenomenon. Our objective is to take closer look at when each foul/flop occurs in the game and its relationship to field position, player nationality, and score at the time.

Problem Statement

Most internet and even newspaper articles that address the issue of simulation are based on very shallow and small sets of information. We aim to take an in-depth look at simulation incidents in their full context to identify possible distributions.

Methodology

We began collecting data by watching about five or six world cup soccer matches and looking for when players flopped. It is mostly clear when a player dives in a game, however many of these dives go unnoticed by the referee or the wrong call is made and a penalty is given to the player who seemed guiltiest of committing what looked to be a foul. As we watched matches, we

took note of each incident that occurred which we believed to be a dive. We were able to obtain a good amount of data using this method, but we realized that this was introducing a lot of bias. Since most of our data was not confirmed with a corresponding call from the referee, we decided it would be best to discard the data and start over: this time only taking into consideration the dives that were officially called by the referees.

This realization led us to the method we ended up resorting to in order to find usable data points. After some brief research, we decided to focus our attention on the European Premiere League. This league seemed to have a decent amount of yellow cards being awarded to players for flopping. Data collection proved to be a very tedious task. The most effective way to find incidents where players were yellow carded for diving was to read live commentaries from each match. We used websites like ESPN or BBC as a starting point for the majority of our data collection. The live commentaries included each yellow card awarded during the game, the reason for the penalty, and the time at which it occurred. Using context clues we could find what the score of the game was at the time of the dive, and after some further research we could find where the player was on the field.

In addition to using these websites, we also obtained information from some Premier League teams' websites. These websites typically also had live commentaries, and some had more convenient charts for finding yellow carding incidents and where they happened. By researching the statistics for each match played by that particular team, we could extract information regarding these yellow carding incidents a little bit more efficiently than our previous methods.

After obtaining 40 diving instances over the years of 2011-2016, we were able to compile our data into a few histograms to show where the statistical trends exist. We used Minitab to create four histograms and one dot plot to best illustrate the trends in the data. The first histogram we are including shows the dives with respect to the guilty player's position on the field. We studied the relationships between diving and the nationality of the guilty player and the time elapsed in the game. We also created a histogram showing dives with respect to whether the guilty player's team was losing, tied with the other team, or winning.

Results

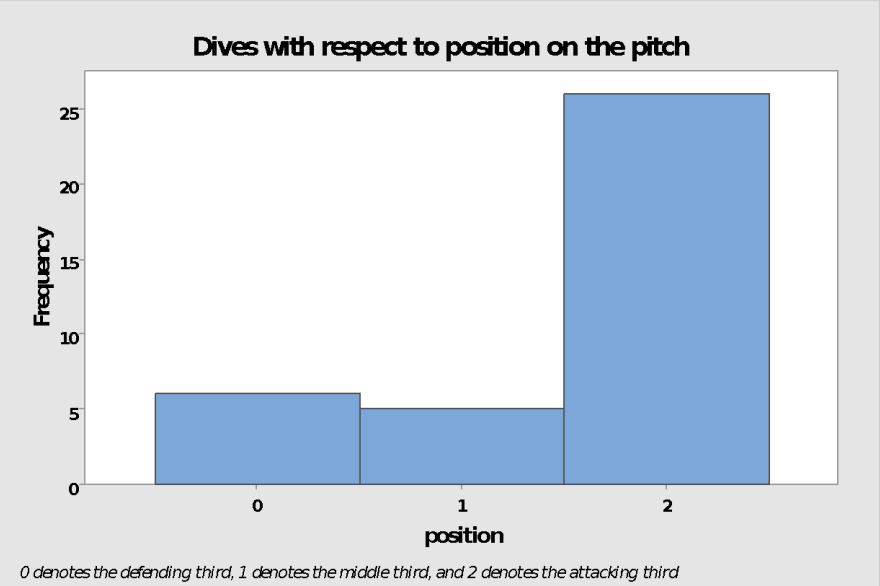


Figure 1: The frequency of dives and flops that were given a yellow card according to the player's position on the pitch.

It appears that players are much more likely to dive when they are in their own attacking third. Drawing a free kick or penalty that close to the opponent's goal is an excellent scoring opportunity. In the histogram shown above, position 0 refers to the defending third of the field or in other words, the section of the field including the goalie of the guilty team. Position 1 refers to the middle third of the court or midfield. Position 2 denotes the attacking third of the field or the section of the field including the other team's goalie. Of the 37 dives that were recorded, 27 were performed in the player's attacking third. From the sample taken, we determined that there is a 73% chance that players will dive when in the attacking third of the pitch as opposed to the middle or defending third. This is significantly higher than the probabilities for the middle third, 13.5%, and defending third, 16.2%. This follows popular belief that players are more likely to dive and receive a yellow card for doing so when in the attacking third. We can also conclude that a player is either less likely to dive or receive a yellow card for diving when in their own defending third, perhaps because of the high-stakes and overall vulnerability of their goal and defense.

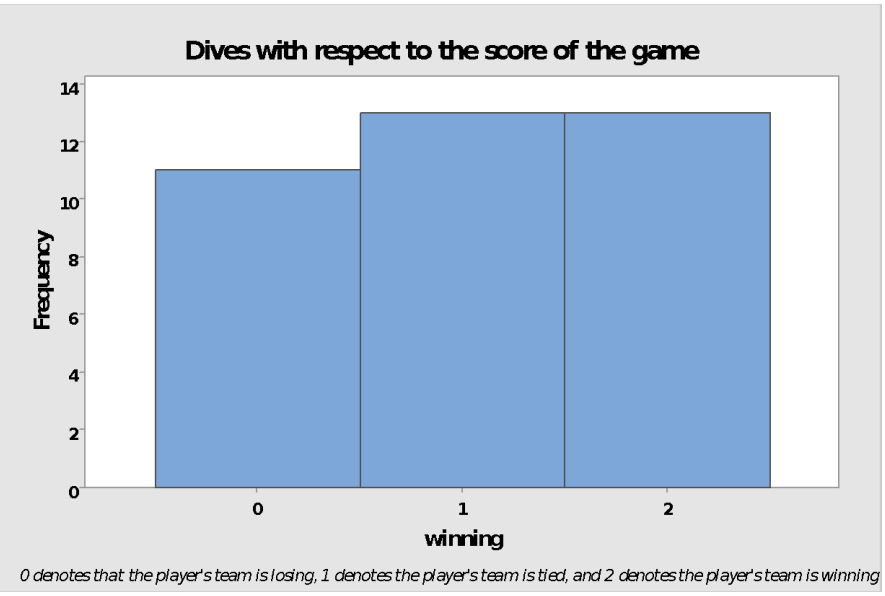


Figure 2: The frequency of dives and flops that received a yellow card with respect to the score of the game. The score of the game is relative to the diving player's team.

There doesn't appear to be any remarkable tendency for players to dive depending on the score of the game. From the sample size used, we could only determine a 6% difference between the probability that a player will dive or flop when they are winning compared to when they are losing. In our data, this was not a significantly important comparison to make, but among a larger data sample, one may be able to identify a more concrete trend.

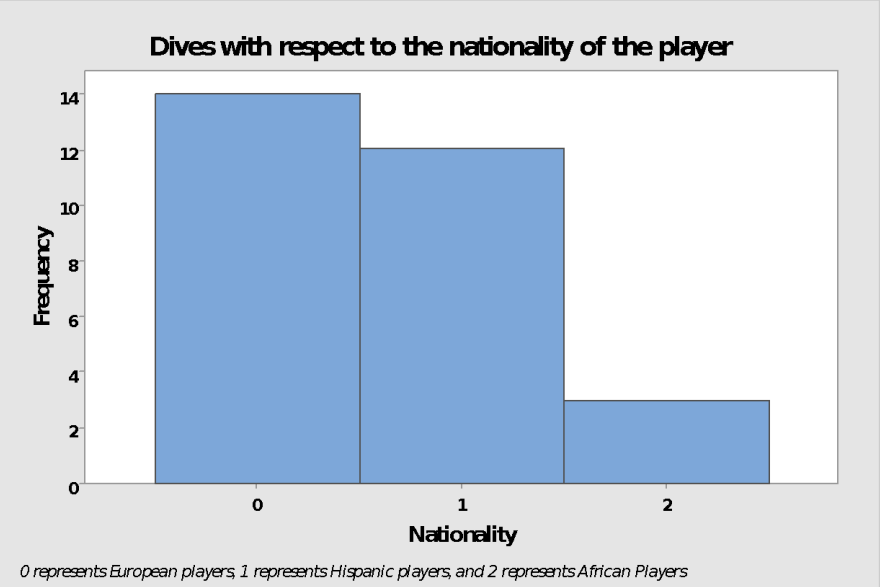


Figure 3: The frequency of players receiving a yellow card for diving with respect to their nationality. For the sake of this histogram, Spanish players were grouped with Americans (Americans including North, Central, and South America) because the stereotype follows that Spanish and South American players dive more often. It is important to note that repeat offenders were not included.

The notion that Spanish and South, North, and Central American players flop more than others isn't entirely correct. For our sample size, we found that Europeans were responsible for 49% of the dives that received yellow cards, Hispanics were responsible for 43%, and Africans were responsible for 8%. It is important to note that these statistics were taken after including repeat offenders. There are, however, far more European players than Hispanic players in the EPL. Out of the 626 players that played in the EPL during the 2014-2015 season, we found that 73.3% were European, 14.5% were Hispanic, and 7.6% were African. Although Hispanics represented 14.3% of the entire player population in the EPL, they were responsible for 43% of the dives within our sample size. We can conclude from the raw data that in the EPL, a yellow card for diving is most likely given to a European player. However, we can conclude that for equal European and Hispanic players, Hispanic players are more likely to dive or receive a yellow card for diving.

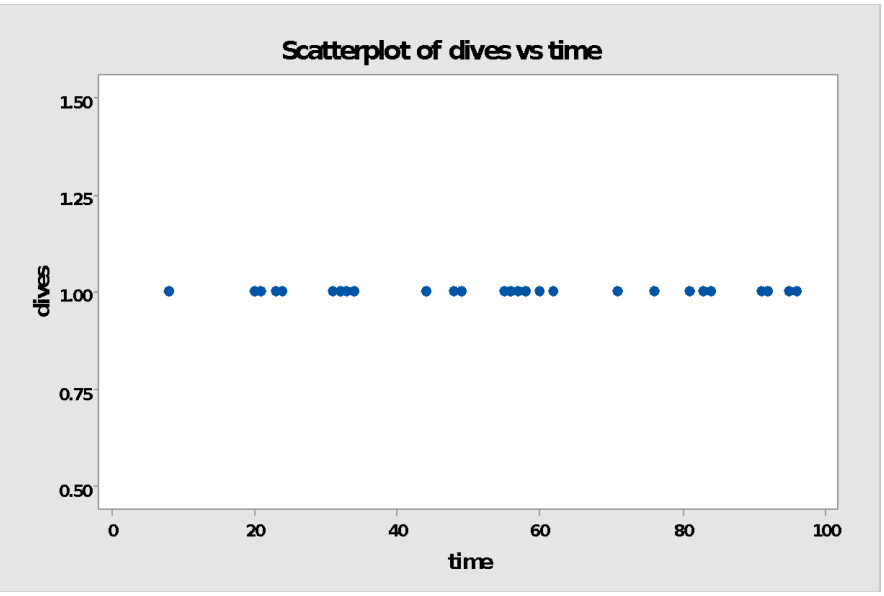


Figure 4: Scatterplot showing the distribution of dives for each player according to the time elapsed in the game. This represents a timeline.

From the scatter plot above, it is reasonable to conclude that the smallest probability of a player receiving a yellow card for diving occurs within the first 20 minutes of the game. This could be due to the overall lack of aggression as games start, or that players aren't as tired and don't take time to rest when they dive or flop. At the beginning of a game, there is less pressure to deliver. In players' minds, they have plenty of time to score or make a comeback but as the game starts to wrap up, the pressure builds and they get more desperate. A reasonable conclusion to this would be that when the pressure to score is on, players will try whatever it takes to make things go their way. The data may be a reflection of referees as well. Referees may be more reluctant to give yellow cards for diving early in games so that players do not need to play the majority of the game with a yellow card for a non-malicious offense.

Figure 5: Frequency of dives with respect to the time elapsed in the game. Rather than examining each dive occurring at a specific time, we grouped each dive according to categories. Category 1 represents the beginning of the game, or 0-29 minutes elapsed. Category 2 represents the middle of the game, or 30-59 minutes. Category 3 represents the end of the game, or 60-90^{+} minutes

From the data organized in the histogram, it is reasonable to conclude either that a player has a higher probability of receiving a yellow card for a dive in the middle of the game or that there is a higher probability that a player dives in the middle of the game. We found that there is a 43.2% probability that players will receive a yellow card for diving in the middle of the game as compared to 21.6% and 35.2% probability for flopping in the beginning or the end of the game, respectively.

$$\Pr(x) = \frac{n}{N} \qquad (1)$$

Where $\Pr(x_i)$ represents the probability that a player dives during a given time division. n_i represents the number of dives within the given time division and N represents the number of dives throughout all time divisions.

Faults in our data may come from the fact that it is not entirely clear which third the dive occurred in for every game. Adjacent comments usually offer strong hints to the direction of play; however, not all real-time commentary sources provide substantial detail. Examining footage from each match during the relevant minutes is an intense process, but solves this issue.

The accuracy of the trends we found could be increased greatly if a larger sample could be obtained. A possible solution would be to write a Linux script to sift through the tens of

thousands of publicly available match commentaries. It would flag commentaries that feature combinations of keywords including “simulation”, “oversimulation”, “dive”, “booking”,“yellow card”. The script would compensate for the most time-consuming part of data collection: identifying games in which yellow cards were awarded for simulation. Other factors that could be examined are player age, player position, total number of yellow cards, referees, and home versus away matches.

Bibliography:

"Premier League Results." *BBC Sport*. N.p., n.d. Web. <http://www.bbc.com/sport/football/premier-league/results>.

"English Premier League Scores & Fixtures." *US Edition*. ESPN FC, n.d. Web. <http://www.espnfc.us/english-premier-league/23/scores>.

"FA Premier League 2014/2015 Nationalities." *Football-Lineups*. Football-lineups.com, n.d. Web. 02 Dec. 2016.

Foster, Geoff. "The World Cup Flopping Ranking." *The Wall Street Journal*. N.p., 27 June 2014. Web. 01 Dec. 2016. <http://www.wsj.com/articles/the-world-rankings-of-flopping-1403660175>.

SportsMole. "Premier League." *Premier League - Football Rumours, Gossip, Transfer News and Results - Sports Mole*. N.p., n.d. Web. <http://www.sportsmole.co.uk/football/premier-league/>.

"Tottenham Hotspur Fixtures." *Tottenham Hotspur Fixtures and Results | Match Centre*. N.p., n.d. Web. <http://www.tottenhamhotspur.com/matches/>.

Appendix A

Match	Offending Team	Score	Score	Time	Position	Official Call	Player	Nationality
2012 Chelsea Stoke City	Chelsea	Chelsea 0	Stoke City 0	49	A	unsporting behaviour	Oscar	Brazilian
2012 Chelsea Arsenal	Chelsea	Chelsea 1	Arsenal 1	48	D	unsporting behaviour	David Luiz	Brazilian
2012 Chelsea Tottenham Hotspurs	Chelsea	Chelsea 1	Tottenham 0	24	M	unsporting behaviour	Ivanovic	Serbian
2012 Chelsea Manchester United	Manchester	Chelsea 2	Manchester 3	91	A	unsporting behaviour	Valencia	Ecuadorean
2012 Chelsea Norwich City	Norwich	Chelsea 1	Norwich City 0	81	A	diving	Bradley Johnson	English
2012 Chelsea Reading	Chelsea	Chelsea 2	Reading 2	96	D	diving	Azpilicueta	Spanish
2012 Chelsea Fulham	Fulham	Chelsea 2	Fulham 0	44	M	unsporting behaviour	Bryan Ruiz	Costa Rican
2013 Chelsea Tottenham Hotspurs	Tottenham	Chelsea 0	Tottenham 1	44	M	diving	Andros Townsend	English
2013 Chelsea Southampton	Chelsea	Chelsea 0	Southampton 1	33	D	diving	Michael Essien	Ghanian
2016 Chelsea Swansea City	Chelsea	Chelsea 0	Swansea 1	71	A	diving	Pedro	Spanish
2014 Chelsea Burnley	Chelsea	Chelsea 2	Burnley 1	31	A	simulation	Diego Costa	Brazilian
2014 Chelsea Hull City	Chelsea	Chelsea 1	Hull City 0	58	A	diving	Diego Costa	Brazilian
2014 Chelsea Southampton	Chelsea	Chelsea 1	Southampton 1	55	A	diving	Fabregas	Spanish
2015 Chelsea Manchester United	Manchester	Chelsea 1	Manchester 0	95	D	simulation	Ander Herrera	Spanish
2015 Chelsea Arsenal	Chelsea	Chelsea 0	Arsenal 0	23	M	simulation	Fabregas	Spanish
2015 Chelsea Southampton	Southampton	Chelsea 1	Southampton 0	33	A	diving	Sadio Mane	Senegalese
2015 Chelsea Southampton	Chelsea	Chelsea 1	Southampton 1	57	D	diving	Falcao	Colombian
2016 Chelsea Swansea City	Chelsea	Chelsea 0	Swansea 1	71	D	diving	Pedro	Spanish
2009 Manchester United v Arsenal	Arsenal	Manchester 2	Arsenal 1	71	A	diving	Eboue	Ivory Coast
2012 Tottenham Liverpool	Tottenham	Tottenham 2	Liverpool 1	76	A	simulation	Gareth Bale	Welsh
2012 Tottenham Fulham	Tottenham	Tottenham 0	Fulham 0	23	A	diving	Gareth Bale	Welsh
2013 Tottenham Sunderland	Tottenham	Tottenham 2	Sunderland 1	80	A	diving	Gareth Bale	Welsh
2013 Tottenham Sunderland	Tottenham	Tottenham	Sunderland	21	A	simulation	Gareth Bale	Welsh
2013 Tottenham Chelsea	Tottenham	Tottenham 1	Chelsea 0	44	M	diving	Townsend	English
2013 Tottenham Manchester Unite	Manchester U	Tottenham 2	Man U 1	89	A	simulation	Adnan Januzaj	Belgian
2014 Tottenham West Brom	Tottenham	Tottenham 1	West Brom 3	55	A	diving	Danny Rose	English
2016 Tottenham Stoke City	Stoke City	Tottenham 0	Stoke City 0	34	A	simulation	Marko Arnautovic	Austrian
2013 Tottenham Everton	Tottenham	Tottenham 1	Everton 2	84	A	simulation	Clint Dempsey	American
2013 Tottenham Sunderland	Tottenham	Tottenham 0	Sunderland 0	20	A	simulation	Gareth Bale	Welsh
2015 Swansea Tottenham	Tottenham	Tottenham 0	Swansea 0	83	A	simulation	Jan Vertonghen	Belgian
2015 Tottenham Leicester City	Tottenham	Tottenham 2	Leicester City 0	32	A	simulation	Nacer Chadli	Belgian
2015 Liverpool Tottenham	Liverpool	Tottenham 0	Liverpool 0	24	A	simulation	Phillipe Coutinho	Brazilian
2015 Tottenham Manchester	Tottenham	Tottenham 3	Manchester 0	92	A	simulation	Dele Alli	English
2014 Manchester United Sunderlan	Man U	Man U 1	Sunderland 0	62	A	diving	Ashley Young	English
2015 Arsenal Chelsea	Arsenal	Arsenal 0	Chelsea 0	23	A	diving	Cesc Fabregas	Spanish
2014 Arsenal Sunderland	Arsenal	Arsenal 1	Sunderland 0	49	A	diving	Danny Welbeck	English
2014 Chelsea Hull city	Chelsea	Chelsea 1	Hull City 0	60	A	diving	Gary Cahill	English

Appendix A: Refined Data

We originally examined World Cup matches, but ignored those data points when our data collection method changed. Only the 37 Premier League incidents above were considered to create histograms, etc.

MATH 3320

Group 9: Jacob Hudson, David Kramer

Project's Report Feedback

Required Sections	Maximum Grade	Your grade	My comments
Introduction	15	15	Well written
Statement of the Problem	25	25	Well done
Methodology	25	20	Well explained. You cover all the key points in your statistical analysis except for the residual analysis
Results	25	25	Well explained as well. Your findings are somewhat surprising and contradicts the common belief.
Bibliography	10	10	Well done
Other comments			You lacked a cover page for you report.
Total	100	**95**	

Project's Presentation Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Be properly attired	5	5	Well done. Adequate for the subject chosen
Comments of another presentation	35	30	Well done
Chemistry among group members	20	20	Good coordination but lack of passion and energy
Timely submission of the presentation	20	20	You submitted your milestones on time
Pass all 4 Milestones	20	20	Well done
Other comments			Overall very good job.
Total	100	**95**	

Jacob Hudson and David Kramer
Math 3360 Project Report
Faculty Advisor: Dr. Kwessi
December 1, 2016

Determination of the More Important Performance Aspect for Par Four Scoring Between Driving and Putting

Introduction:

Golfers have long debated the key to low scores, and among skills such as short game, iron play, and recovery, driving and putting are the most attractive answers. Between these two, most have reached the consensus that driving is more important, even to the extent of the cliché that one "drives for show and putts for dough." Because golfers only have so much time to practice different areas of their games in order to maximize their success, and because so many people are want for a clear answer, and because we ourselves are golfers and golf enthusiasts, we used several techniques in statistical analysis to determine whether driving or putting is more important for good scoring.

Problem Statement:

Driving is the first shot taken on par fours and par fives, and it is beneficial to hit long and accurate drives since shots can result in penalty strokes or awkward shots from the trees. But putting ultimately determines score, and, though frustrating, a two foot putt counts for just as many strokes as any other shot. Perhaps it is the notion that putting determines the final score on a hole that leads people to believe it is the most important shot.

The problem in determining if this is true lies in the absence of clear boundaries and clear statistics from which meaningful conclusions can be drawn. For example, the quality of a golfer's drives and other shots has an immediate impact on the number of putts that the player needs to hit once on the green. So for our question, a statistic like "putts per round" gives us no information, as poor drives lead to difficult approach shots which often lead to longer, more difficult putts. Furthermore, different types of holes place emphasis on different parts of one's game. Par threes do not have a fairway for tee shots, and par fives give a serious advantage to long hitters.

Methodology:

For this reason, we decided to analyze the correlation between par four scoring average, strokes gained driving, and strokes gained putting collected from *ShotLink*™ data from the PGA Tour stat webpage. We chose to constrain our data to par fours, because there is only one tee shot before the approach, leaving less room for error, and there are more putts from an intermediate distance. However the foundation for our analysis lies in our decision to use "strokes gained" as the medium to measure relative performance of driving and putting. This statistic was developed by Mark Broadie, a business professor at Columbia University. It is calculated by summing the

average number of shots gained or lost depending on the location reached by the drive or putt being analyzed (PGATour.com). For example, a player may lose 0.3 strokes on his drive if it lands in a fairway bunker. We felt this gave the most accurate representation of the performance of putting and driving for each player.

Current Data, Tables, and Graphs:

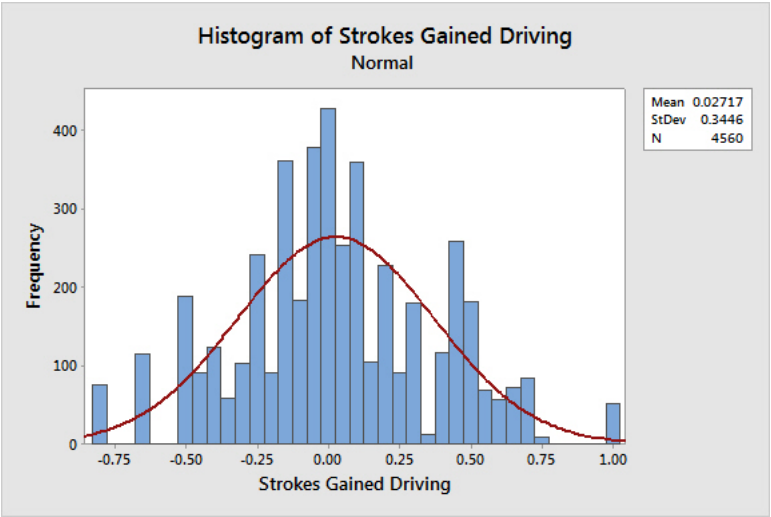


Figure 1: Histogram of strokes gained driving with normal distribution

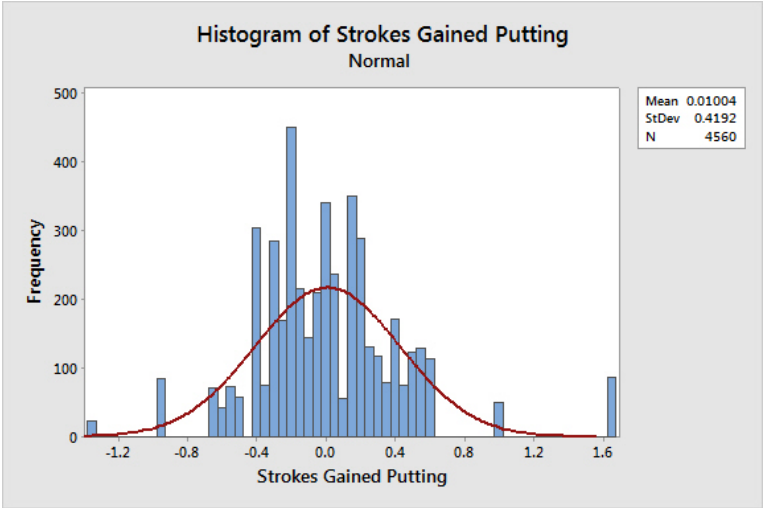


Figure 2: Histogram of strokes gained putting with normal distribution

The histograms above represent the distribution of strokes gained driving (SGD) and strokes gained putting (SGP) from the sample of players. Histograms were chosen here because they appropriately display this data even though it has several instances where multiple players have the same par four scoring average. As shown in the plot, both data sets take on the general shape of a bell curve, and we approximate them with normal distributions. For strokes gained driving the mean is 0.02717, and the standard deviation is 0.3446. For strokes gained putting the mean is 0.01004 and the standard deviation is 0.4192.

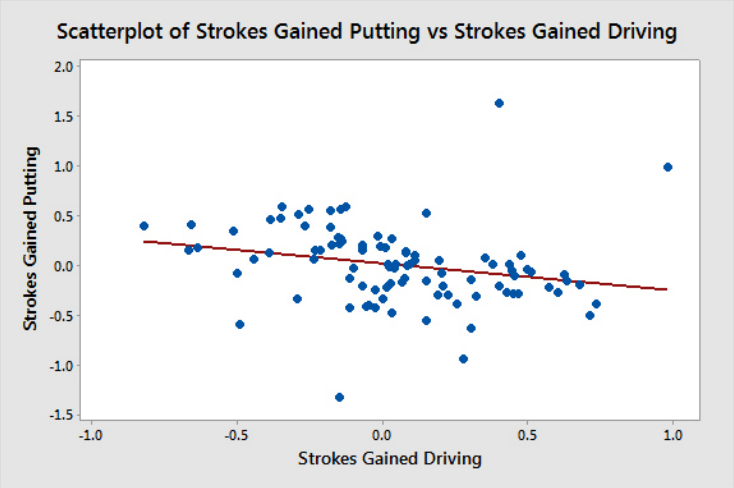


Figure 3: Scatterplot of SGP vs SGD with a fitted straight line

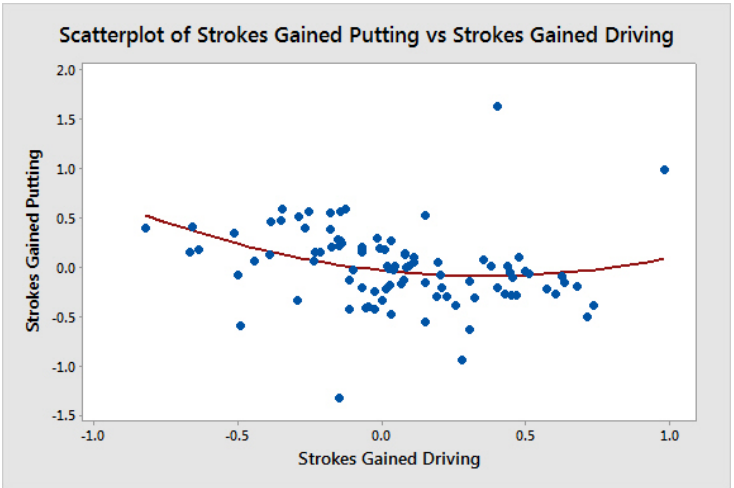


Figure 4: Scatterplot of SGP vs SGD with a quadratic fit

The two figures above are identical scatterplots of SGP vs SGD, and figure 3 has a linear fit while figure 4 has a quadratic fit. The reason both are included is the subtle possibility of nonlinear data. The straight line seems reasonable enough, but the curved shape of the quadratic appears slightly closer. This could suggest that as players stray from the mean of each category, the difference in their performance on SGP and SGD become larger. However, because we are only analyzing the middle 50% of players with a statistic that averages to 0 for all the players, it follows that a quadratic trend should not apply outside our window. The coefficient of determination for the linear fit was only 0.0567, and we could not do the same fit for the quadratic because it is nonlinear. Furthermore, the notion of driving performance and putting performance being related seems unlikely, and the poor fit of both lines supports this.

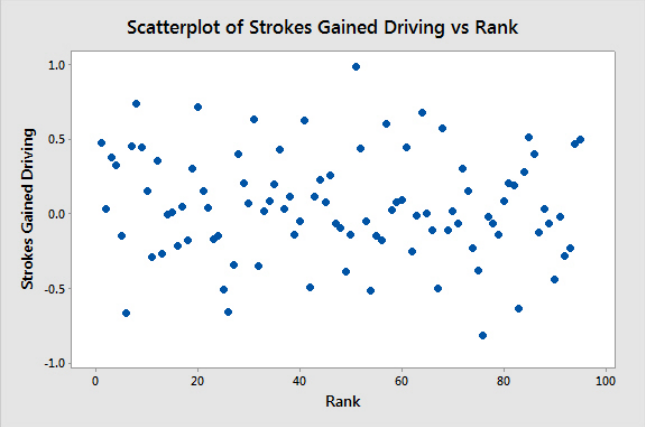


Figure 5: Scatterplot of SGD vs rank

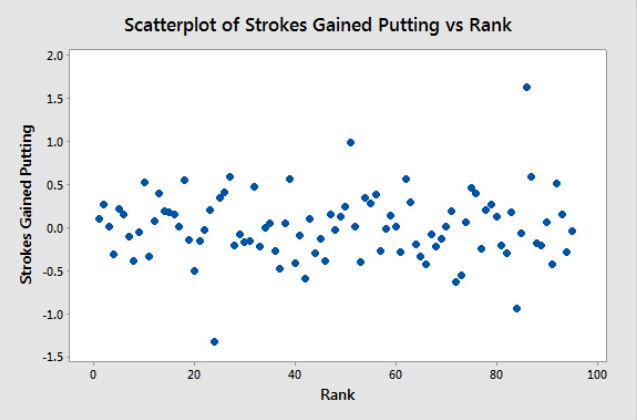


Figure 6: Scatterplot of SGP vs rank

For our last two figures we simply constructed scatterplots of SGD vs par four scoring rank and SGP vs par four scoring rank. These are the best indication of whether or not putting or driving has the greatest impact on par four scoring average because they directly compare the two strokes gained with the scoring rank. However, by all appearances from the graphs, there seems to be no relationships whatsoever. The SGP vs rank graph at first looks tighter, but that is only a result of the expanded y-axis to accommodate the outliers.

In order to confirm our suspicion that neither putting nor driving contribute more to the rank, we computed three correlation coefficients: Pearson, Spearman and Kendall. The following results were calculated using Minitab for the Pearson and Spearman, and VassarStats.com for the Kendall:

Table 1: Correlation coefficients and corresponding P-values

Pearson		Spearman		Kendall
Putting	P-value	Putting	P-value	Putting
0.009	0.933	-0.052	0.618	-0.1236
Driving	P-value	Driving	P-value	Driving
-0.092	0.618	-0.093	0.369	-0.0448

All three of these coefficients are very low, showing that no matter the test, the correlation between SGD and SGP and rank are extremely weak. However, strictly speaking, that is not the aim of this discussion. We are trying to determine whether driving or putting is more important. This is why we use a student t-test to compare the driving and putting distributions since we have a small data set.

Table 2: Student t-test, P-value and degrees of freedom

T-test	P-value	Degrees of Freedom
0.69	0.489	185

The T-test is not very high, so we know there at least a small relationship. However, because the P-value is greater than 0.05, that relationship is very weak.

Conclusion:

There is no clear winner here between driving and putting for impact on par four scoring. The golf community almost unanimously support that good putting is clearly more important than good driving, but that is not the case. Our selected population of PGA Tour strokes gained driving and strokes gained putting took the shape of a normal distribution when displayed in a histogram, confirming our theory that the data would tend towards the mean. We added linear and quadratic fits to scatterplots of SGD vs SGP, which indicate a lack of a direct relationship between driving and putting. Our last figures were scatterplots of both SGD vs rank and SGP vs rank. These showed the notably poor correlation for both statistics, and calculations for Pearson, Spearman and Kendall correlations coefficients further support this.

Possible explanations include our data range, choice of population and limited scope. Our data range was only the middle 50% of available players, and though the intent was to not

consider exceptions to any trends, the opposite effect may have occurred. The population was exclusively professional golfers, and the fact that most areas of their game are extremely good could keep the variations from being more pronounced. Finally, perhaps the most likely, is the difficulty to compare driving and putting using simple methods. There are literally hundreds of different aspects to a golfer's game, and these are just two. So the extremely weak correlation seen here is not necessarily a great measure of the importance of either stat. Simply put, too much is going on that affect scoring average to single out a single stat. Therefore, a better project would be with a much larger data set of a varying range of golfers and several other, less directly related, statistics as additional information.

Bibliography:

"Strokes Gained-Putting Statistic - Strokes Gained Explained." *PGATour*, 28 Oct. 2016, http://www.pgatour.com/stats/academicdata/shotlink.html.

"Statistics." *PGATour*, 26 Oct. 2016, http://www.pgatour.com/stats.html

Broadie, Mark. "Closer Encounters." Golf Magazine 57.8 (2015): 31. SPORTDiscus with Full Text. Web. 28 Oct. 2016.

Chimka, Justin R., and Thomas P. Talafuse. "Poisson Regression Analysis Of Additional Strokes Assessed At Golf." International Journal Of Sports Science & Coaching 11.4 (2016): 619-622. SPORTDiscus with Full Text. Web. 28 Oct. 2016.

Lowry, Richard. "Spearman Rank Order Correlation Coefficient." *VassarStats*. 30 Nov.2016, http://vassarstats.net/corr_rank.html

MATH 3320

Group 10: Christina Nielsen, Brianna Riley, Daniel Sunderland

Project's Report Feedback

Required Sections	Maximum Grade	Your grade	My comments
Introduction	15	15	Well written
Statement of the Problem	25	25	Well done
Methodology	25	20	Well explained. You covered all the key points in your statistical analysis except for the residual analysis. Your project has potential to yield more meaningful results, particularly for parameters estimation.
Results	25	25	Well explained as well.
Bibliography	10	10	Well done
Other comments			
Total	100	**95**	

Project's Presentation Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Be properly attired	5	5	Well done. Adequate for the subject chosen
Comments of another presentation	35	30	Well done
Chemistry among group members	20	10	The coordination need an improvement.
Timely submission of the presentation	20	20	You submitted your milestones on time
Pass all 4 Milestones	20	20	Well done
Other comments			Your video was a bit blurry at time, and the sound could be improved
Total	100	**85**	

Optimal Prediction Factors for a Victory in the NFL

Group #10:
Chrissy Nielsen
Brianna Riley
Daniel Sunderland

Chrissy Nielsen, Brianna Riley, Daniel Sunderland
MATH 3320 Project Report

Optimal Prediction Factors for a Victory in the NFL

I. Introduction:

Every year, many people around the country compete to create a model that best predicts who will come out on top at the end of the NFL season. Instead of creating a model trying to predict this year's NFL rankings, we decided to find which single parameter contributed the most to a team's chance of winning. The parameters we chose to test included the total yards ran and passed by a team, the rank of players for each team, and the notion of home field advantage for each team. Using data and rankings from a previous season, we constructed scatter plots, used Pearson correlation coefficients, and ran a Tukey test as a basis for comparison for the different parameters. We also used binomial distributions for the win/loss record to determine the probability of winning or losing games and then comparing our predictions to the games throughout the season. We collected data from all 32 teams regarding the chosen parameters to best achieve an accurate model to predict wins and losses for each team during the regular season.

II. Problem Statement:

We want to be able to predict which team will win any game in an NFL season. Do a statistical analysis of yardage, player ranking, and home field advantage, and their ability to predict the outcome of future games. Then compare each parameter to determine which is the optimal prediction factor.

III. Methodology and Results:

Each person on our team split up to tackle one factor individually in order to distribute the work. Our methods and results are below, sorted by factor.

A. Home Field Advantage

Chrissy's job in the project was to determine the effect of home field advantage on a team's chance of winning a game. She first recorded data from each game between all 32 teams of the NFL in the 2014 season. For the Packers, the data included who the Packers faced in every game, whether the Packers were playing at home or away, and whether the Packers won or lost. This data was then analyzed to count a "Home-Win" and an "Away-Loss" as a 1, which means the home team of that game won. The "Home-Loss" and "Away-Win" games were counted as a 0, meaning the home team of that game lost. These two categories, referred to as "Home-Win" and "Home-Loss" respectively, were then averaged over all of the games, giving a percentage of home wins and home losses. This was repeated for the other 31 teams of the NFL, and each of the home win and home loss percentages were averaged together. This gave us the choice of looking at individual home field advantages for teams or overall home field advantage for football games in general.

Table 1: Home and away wins and losses for the Patriots during their 2014-2015 NFL season.

Games	Home/Away	Win/Loss	Home Win	Home Loss
@Dolphins	0	0	1	0
@Vikings	0	1	0	1
Raiders	1	1	1	0
@Chiefs	0	0	1	0
Bengals	1	1	1	0
@Bills	0	1	0	1
Jets	1	1	1	0
Bears	1	1	1	0
Broncos	1	1	1	0
@Colts	0	1	0	1
Lions	1	1	1	0
@Packers	0	0	1	0
@Chargers	0	1	0	1
Dolphins	1	1	1	0
@Jets	0	1	0	1
Averages:	**0.47**	**0.80**	**0.67**	**0.33**

Table 1 was then repeated for all 32 total teams in the NFL which gives a large sample size because there are more than 30 points to analyze. The column with names shows who and where the Patriots played in each game. The averages at the bottom show that the Patriots, for example, played 47% of their games at home and won 80% of their games in total. Throughout the season, 67% of their games resulted with the home team winning, while 33% of the games resulted in the away team winning. This corroborates our assumption that teams playing at home have a slight advantage over the visiting team.

When each of these games were recorded, each particular point is counted twice because, as seen above in the highlighted row, when the Patriots play the Green Bay Packers in Green Bay, the away loss recorded for the Patriots counts as a "Home Win" because the Packers won at home. When the data was recorded for the Packers in the same manner, the data point where the Packers play at home against the Patriots also counts as a home win. The doubling of data points would usually skew the data favoring those points which were counted multiple times; however, because every single point was counted twice, the data is not biased and still reports the correct trend.

Most teams had a higher percentage of home wins than away wins, with the exception of a few possible outliers. The overall average percentage of home wins was 57.29%, while the overall away win average was 42.29%. These do not quite add up to 100% exactly because of the singular, double counted, tie game between the Carolina Panthers and the Cincinnati Bengals which we counted as a loss for both teams. Because the tie shows up twice, the home loss for each still balances with the away loss for each. The complete list of winning percentages for home and away games per team can be seen in the table below.

Table 2: Home and away loss and win percentages for each team during the 2014-2015 NFL season.

Team	Home Win = Away Loss	Home Loss = Away Win
49ers	46.67	53.33
Bears	40.00	60.00
Bengals	46.67	46.67
Bills	60.00	40.00
Broncos	73.33	26.67
Browns	53.33	46.67
Buccaneers	40.00	60.00
Cardinals	66.67	33.33
Chargers	53.33	46.67
Chiefs	66.67	33.33
Colts	60.00	40.00
Cowboys	26.67	73.33
Dolphins	53.33	46.67
Eagles	66.67	33.33
Falcons	53.33	46.67
Giants	53.33	46.67
Jaguars	66.67	33.33
Jets	53.33	46.67
Lions	66.67	33.33
Packers	73.33	26.67
Panthers	53.33	40.00
Patriots	66.67	33.33
Raiders	66.67	33.33
Rams	46.67	53.33
Ravens	60.00	40.00
Redskins	66.67	33.33
Saints	46.67	53.33
Seahawks	60.00	40.00
Steelers	66.67	33.33
Texans	53.33	46.67
Titans	53.33	46.67
Vikings	73.33	26.67

As seen above, the majority of the teams have a higher Home Win percentage than Away Win percentage, thus corroborating the notion of a “home field advantage”. Not every team followed this same trend. In fact, the Dallas Cowboys won almost all of their games, yet the only three games they lost were at home. This appears to be a possible outlier based on the rest of the data points, but cannot be ignored as an important percentage to investigate. We tested the

difference of the two data sets using a Tukey test which helps determine whether the two sets of data--Home Wins and Away Wins--are statistically different. So long as the difference of means is larger than the expected error of the data sets, the two data sets are said to be distinct, and have a significant difference. In laymen's terms, the Tukey test determines if the data is different enough to consider the two independent variables (Home Win or Away Win) different inputs.

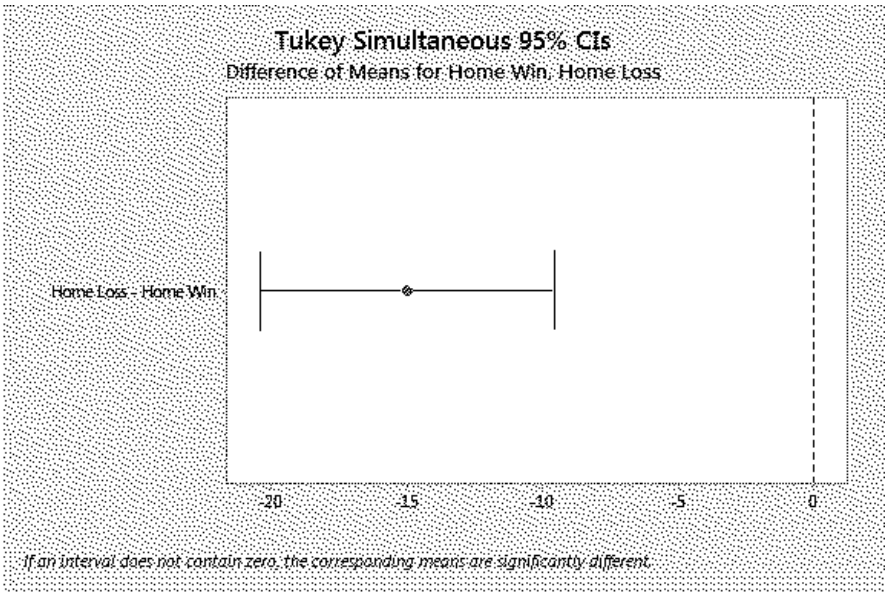


Figure 1: Tukey test showing that Home Win percentages and Away Win percentages are statistically different.

As seen in Figure 1, the Tukey test with 95% confidence intervals yielded negative values which never included 0. By the note given at the bottom of Figure 1, because the interval is all negative, we can see the average Home Win percentage is significantly different from the average Away Win percentage. The difference in data can also be seen with more clarity when viewed as a boxplot, as shown below in Figure 2. The singular outlier in the sets is from the highlighted Cowboys data in Table 2.

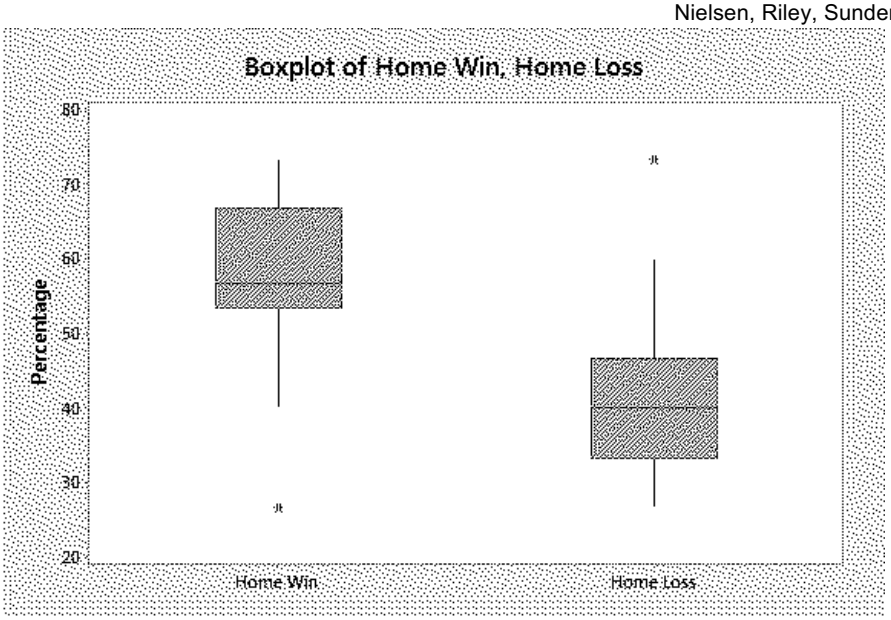


Figure 2: Boxplot showing the difference in data between Home Wins and Home Losses.

Because games end in a binary response, "win" or "loss", each game can be represented by a Bernoulli distribution. Multiple games can be represented by multiple Bernoulli distributions, which can also be referred to as a binomial distribution. A binomial distribution has two parameters: a probability of a trial resulting in a "success", and the number of trials. In this case, there are 15 games per team per season, therefore 15 trials. If we define a "success" as winning a home game, the corresponding probability would be the average percentage of Home Wins in decimal form: 0.5729. When organized together, the probability of winning any number of home games in a season of 15 home games is given by equations (1) and (2).

$$x \sim \text{Bin}(p=0.5729, n=15) \quad (1)$$

$$\sum_{x=0}^{n} \binom{n}{x} p^x (1-p)^{n-x} \quad (2)$$

By adjusting the values of x, this equation can also be used to determine the probability of a team winning ranges of games, such as "less than 4 home games" or "between 8-10 home games". The probabilities, respectively, are given by equations (3) and (4).

$$\Pr(x<4) = P(0) + P(1) + P(2) + P(3) \quad (3)$$

$$\Pr(8 \leq x \leq 10) = P(8) + P(9) + P(10) \quad (4)$$

These probabilities can be important for estimating how many wins a team will get based solely on how many home and away games they have; however, an individual team has an average probability of winning of 0.5729.

Yardage

To find the correlation, if any, between yardage and winning, we originally chose to collect data from the New England Patriots', Carolina Panthers', and Denver Broncos' 2014-2015 NFL season. We predict the relationship between percent yards and percent points to be about 1:1 and for there to be a low correlation. We predict that total yardage will not be a reliable way to predict the winner of a game. We chose to only collect data from one season so that there would be fewer changing variables, such as changes in players or coaches. We decided to find the percentage of total yards ran and the percentage of the total points scored by each team in each game. Then, a scatter plot was used to visually compare the data, as shown in Figure 3.

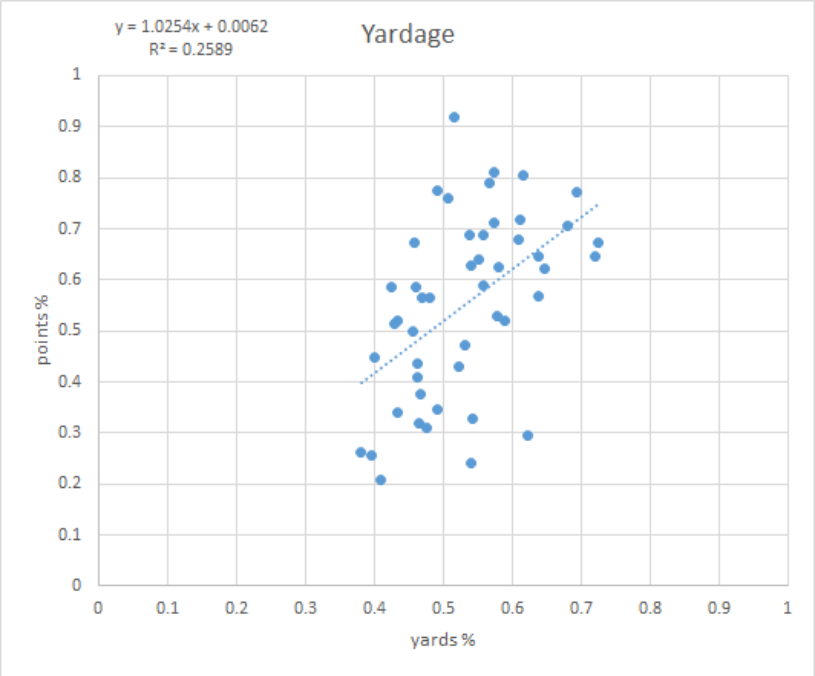


Figure 3: Scatter plot comparing yardage to points in individual football games of three football teams in 2014-2015 NFL season.

As expected, the linear trendline has a slope of about 1. The data had a Pearson correlation coefficient of 0.5088. The correlation coefficient suggests a moderate relationship. It is clear that, since running yards is required to gain points, there should be a positive relationship between the yards ran and the points scored by a team. There is also a high likelihood that other variables affect both a team's yardage and score.

We decided that the data collected from the three teams was not an accurate representation of the NFL. Each team chosen for our sample won their division, which could possibly bias the results. Additionally, three teams is a small sample for a population of 32 teams. We decided to include two more teams into our data. We collected additional data for the Tennessee Titans, who lost their division, and the San Francisco 49ers, who had an equal number of wins and loses in 2014. A scatterplot of the new data for these five teams is shown in Figure 4.

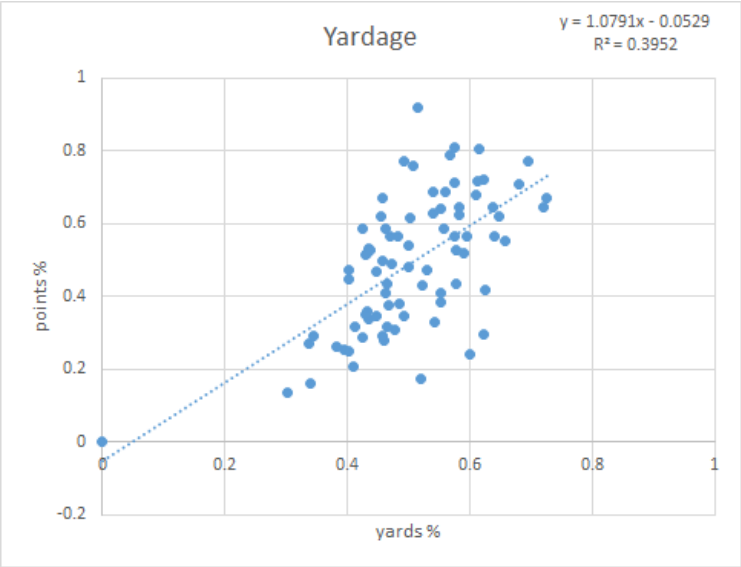


Figure 4: Scatter plot comparing yardage to points in individual football games of five football teams in the 2014-2015 NFL season.

By adding two more teams, the Pearson correlation coefficient increased to 0.6286, without dramatically changing the slope of the trendline. The more teams and seasons we include in our sample, the better representation we will have of the data. We decided that 5 teams in a single season is a sufficiently large sample for our purpose.

In this study, we want to find out if past total yard percentages are a reliable way to predict the outcome of a game. We hypothesize that the team with the highest previous average number of yards will win the game. Before testing our hypothesis, we wanted to determine if teams who run more yards are generally ranked higher. To show this, we made a boxplot, shown in Figure 5, comparing the distribution of yards run by each team in each game.

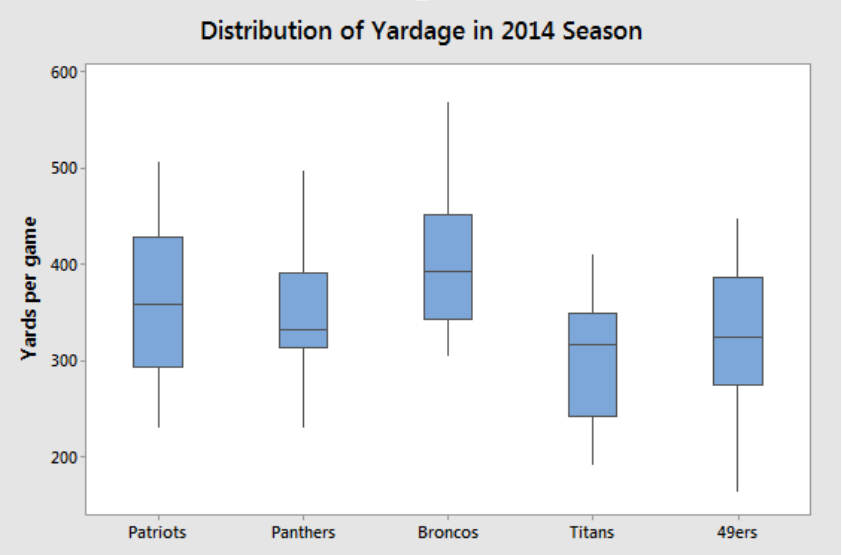


Figure 5: Distribution of each team's yardage throughout the 2014-2015 NFL season.

The patriots, who won the Superbowl, have the second highest average of the five teams. Otherwise, the results are as expected. The titans, who did the worst, have the lowest average, follows by the 49ers, who were the next worst team of the five. There are no outliers in the data, but each data set has a large deviation from the mean. This suggests that better teams run more yards, but a team's yardage still varies greatly from game to game, likely due to changes in conditions and strategies of the opposing team.

To test the hypothesis, we found the average number of yards ran by a team and their opponent in all past games of the season. If past yardage can be used to predict the winning team of a game, the team with the largest sum of yards should win the game. In order to obtain the largest data set, we will observe the last game of each team's season, summing up the yardage of their first 15 games. When this method was applied, the total yardage only successfully predicted the outcome of the game three out of the five games studied.

Finally, we decided to use the binomial model to find the probability that the team with the most yards will win in every game of the season. We have data for five teams, each playing 16 games in 2014, giving us a total of 80 trials. Since each game is independent and has no bearing on other games, we can use this method. We defined success as the team with the most yards in a game having the most points. Of the 80 trials, 69 were successful, making the probability of success 0.7375. Using the binomial equations (4) and (5), we found that there is only a 0.00049% chance that less than 40 of 80 trials will be successful. There is a 35.817% chance that at least 60 of 80 new trials will be successful. It is almost certain that the team that runs the most yards will win at least half the time, but we cannot expect all the trials to be successful.

$$x \sim \text{Bin}(p=0.7375, n=80) \quad (5)$$

We have shown that there is a positive relationship between the number of yards a team runs and the percentage of the total points that team won. However, yards ran in previous games is a poor indicator of who should win a game.

Table 3: Data used in the analysis of each of the 5 teams in our sample.

Team	Patriots		Panthers		Broncos		Titans		49ers	
Game	% yard	% points	% yard	% points	% yard	% points	% yard	% points	% yard	% points
1	0.47	0.38	0.56	0.59	0.47	0.56	0.44	0.41	0.53	0.44
2	0.57	0.81	0.49	0.77	0.46	0.59	0.51	0.23	0.54	0.41
3	0.55	0.64	0.43	0.34	0.46	0.43	0.57	0.66	0.54	0.57
4	0.40	0.25	0.41	0.21	0.73	0.67	0.59	0.79	0.27	0.33
5	0.61	0.72	0.48	0.56	0.64	0.65	0.52	0.44	0.36	0.35
6	0.54	0.63	0.46	0.50	0.57	0.71	0.54	0.50	0.43	0.29
7	0.43	0.52	0.48	0.31	0.58	0.63	0.52	0.69	0.42	0.38
8	0.56	0.69	0.46	0.41	0.54	0.33	0.54	0.59	0.46	0.67
9	0.46	0.67	0.38	0.26	0.68	0.71	0.62	0.74	0.32	0.29
10	0.61	0.68	0.46	0.32	0.54	0.24	0.54	0.68	0.46	0.76
11	0.57	0.79	0.53	0.47	0.59	0.52	0.47	0.53	0.41	0.48
12	0.40	0.45	0.62	0.30	0.72	0.64	0.38	0.70	0.28	0.36
13	0.65	0.62	0.62	0.80	0.42	0.59	0.38	0.20	0.58	0.41
14	0.51	0.76	0.58	0.53	0.54	0.69	0.42	0.47	0.46	0.31
15	0.43	0.52	0.64	0.57	0.52	0.43	0.36	0.43	0.48	0.57
16	0.49	0.35	0.52	0.92	0.69	0.77	0.48	0.08	0.31	0.23

Player Rankings

The player rankings parameter was used to identify if there was a correlation between the amount of top ranking players and each team's chance of winning per game. Each team was given a certain number of points for every top 100 player on its roster, based on the ranking assigned at the end of the season. The number 1 player gave 100 points to his team whereas the number 100 player only gave one point to his team. Once all the top 100 players had given the amount of points they were worth to their respective teams seen in Table 4, we created a scatterplot of the player points compared to the team ranking seen in Figure 6.

Table 4: Player points and percentage of player points per team.

Team	Player Points	Percent of Player Points
49rs	0	0.00
Bears	52	1.06
Bengals	63	1.29
Bills	241	4.92
Broncos	260	5.31
Browns	185	3.78
Bucaneers	141	2.88
Cardinals	114	2.33
Chargers	119	2.43
Cheifs	230	4.69
Colts	201	4.10
Cowboys	222	4.53
Dolphins	199	4.06
Eagles	217	4.43
Falcons	110	2.24
Giants	68	1.39
Jaguars	79	1.61
Jets	197	4.02
Lions	173	3.53
Packers	298	6.08
Panthers	124	2.53
Patriots	196	4.00
Raiders	87	1.78
Rams	64	1.31
Ravens	181	3.69
Redskins	125	2.55
Saints	70	1.43
Seahawks	506	10.33
Steelers	217	4.43
Texas	119	2.43
Titans	4	0.08
Vikings	38	0.78

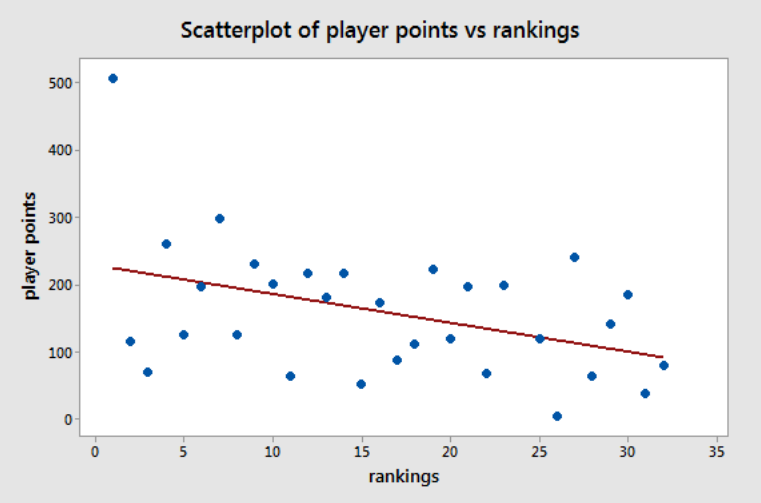


Figure 6: Player points versus team rankings for the 2014-2015 NFL season with a best fit line.

While the best fit line does not provide a very good fit to the data, a trend does appear in that the teams with fewer player points do tend to have a lower team rank than the teams with more player points. This trend has a moderate relationship with a Pearson correlation coefficient of 0.42. There is one potential outlier belonging to the Seattle Seahawks who not only had the most players in the top 100, but also all their top 100 players had high rankings giving them an overwhelming amount of player points. It is no coincidence that the Seahawks were also ranked number one at the end of the season. The Seahawks were also one of the contenders in the 2015 Super Bowl though they did lose to the New England Patriots who were then ranked number 6. The Patriots did not have one of the most player points but actually were closer to the average number of player points per team. Other discrepancies in the data show that the team with the fourth highest amount of top ranked players was in the bottom 25% of the team rankings; meanwhile, two of the top three teams had less than 100 player points, which was below average, and still managed to be the second and third ranked teams in the country.

To find how far this discrepancy goes, we derived the average percentage of player point differences for each team. This was done by using equation (6). That way, we could find the percentage of player points belonging to each team for each game.

$$\frac{\%\ Player\ Points\ of\ Basis\ Team}{\%\ Player\ Points\ of\ Basis\ Team + \%Player\ Points\ of\ Other\ Team} \tag{6}$$

Once this was done for each team, we got the average of those percentages and compared them to the actual percentage of games won by that team. For example, the Seahawks have 10.33% of all the player points. When playing against the Bears who have 1.06% of all the player points, the Seahawks have a 90.68% of all the player points giving them a 90.68% chance of winning that game based only the amount of top ranked players per team. We then continued to get the chance

of winning a game against every other team and averaged all those probabilities to show that the Seahawks have a 79.24% chance of winning a game. After gathering the chance of winning a game for each team, we compared these chances to the actual percentage of games won by each team shown in Table 5.

Table 5: Comparison between predicted games won and actual games won for each team during the 2014-2015 NFL season.

Team	Actual Win %	Predicted Win %	Absolute Value Difference	% Difference
49ers	46.67	0	46.67	infinite
Bears	33.33	32.23	1.1	3.412969283
Bengals	73.33	35.93	37.4	104.0912886
Bills	53.33	65.16	11.83	18.15531001
Broncos	73.33	66.75	6.58	9.857677903
Browns	46.67	59.41	12.74	21.44420131
Buccaneers	13.33	53.33	40	75.00468779
Cardinals	73.33	48.56	24.77	51.00906096
Chargers	60.00	49.52	10.48	21.1631664
Chiefs	53.33	64.16	10.83	16.87967581
Colts	66.67	61.24	5.43	8.866753756
Cowboys	73.33	63.4	9.93	15.66246057
Dolphins	53.33	61.02	7.69	12.60242543
Eagles	60.00	62.91	2.91	4.625655699
Falcons	40.00	47.76	7.76	16.2479062
Giants	40.00	37.46	2.54	6.780565937
Jaguars	20.00	40.57	20.57	50.70248952
Jets	20.00	60.79	40.79	67.09985195
Lions	73.33	57.92	15.41	26.60566298
Packers	73.33	69.55	3.78	5.434938893
Panthers	46.67	50.44	3.77	7.474226804
Patriots	80.00	60.68	19.32	31.83915623
Raiders	20.00	42.62	22.62	53.07367433
Rams	40.00	36.24	3.76	10.37527594
Ravens	60.00	58.92	1.08	1.83299389
Redskins	26.67	50.62	23.95	47.3133149
Saints	40.00	38.05	1.95	5.124835742
Seahawks	73.33	79.24	5.91	7.458354366
Steelers	53.33	62.91	9.58	15.22810364
Texans	53.33	49.52	3.81	7.693861066
Titans	13.33	6.38	6.95	108.9341693
Vikings	46.67	26.73	19.94	74.59783015
Average	**50.00**	**50.000625**	**13.8078125**	**29.24492082**

Continuing with the Seahawks example, the Seahawks won 73.33% of their games making the absolute difference betwen the chance of winning a game and the actual games won 5.91 and the percent difference in predicted win percentage of games and actual games won 7.46%. In the end, the average absolute difference was 13.81 and the average percent difference was 29.24% therefore we are able to predict the amount of games won by each team with a success rate of 70.76%. There are a few outliers in this data that are important to note. One of which is the fact that for the 2014-2015 NFL season, the San Francisco 49ers had no players ranked in the top 100 players for that season. So according to the model, the 49ers should not have won any games, but in reality they won 44% of their games. This would turn out an infinite percent difference which would throw off the success rate. For that reason, the 49ers were not included in this model except to be compared to other teams and giving the other teams a 100% chance of beating the 49ers. Other outliers include the Bengals and the Titans who had more than a 100% difference. These huge percent differences were countered by most of the other teams who had percent differences below 10%.

Discussion:

After reviewing each parameter, we compared their ability to predict a winner. The percentage of player points had a success rate of 70.76%, the number of yards a team ran by each team had a success rate of 60.00%, and the home field advantage had a 57.29% success rate.

While home field advantage was found to have the least impact on the game, it is the only parameter that can be determined before the season begins. With no prior knowledge of the season, home field advantage can pick the winner 57.29% of the time. To use yardage as a predictor, teams must play games first. The more games each team plays, the more data we have to predict the winner. Player rankings are given after the season, so we cannot use that season's ranking to predict a winner of a game. We could make a prediction based on past season rankings, but this will not be as accurate and could leave holes in the data due to new players being drafted and older players being let go.

Depending on which data and how much data each person has, any of the three proposed factors could have different benefits when comparing two teams head to head. In addition, it may be more helpful to look at an individual team's factor percentages over multiple years in case one team always plays better at home, or tends to score more points with less yards. These parameters could be improved by increasing the sample size with more teams and more seasons.

Conclusion:

In this project, we confirmed that in each data sample, there was a correlation between each parameter chosen and chance of winning. Additionally, each parameter was successful in predicting the winner of a game the majority of the time. Through the project we determined the

most reliable factor in predicting a team's chance of winning a game is the percentage of player points, followed by the number of yards a team ran and passed and the least reliable factor being whether the team played at home or away.

References:

[1] "2014 Carolina Panthers Statistics & Players | Pro-Football-Reference.com." *Pro-Football-Reference.com*. N.p., n.d. Web. 30 Nov. 2016.
[2] "2014 Denver Broncos Statistics & Players | Pro-Football-Reference.com." *Pro-Football-Reference.com*. N.p., n.d. Web. 30 Nov. 2016.
[3] "2014 New England Patriots Statistics & Players | Pro-Football-Reference.com." *Pro-Football-Reference.com*. N.p., n.d. Web. 30 Nov. 2016.
[4] "2014 NFL Schedules | National Football League". *FBSchedules.com*. N.p., 2016. Web. 30 Nov. 2016.
[5] "2014 San Francisco 49ers Statistics & Players | Pro-Football-Reference.com." *Pro-Football-Reference.com*. N.p., n.d. Web. 30 Nov. 2016.
[6] "2014 Tennessee Titans Statistics & Players | Pro-Football-Reference.com." *Pro-Football-Reference.com*. N.p., n.d. Web. 30 Nov. 2016.
[7]"4 Game Sweep :: NFL 32 Ranking All 32 NFL Teams For The 2014-2015 Season". *4gamesweep.sportsblog.com*. N.p., 2016. Web. 2 Dec. 2016.
[8] "Pro Football Statistics And History | Pro-Football-Reference.Com". *Pro-Football-Reference.com*. N.p., 2016. Web. 2 Dec. 2016.
[9]"Top 100 Players Of 2015". *NFL.com*. N.p., 2016. Web. 2 Dec. 2016.

MATH 3320

Group 11: Catherine Goodin, Regina Mangold, Laura Preston

Project's Report Feedback

Required Sections	Maximum Grade	Your grade	My comments
Introduction	15	15	Well written
Statement of the Problem	25	20	Well done in general, although there was an entire paragraph that belonged in the conclusion.
Methodology	25	25	Well explained. You cover all the key points in your statistical analysis.
Results	25	25	Well explained as well. Your findings are somewhat surprising and contradicts the common belief.
Bibliography	10	10	Well done
Other comments			
Total	100	**95**	You could mention attempts you made to deal with large values of n, like using the Sterling Formula etc.

Project's Presentation Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Be properly attired	5	5	Very Well done.
Comments of another presentation	35	30	Well done
Chemistry among group members	20	15	Good coordination, but Catherine did not speak as much as the other two
Timely submission of the presentation	20	20	You submitted your milestones on time
Pass all 4 Milestones	20	20	Well done
Other comments			I like the professional look and tac of all group members.
Total	100	**90**	

Weather Related Car Crashes in Texas

By
Catherine Goodin
Regina Mangold
Laura Preston

MATH 3320
1 December 2016
Dr. Kwessi

1.) Introduction

A posted speed limit on a given Texas road only indicates the safest maximum speed drivers should travel 'in good conditions' (clear weather, in daylight, and on dry roads). The Texas Department of Public Safety [7] recommends that drivers slow their speeds and drive more cautiously in unsafe driving conditions. In spite of these recommendations thousands of drivers are involved in car accidents that result in fatalities or injuries every year on Texas roads. We wanted to determine if poor weather and poor road conditions were less safe than ideal road conditions (dry roads and clear/sunny weather). This research paper examines several different weather conditions (clear, fog, rain, sleet/snow), and their resulting road conditions (dry, wet, and icy) to determine the relationship, if any, between poor weather/road conditions and driving-related injuries and fatalities on Texas roads between 2006 and 2015.

2.) Statement of the Problem

There are several things we wanted to learn from our research. First, we wanted to know in which type of weather and on which road conditions a person is most likely to get into a crash, get into a fatal crash, and to be injured in a crash simply based upon the raw data. Secondly, we wanted to know the probability of there being x_1 fatalities/injuries in dry road conditions, x_2 fatalities/injuries in wet road conditions, and x_3 fatalities/injuries in icy road conditions in a given year. Thirdly, we wanted to know which weather condition is the worst to drive in when taking into account the number of days a year of that particular weather condition.

We predicted that clear weather and dry road conditions would produce the fewest weather related car accidents for both fatal accidents and accidents in which injuries are sustained. Our initial analysis of this data did not support this prediction which led us to pursue question three. For the new model which takes into account the number of days of each type of weather per year, we predicted that the number of accidents per day of rain, sleet/snow, and fog will be greater than the number of accidents per day of clear weather. This follows more closely to our initial intuition; a more indepth analysis of these phenomena is included in the Results section.

3.) Methodology

We wanted to connect our project back to probability and statistics by observing the probabilities of getting into certain car accidents using several different techniques. Our first approach was used in an attempt to answer our first question, "In which type of weather and on

which road condition is a person most likely to get into a car crash, a fatal crash, and be severely injured in a crash?"

To collect data, we went to the Texas Department of Transportation website [6] where we found comprehensive data for all of the car accidents in the state of Texas. The data were separated by weather type as well as road condition for the years 2003-2015. We decided to focus only on 2006-2015 and omit the data from 2003, 2004, and 2005 simply because we wanted to limit the scope of our project, and we felt that ten years was a healthy sample. We also thought that it was important not to go back too far, so that we did not risk our results being skewed due to changing technology, long-term changes in climate, laws, or culture.

The site had data for the weather categories: blowing dust, blowing sand/snow, clear/cloudy, fog, rain, severe crosswinds, sleet/hail, smoke, snow, other, and unknown. We chose to overlook the data from other and unknown as we felt it would not contribute to the overall conclusions of our project. We combined the data for sleet and snow into one sleet/snow category since they were effectively the same for our purposes. We also did not use the data from the blowing dust, blowing sand/snow, severe crosswinds, and smoke as these categories had very low contributions to the total crashes and they seemed to be less significant to the focus of our project. Regarding road conditions, the data on this site was divided into the following categories: dry, ice, sand/mud/dirt, snow, standing water, wet, other, and unknown. Again, we chose to leave out the other and unknown categories for the same reasons, and we did not use the sand/mud/dirt, slush, or standing water categories as we felt that they were less significant than the others. This left us with five weather condition categories: clear/cloudy, fog, rain, sleet, snow, and three road condition categories: dry, icy, and wet. Our data for total accidents, fatal accidents, and accidents resulting in injuries for road conditions and weather conditions are all shown in the tables located in Appendix A.

To analyze our data in an attempt to answer our first question, we first used several bar charts to visually display the results from our data acquisition. We used bar charts because the data is categorical. Then we found the probability of getting into an accident, getting into a fatal accident, and getting into an accident resulting in injury for each road condition and each weather type. To find the probability of being injured in a crash for dry road conditions, we divided the average number of car crashes resulting in injuries for dry road conditions by the averaged total for each type of crash for all of the road conditions. We repeated this process for each weather category and each type of road condition for each type of accident and calculated the probabilities shown in Appendix C.

Regarding the second question, we analyzed our data using a multinomial model. Originally we considered a binomial model because we had a select number of trials, we knew what a success was, and we could find the probability of each success. However, the binomial model did not account for the number of categories contained in our data, which led us to implement the multinomial model. A multinomial model is used to create a logistic regression of data with more than two possible discrete outcomes. In the model for car accidents during

specific road conditions, we defined x_1 as the fatalities/injuries in dry road conditions, x_2 as the fatalities/injuries in wet road conditions, and x_3 as the fatalities/injuries in icy road conditions.

$$P_r(x_1, x_2, x_3) = \frac{N!}{x_1!x_2!x_3!}(P_1^{x_1}P_2^{x_2}P_3^{x_3})$$

Where: x_i = the number of fatalities due to the given condition

$$n_1 = \sum_{i=1}^{n} Dry_i \quad n_2 = \sum_{i=1}^{n} Icy_i \quad n_3 = \sum_{i=1}^{n} Wet_i$$

$$N = n_1 + n_2 + n_3$$

$$P_i = \frac{n_i}{N}$$

Figure 1*: Multinomial model used to represent the data for fatalities due to road conditions. This model can also be used to represent injuries and total car accidents due to road conditions.*

The trouble was that *N* very large and as a result *N!* was impossible to calculate without a supercomputer. Similarly, to model the probabilities of getting into a car accident under certain weather conditions we used a multinomial model. In this model, we defined x_1 as the fatalities/injuries in sunny/clear weather conditions, x_2 as the fatalities/injuries in rainy conditions, x_3 as the fatalities/injuries in foggy weather conditions, and x_4 as the fatalities/injuries in sleet/snowy weather conditions. Again, the value of *N!* was too large to be calculated without the use of a supercomputer.

$$P_r(x_1, x_2, x_3, x_4) = \frac{N!}{x_1!x_2!x_3!x_4!}(P_1^{x_1}P_2^{x_2}P_3^{x_3}P_4^{x_4})$$

Where: x_i = the number of fatalities due to the given condition

$$n_1 = \sum_{i=1}^{n} Clear_i \quad n_2 = \sum_{i=1}^{n} Rain_i \quad n_3 = \sum_{i=1}^{n} Sleet/Snow_i \quad n_4 = \sum_{i=1}^{n} Fog_i$$

$$N = n_1 + n_2 + n_3 + n_4$$

$$P_i = \frac{n_i}{N}$$

Figure 2*: Multinomial model used to represent the data for fatalities due to weather conditions. This model can also be used to represent injuries and total car accidents due to weather conditions.*

We were suspicious that our first model did not provide an accurate indication of which weather condition was the worst to drive in, so we decided to do further research in an attempt to answer question 3. We found sources [3], [8], [2], and [1] which listed average days of sunshine,

fog, snow/sleet, and rainy weather, respectively, for roughly 60 major cities in Texas. For the purposes of this research paper, we had to assume that the Texas climate is uniform across the state. We took the sum of all of the days of sunshine, rain, fog, and snow/sleet, and divided them by the total number of cities we had data for to calculate the average days of each type of weather for the state of Texas. We felt our calculations (as shown in Table 1 below) were feasible approximations as the total days of each type of weather added up to 365.25 days.

***Table 1**: Average days of weather in Texas assuming weather is uniform across the entire state*

Weather Type	**Rain**	**Snow/Sleet**	**Fog**	**Sunshine/ Clear**	**Total**
Average Days of weather	99	1.15	135.5	129.6	365.25

We then calculated the probabilities of getting into a car accident, a fatal car accident, and an accident resulting in injury for our four types of weather: clear, fog, rain, and sleet/snow. We used the same method from question one to find the probabilities for question three. We took the number car accidents with each outcome during a given weather condition and divided these values by the averaged total of each type of accident for each type of weather condition as shown in the pie charts in Figures B.13-B.18.

4.) Results

One thing that is troubling about our data is that it is statewide meaning that there is no distinction between rural and urban roads. For this reason, we cannot see how the different weather and road conditions affect the amount of crashes on different types of roads and in different areas. Statewide data also causes concern about the difference in local climates. Texas is a very large state with varying terrain and respective climates. For example, the panhandle has a very different climate than the gulf coast. We were able to find data for local weather climates in Texas based on different major cities in several regions of Texas, but we had to assume that the weather was uniform across the entire state of Texas even though we know that is inaccurate.

In spite of the limitations of our data, our statistical analysis provided intriguing results. The bar charts shown in Figures B.1, B.3, B.5, B.7, B.9, and B.11 were created to analyze the raw data and show that clear days account for the vast majority of the total car accidents, fatal car accidents, and accidents involving serious injury. It wasn't until these bar charts were cropped that accidents due fog, rain, sleet/snow and wet or icy could be seen on the weather condition and road condition graphs. This is shown even further in the pie charts located in Figures B.2, B.4, B.6, B.8, B.10, and B.12 where the clear weather condition probability

overwhelms the probabilities for fog, rain, and sleet/snow, and dry road condition probabilities overwhelm the probabilities for wet or icy for total accidents, fatal accidents and accidents resulting in injury.

Unfortunately, we were not able to compute actual values for our multinomial model due to our limited computational power. Though the model allowed us to explore the theoretical representation of these probabilities.

After seeing such a large proportion of of accidents attributed to clear weather conditions, we removed the clear weather data condition from our analysis. This allowed us to focus only on accidents that occurred during poor weather conditions. The probability of being involved in an accident during rainy, foggy, or sleet/snowy weather conditions is shown in Figures B.19-B.21, along with the pie charts for fatal accident and accidents that result in an injury. These pie charts indicate that rainy conditions are the most dangerous type of weather to drive in.

Furthermore, once we took into account the frequency of each type of weather we produced the pie charts shown in Figures B.25-B.27. In these charts we saw that clear weather still accounts for the majority of accidents, but it no longer completely overshadows the other types of weather. Taking this a step further, we again decided to remove the clear weather data from our analysis for a better understanding of the effect of weather on car accidents. These pie charts are shown in Figures B.28-B.30. These charts show that snow and sleet account for the majority of car accidents per day of snow or sleet, followed by rain and least of all fog.

Another interesting result of our data analysis was that in 2009 there was a huge decrease in injuries due to car accidents for clear weather. We were intrigued as to why this drop was so large, and after more research we found that the Texas state Congress passed a plethora of bills regarding road and traffic safety in this year. Among these bills were House Bill 537 which declared that everyone in a car must wear a seatbelt and House Bill 55 which prohibited the use of a hand-held communication device while driving. This was done in an attempt to curtail such high injury and fatality rates from automobile accidents. Our data suggests that these bills may have been successful, but further testing would be required to be sure.

5.) Bibliography

[1] "Average Annual Precipitation for Texas." Average Yearly Precipitation for Texas - Current Results. N.p., n.d. Web. 21 Oct. 2016.

[2] "Average Annual Snowfall in Texas." Average Annual Snowfall Totals in Texas - Current Results. N.p., n.d. Web. 21 Oct. 2016.

[3] "Days of Sunshine Per Year in Texas." Annual Days of Sunshine in Texas - Current Results. N.p., n.d. Web. 21 Oct. 2016.

[4] S. HB 537, 81st Cong. (2009) (enacted). Web. 23 Oct. 2016

[5] S. HB 55, 81st Cong. (2009) (enacted). Web. 23 Oct. 2016

[6] Texas Department of Transportation (State of Texas). "Texas Motor Vehicle Crash Statistics - 2015." Texas Motor Vehicle Crash Statistics - 2015. N.p., n.d. Web. 21 Oct. 2016.

[7] Texas Driver Handbook. Austin, TX: Texas Department of Public Safety, Driver License Division, 2016. Print.

[8] "Total Cloudy and Foggy Days at US Cities a Year." Annual Days of Cloud and Fog at US Cities - Current Results. N.p., n.d. Web. 21 Oct. 2016.

Appendix A

Tabulated Raw Data

Table A.1

Fatalities in given Weather Conditions											
Types of Weather	2015	2014	2013	2012	2011	2010	2009	2008	2007	2006	2005
Clear	2780	2911	1477	2810	2618	2541	2563	2876	2758	2836	2928
Fog	44	54	22	52	20	35	39	36	56	41	29
Rain	279	191	121	142	113	180	167	170	239	181	170
Sleet	12	10	13	1	12	3	6	6	6	13	4
Snow	8	2	5	5	8	8	5	7	9	2	1

Table A.2

Fatalities in Given Road Conditions											
Road condition	2015	2014	2013	2012	2011	2010	2009	2008	2007	2006	2005
Dry	2644	2837	2688	2753	2549	2444	2489	1413	2660	1563	2818
Icy	29	25	28	5	22	7	24	13	19	13	11
Wet	407	280	293	223	176	270	273	171	360	173	312

Table A.3

Injuries in Given Weather Conditions											
Types of Weather	2015	2014	2013	2012	2011	2010	2009	2008	2007	2006	2005
Clear	60362	60623	16919	58712	54428	54370	136656	144579	149403	154283	168954
Fog	353	410	172	391	273	272	873	885	1013	1009	837
Rain	6691	4892	1573	4234	3038	4898	14974	11865	18590	13791	12004
Sleet	273	401	153	57	162	58	312	379	610	492	204
Snow	261	204	60	118	203	282	396	234	401	96	116

Table A.4

Injuries due to Road Conditions											
Road condition	2015	2014	2013	2012	2011	2010	2009	2008	2007	2006	2005
Dry	57241	57963	57151	56620	52519	51799	131098	30225	142095	33048	161338
Icy	672	833	595	164	589	325	1046	244	1425	419	689
Wet	9062	7100	7040	6016	4388	6881	20507	4317	25674	4832	20893

Table A.5

Total Weather Condition Accidents											
Total Crashes	**2015**	**2014**	**2013**	**2012**	**2011**	**2010**	**2009**	**2008**	**2007**	**2006**	**2005**
Clear	450509	426099	396973	378169	352895	346365	370920	392077	389786	385009	423259
Fog	2564	2459	1994	2275	1618	1512	2677	2544	2882	2654	2270
Rain	57958	40444	40933	33083	23641	38246	46509	36126	53230	38161	32066
Sleet	2184	3106	2089	359	1422	430	1198	1114	2091	1374	715
Snow	2572	2046	1034	1046	2535	2390	1672	909	1675	450	338

Table A.6

Total Road Condition Crashes											
Total Crashes	**2015**	**2014**	**2013**	**2012**	**2011**	**2010**	**2009**	**2008**	**2007**	**2006**	**2005**
Dry	425197	406240	378195	364295	339611	329549	354791	84204	369232	86981	403094
Icy	5595	6514	4582	1192	5914	2442	4080	847	5189	1296	2445
Wet	77883	57001	55973	45492	33057	51877	62481	13991	72351	14129	55288

8

Appendix B

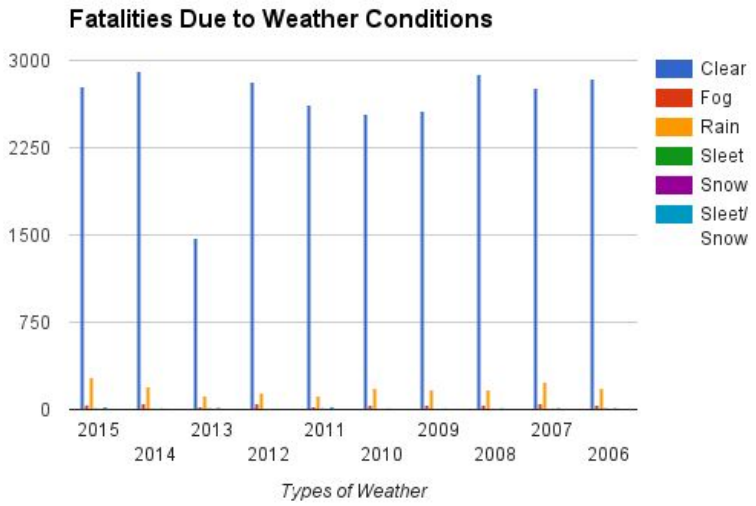


Figure B.1: *Bar chart showing the fatal car crashes due to weather conditions from our raw data.*

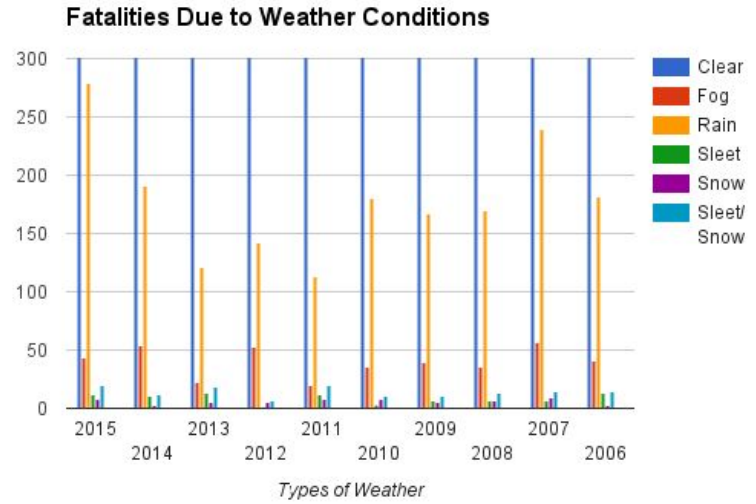


Figure B.2: *Bar chart showing the fatal car crashes due to weather conditions from our raw data. Bar charts are zoomed in to minimize skew from clear data category.*

9

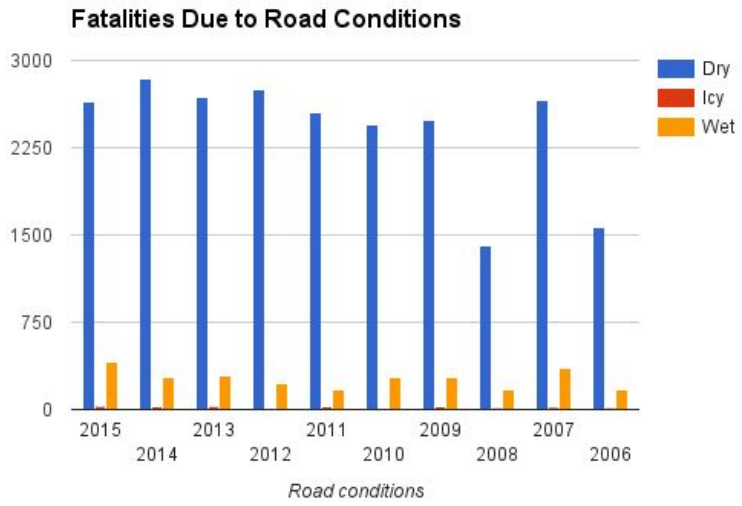


Figure B.3: *Bar chart showing the fatal car crashes due to road conditions from our raw data*

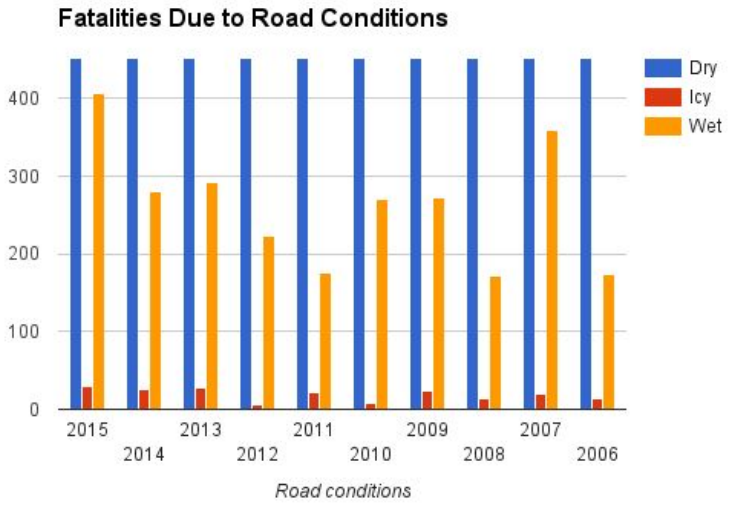


Figure B.4*: Bar chart showing the fatal car crashes due to road conditions from our raw data. Bar charts are zoomed in to minimize skew from dry data category.*

10

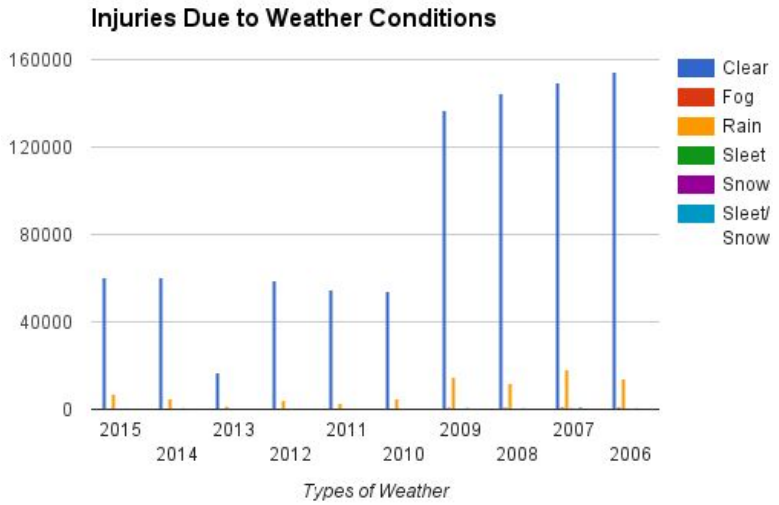


Figure B.5: *Bar chart showing the car accidents resulting in injuries due to weather conditions from our raw data.*

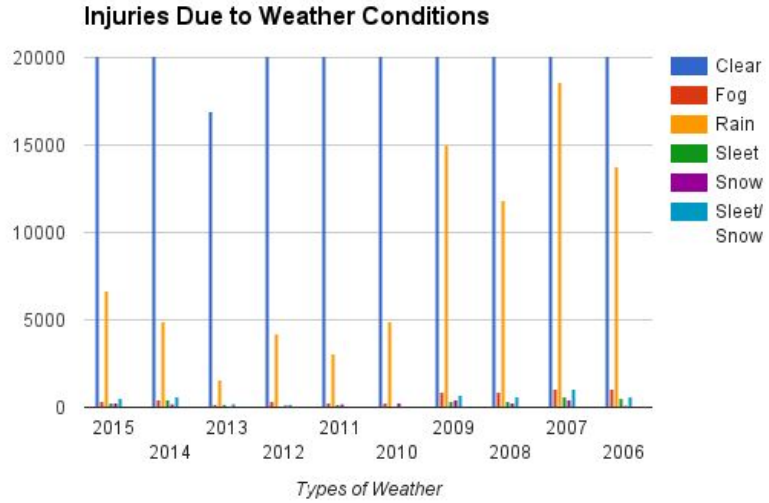


Figure B.6: *Bar chart showing the car accidents resulting in injuries due to weather conditions from our raw data. Bar charts are zoomed in to minimize skew from clear data category.*

11

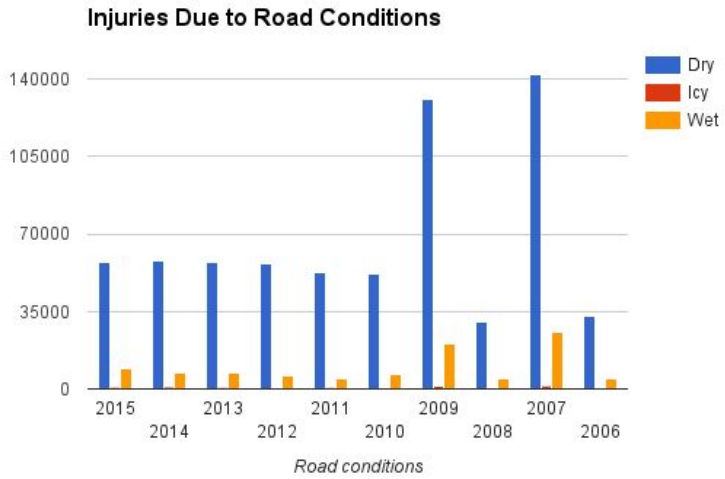


Figure B.7: *Bar chart showing the car accidents resulting in injuries due to road conditions from our raw data.*

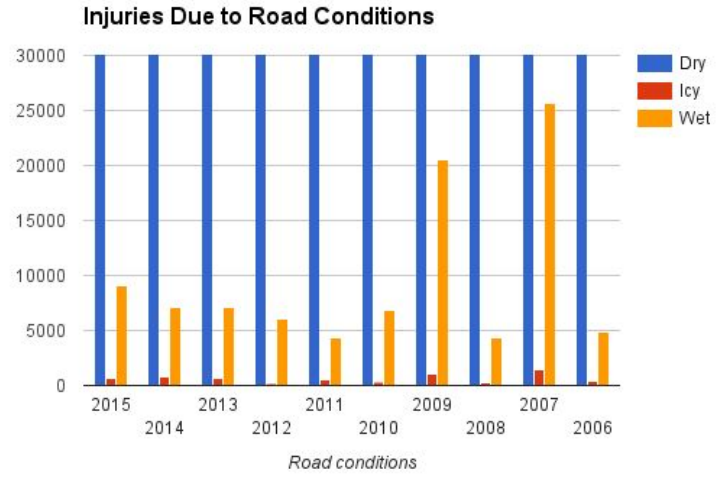


Figure B.8: *Bar chart showing the car accidents resulting in injuries due to road conditions from our raw data. Bar charts are zoomed in to minimize skew from dry data category.*

12

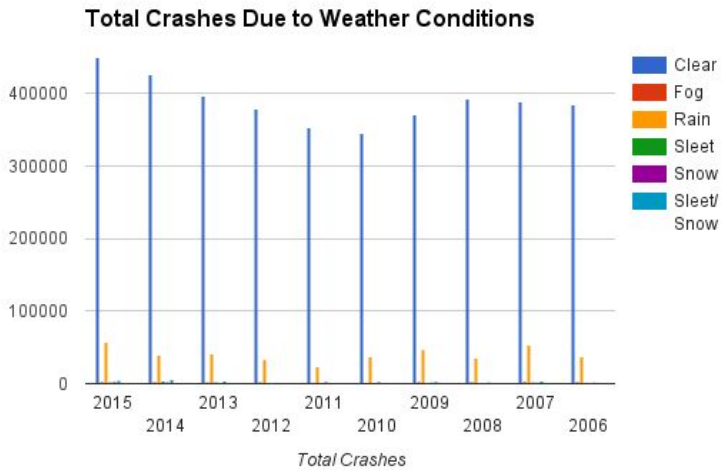


Figure B.9: *Bar chart showing the total car accidents due to weather conditions from our raw data.*

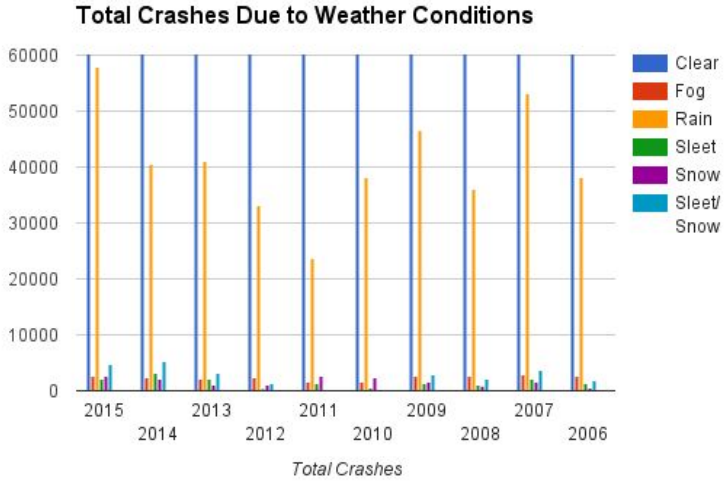


Figure B.10: *Bar chart showing the total number of car accidents due to weather conditions from our raw data. Bar charts are zoomed in to minimize skew from clear data category.*

13

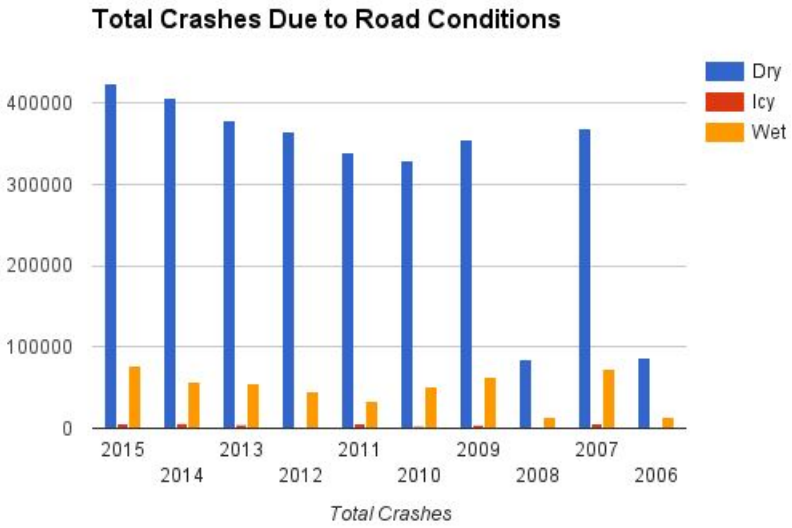


Figure B.11: *Bar chart showing the total number of car accidents due to road conditions from our raw data.*

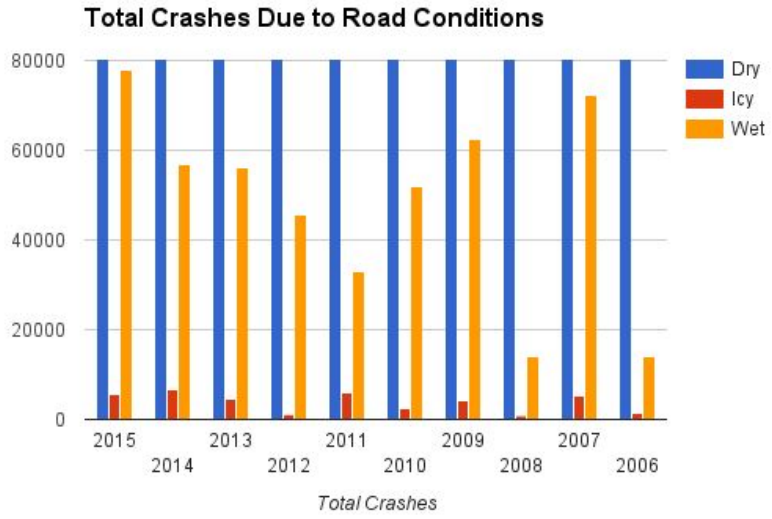


Figure B.12: *Bar chart showing the total number of car accidents due to road conditions from our raw data. Bar charts are zoomed in to minimize skew from dry data category.*

14

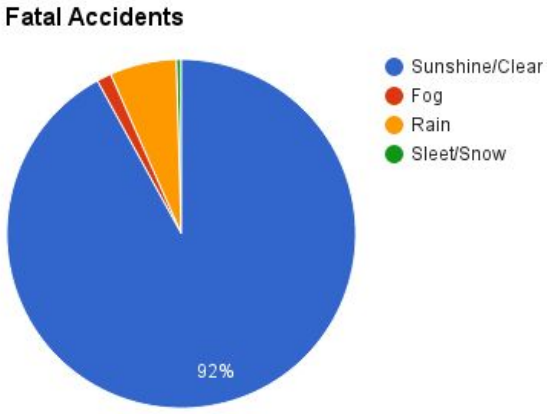


Figure B.13: *Pie chart showing the probabilities of car accidents resulting in fatalities due to weather conditions based upon our raw data.*

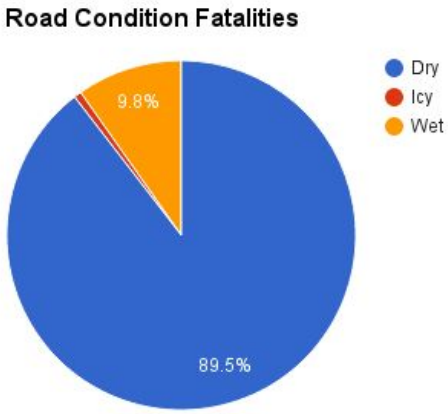


Figure B.14: *Pie chart showing the probabilities of car accidents resulting in fatalities due to road conditions based upon our raw data.*

15

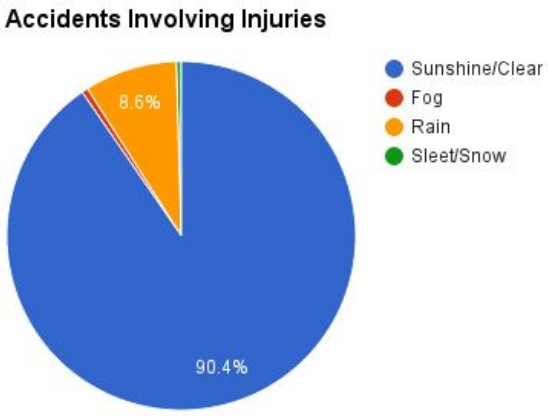


Figure B.15: *Pie chart showing the probabilities of car accidents resulting in severe injuries due to weather conditions based upon our raw data.*

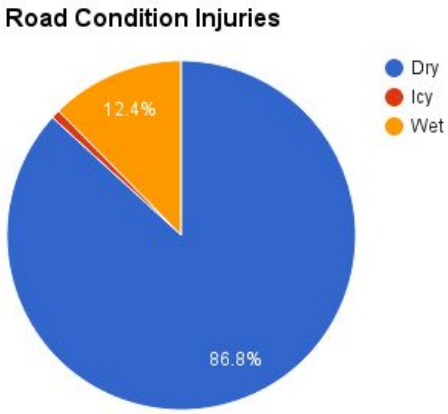


Figure B.16: *Pie chart showing the probabilities of car accidents resulting in severe injuries due to road conditions based upon our raw data.*

16

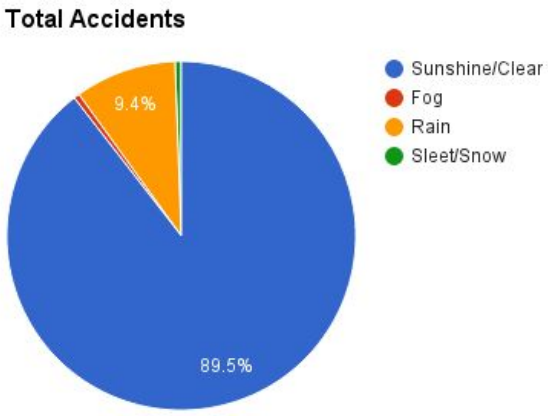


Figure B.17: *Pie chart showing the probabilities of total car accidents due to weather conditions based upon our raw data.*

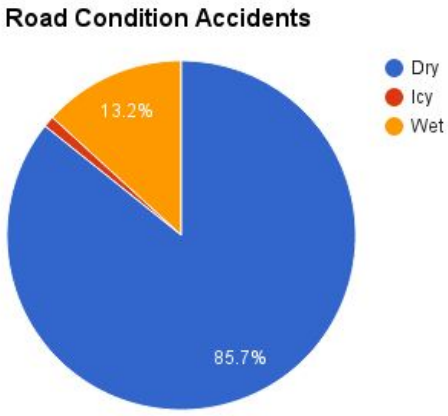


Figure B.18: *Pie chart showing the probabilities of total car accidents due to road conditions based upon our raw data.*

17

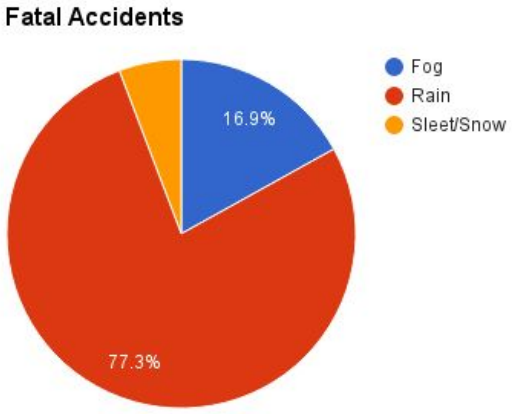


Figure B.19: *Pie chart showing the probabilities of fatal car accidents due to weather conditions based upon our raw data without the clear conditions represented.*

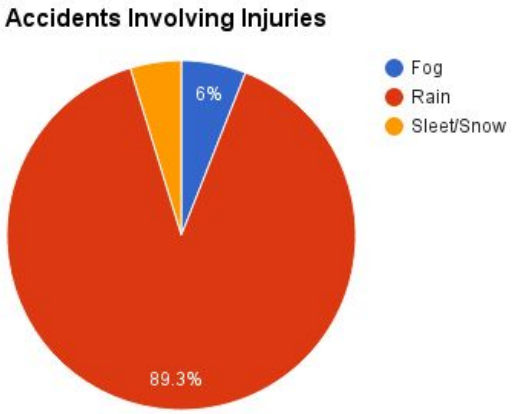


Figure B.20: *Pie chart showing the probabilities of car accidents resulting in severe injuries due to weather conditions based upon our raw data without the clear conditions represented.*

18

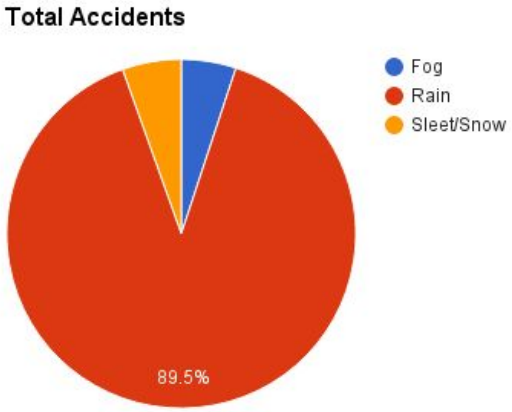


Figure B.21: *Pie chart showing the probabilities of the total number of car accidents due to weather conditions based upon our raw data without the clear conditions represented.*

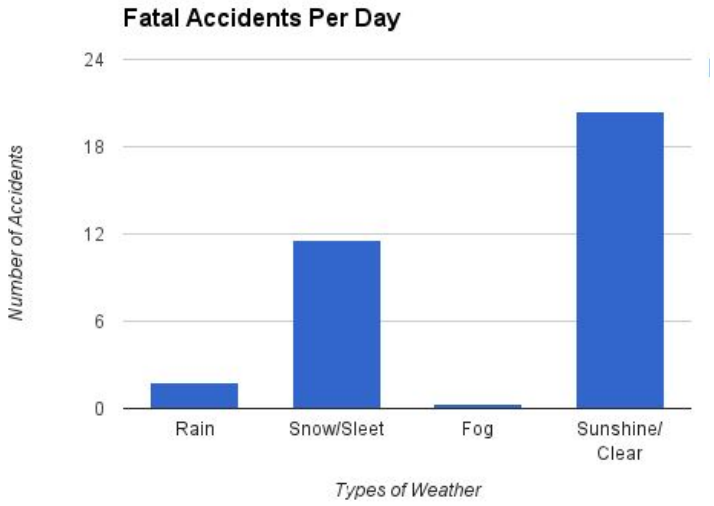


Figure B.22: *Bar chart showing the number of fatal car accidents due to weather conditions per day of that type of weather.*

19

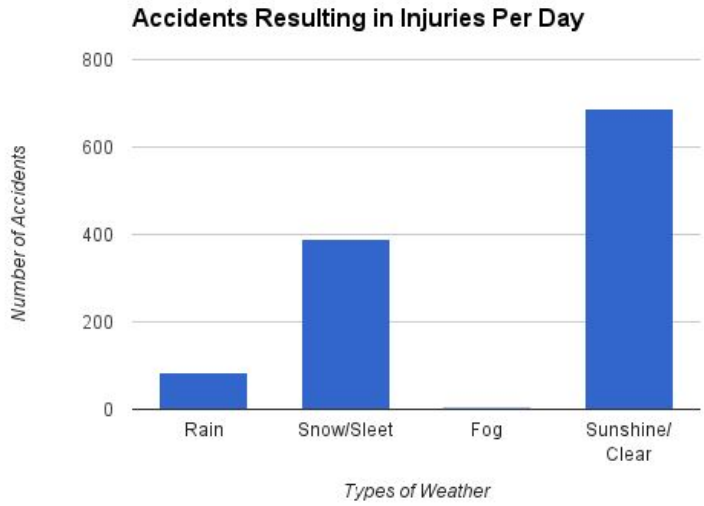


Figure B.23: *Bar chart showing the number of car accidents resulting in severe injuries due to weather conditions per day of that type of weather.*

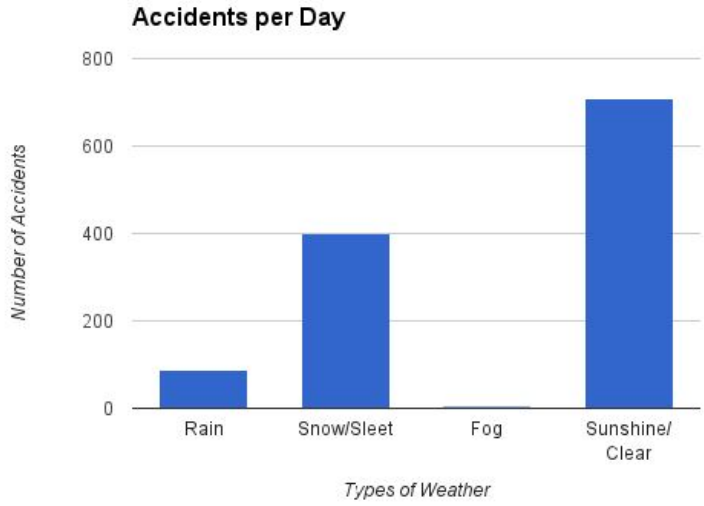


Figure B.24: *Bar chart showing the total number of car accidents due to weather conditions per day of that type of weather.*

20

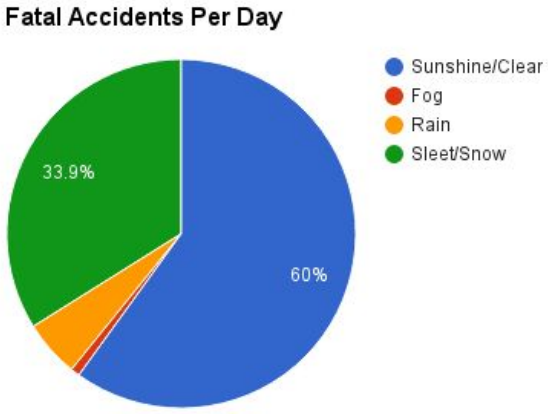


Figure B.25: *Pie chart showing the probabilities of fatal car accidents due to weather conditions per day of that type of weather.*

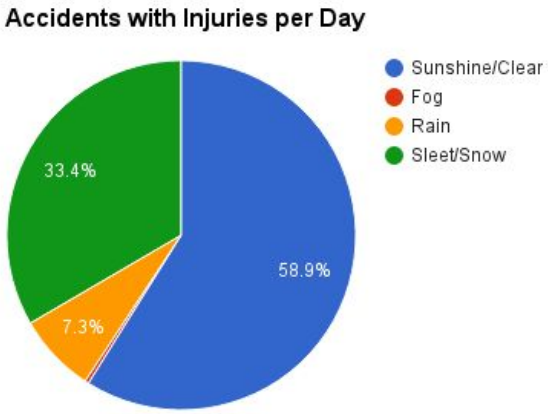


Figure B.26: *Pie chart showing the probabilities of car accidents resulting in severe injuries due to weather conditions per day of that type of weather.*

21

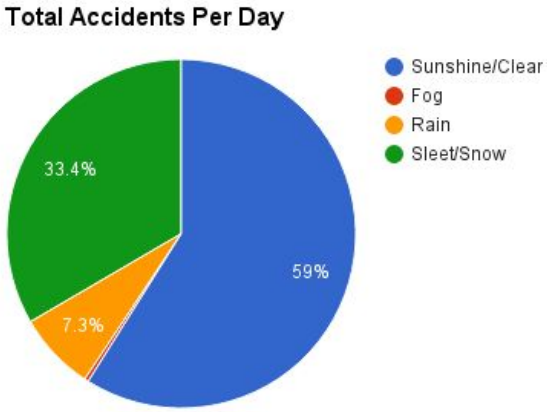


Figure B.27: *Pie chart showing the probabilities of the total number of car accidents due to weather conditions per day of that type of weather.*

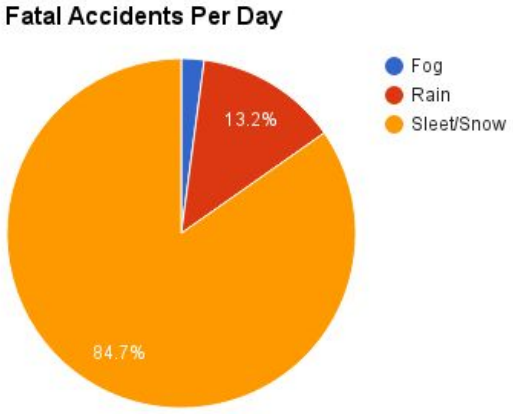


Figure B.28: *Pie chart showing the probabilities of fatal car accidents due to weather conditions per day of that type of weather without the clear conditions represented.*

22

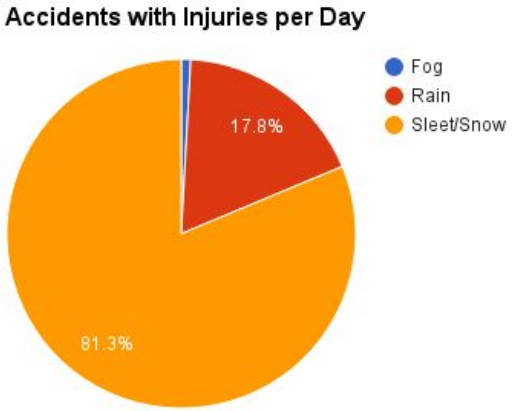


Figure B.29: *Pie chart showing the probabilities of car accidents resulting in severe injuries due to weather conditions per day of that type of weather without the clear conditions represented.*

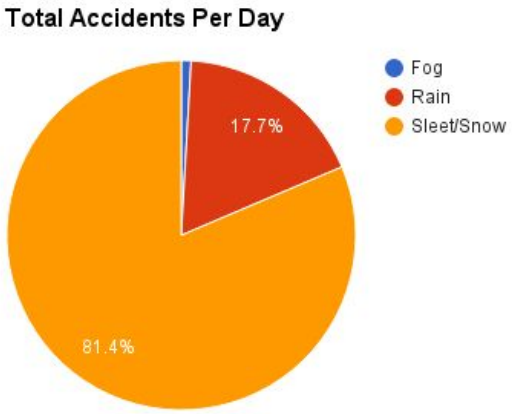


Figure B.30: *Pie chart showing the probabilities of the total number of car accidents due to weather conditions per day of that type of weather without the clear conditions represented.*

23

Appendix C

***Table C.1**: Fatal car accidents: n values and probabilities due to weather.*

n_1	n_2	n_3	n_4	p_1	p_2	p_3	p_4
29098	428	1953	146	0.9201	0.01353	0.06175	0.00462

***Table C.2**: Fatal car accidents: n values and probabilities due to road conditions.*

n_1	n_2	n_3	p_1	p_2	p_3
24040	185	2626	0.98175	0.00756	0.010069

***Table C.3**: Car accidents resulting in injury: n values and probabilities due to weather.*

n_1	n_2	n_3	n_4	p_1	p_2	p_3	p_4
890335	5651	84546	4478	0.90388	0.00577	0.08583	0.00455

***Table C.4**: Car accidents resulting in injury: n values and probabilities due to road conditions.*

n_1	n_2	n_3	p_1	p_2	p_3
669759	6312	95817	0.86769	0.00818	0.12413

***Table C.5**: Total number of car accidents: n values and probabilities due to weather.*

n_1	n_2	n_3	n_4	p_1	p_2	p_3	p_4
3888802	23179	408331	24950	0.89495	0.00533	0.09397	0.00574

***Table C.6**: Total number of car accidents: n values and probabilities due to road conditions.*

n_1	n_2	n_3	p_1	p_2	p_3
3138295	37651	484235	0.85742	0.01029	0.13229

MATH 3320

Group 12: Molly McCollough, Micheal Erickson, James Scranton

Project's Report Feedback

Required Sections	Maximum Grade	Your grade	My comments
Introduction	15	15	Well written
Statement of the Problem	25	20	Well done in general.
Methodology	25	25	Well explained. You cover all the key points in your statistical analysis.You failed to proposed a mathematical model that could be used for future prediction.
Results	25	25	Well explained as well. Your findings are somewhat surprising and contradicts most if not all models proposed prior to the election.
Bibliography	10	10	Well done
Other comments			
Total	100	**95**	

Project's Presentation Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Be properly attired	5	5	Well done.
Comments of another presentation	35	30	Well done
Chemistry among group members	20	15	Good coordination
Timely submission of the presentation	20	20	You submitted your milestones on time
Pass all 4 Milestones	20	20	Well done
Other comments			Your video was a bit blurry.
Total	100	**90**	

TRINITY UNVERSITY

Predicting the Next President Using Historical Data

MATH 3320-1

Instructor: Dr. Eddy Kwessi
PLEDGED: Group 12
Molly McCullough, Michael Erickson, and James Scranton
1 December 2016

Introduction

The majority of election forecasters predict the outcome of elections based on polling data. Forecasts are of particular interest when trying to determine the outcome of the presidential election. By relying on state and national polls, sites like FiveThirtyEight attempt to name the winning candidate by using information on current public opinion. FiveThirtyEight has a statistical model that weights polls according to sample size, how recently it was taken, and known biases to account for the inherent uncertainty in polling. This site also combines historical, economic, and demographic information to make a more accurate prediction. This is similar to the strategies employed by other forecasters. However, in the wake of the outcome of this election, it is clear that the method used by these forecasters was flawed. Major news outlets like NBC, ABC, CNN, and FOX all predicted that Hillary Clinton would be the next president while in actuality, Donald Trump was elected. Even FiveThirtyEight, a site that prided itself on its correct prediction of the election outcome in 2008 and 2012, gave the victory to Clinton. Election predictions can have important effects on voter turnout by discouraging or encouraging certain voters, voter engagement, and public opinion. Swayed public opinion can then influence future polling data. Predictions can also sway the focus of media coverage of candidates. In light of this, it is important to make sure that election predictions are accurate. Because major election predictions incorrectly named Clinton as the winner of the election, it is worthwhile to investigate if there is a more accurate way to predict the outcome without relying on current polling data.

Statement of Problem

Polling data can be unreliable because it can be hard to find accurate and truly random samples. When the sample is not random, the poll is not likely to be a good representation of the total population of the state or the nation. This affects the forecasts made based on the information provided by the polls. Our purpose is to determine if using historical data can make a more accurate prediction of the outcome of the 2016 election than polling data. We are choosing to focus on the historic voting record and patterns for all fifty states and the District of Columbia, whose votes are counted separate from Maryland, and unemployment rates immediately preceding the election. Use of historical data will allow us to avoid the lack of randomness associated with polling and will show trends that can help us predict future behavior of states.

Methodology

To make a prediction of the outcome of the 2016 election, we chose to focus on key states. Key states are those whose electoral votes typically go to the winner of the overall election and are important for candidates to win. Many states in the country vote for the same party in election after election. There are, however, states commonly referred to as “swing” states that could and have voted for candidates of either major party even in recent years. These states become the focus of most presidential campaigns and

are therefore key in predicting the winner of the election. We used past voting behavior of all fifty states and the District of Columbia since 1976 to identify key states. Using election data compiled by The American Presidency Project of UC Santa Barbara, we recorded for each election and each state whether it was won by the overall winning candidate or not and which party's candidate actually won each state. For the first set of data, which can be found in the first column of Table 1, we coded a win of the state by the overall winner as a 1 and a loss as a 0. For each state, the wins and losses were summed and divided by the number of overall elections. If the state had a value of 0.8 or above, meaning that in 80 percent of the elections it was won by the winning candidate, we identified it as a potential strong predictor. If the state had a value of between 0.7 and 0.8, we identified it as a potential moderate predictor. For the second set of data, indicated by the second column of Table 1, we coded for which party's candidate won each state regardless of whether they were the winner of the national election. A win by a Republican candidate was coded a 0 and a win by a Democrat was coded as a 1. In the analysis of this data, we chose to not include Independent candidates. This is because these candidates did not win a state during any of the election years we focused on and their share of votes received is minute compared to those received by the major party candidates. For this second data set, we similarly summed totals for each state and divided by the number of elections in which a major party candidate won the electoral votes of that state. A value of 0.5 meant that this state voted for a Republican candidate at the same rate as a Democratic candidate. We identified states with that had values within 10 points of 50 percent as potential strong indicators. The upper limit would be 60 percent, meaning they had a slight historical preference for the Democratic Party, and a lower limit of 40 percent, which would mean the state would have a slight historical preference for the Republican Party. For a state to be considered a key state, it had to have been identified as a potential strong indicator and either a potential strong or moderate predictor. If a state was not considered key, we predicted its voting behavior to be similar to past behavior. Using the data on which party won each state, we assigned states with values less than 0.40 as Republican and states with values greater than 0.60 as Democratic and gave their electoral votes to that respective candidate. However, because key states were those shown to have a voting record that did not indicate strong preference for one candidate over the other, we choose to look at unemployment data, national GDP in comparison to margin of victory for the incumbent party, and the rate of change of disposable income growth to find voting preference to see if there is a relationship between jumps in the socio-economic factors and a switch in the voting behavior of key states.

We quantitatively identified the unemployment rate by recording the unemployment rate for each key state immediately preceding the November election and comparing this data to how that state voted in the election, whether for the incumbent party or against it. We recorded how many states flipped when there was a jump in the unemployment rate of either 1, 2, or 3 points or if the rate increased by 40, 50, 75 or 100 percent. The number of vote flips over the total number of jumps gave us a probability that the state would

vote against the incumbent party when there was a substantial increase in the unemployment rate. Only once unemployment rates increased by a full 3 points would states begin to show a moderate to strong trend of voting for a non-incumbent candidate with a rate of 77.78 percent. Looking at the rate of change of unemployment by percentage increases gave a clearer picture. Starting at 50% increase in unemployment, states were already willing to switch party votes by 69.23 percent. At a 75% increase, states were willing to switch their votes by 77.78 percent. By 100% increase, states turn their votes away from the incumbent party 80.00 percent of the time. Unfortunately, for the sake of our predictions, none of our identified key states had an increase in unemployment this election cycle by more than 0.2 points and most actually decreased in unemployment, thus invalidating our use of the unemployment data.

Another potential factor we identified was the rate of growth of disposable income in the voting pattern of key states. We recorded data from the U.S. Bureau of Economic Analysis for the real disposable income for election years 1972 through 2012. These values were used to calculate the percent growth from one election year to another for 1976 through 2012. The purpose of this analysis was to find a party preference for each key state. For each election that did not include an incumbent candidate (1980, 1992, 2000, 2008) we compared the growth rate associated with that year to the growth rate associated with the previous election year with respect to the party that the key state voted for. We found probabilities that a decrease in rate would favor each party and that an increase would favor each party.

We used a weighted average of the influence of GDP/margin of victory data, unemployment data, disposable personal income data, and indicator value to predict how each key state would vote. Because we did not use unemployment data, as all of the unemployment rates decreased in each key state from 2012, we used the three other categories. We weighted the indicator value as 0.4, because of the fact that states are resistant to change and tend to follow historical trends. We weighted GDP as 0.35 because although it is an indicator of the margin of victory for the incumbent party, this relationship is weak. There are many variables that affect the margin of victory besides GDP and these are not necessarily consistent for every election. Finally, we weighted the influence of the growth rate of disposable income as 0.25 because since we only could apply it to a decrease in rate, as shown in the Results section, it tended to skew the results towards the Democratic party. Each key state was coded as a 0 for Republican or a 1 for Democrat for each of the three categories and a weighted average was found for each state. Not all states had data for all three categories. When only two of the three could be determined, the weighted average was compared to the total of the two weights used rather than 1. An average value over half the weight total was considered a Democratic vote and a value under half the weight total was considered a Republican vote. Finally, we counted the electoral votes won by each candidate based on our prediction of how each state, key or not, would vote. A candidate that received more than 270 votes was considered the victor.

Results

To identify key states we recorded data on the rate of both whether each state was won by the overall winning candidate and which party won each state in all elections starting in 1976 through 2012. Those states that were identified as key were Florida, Ohio, New Mexico, Nevada, New Hampshire, Vermont, New Jersey, California, Connecticut, Illinois, Michigan, and Maine. Table 1 shows the data used to determine the key states. Potential strong predictors are shown in yellow, potential moderate predictors in green, and potential strong indicators in orange. Key states, highlighted in red, are those that are a strong indicator and either a strong predictor or a moderate predictor.

Table 1. Identification of Key States

State	Predictor	Indicator
Alabama	0.60	0.10
Alaska	0.50	0.00
Arizona	0.60	0.10
Arkansas	0.80	0.30
California	0.70	0.60
Colorado	0.80	0.30
Connecticut	0.70	0.60
Delaware	0.80	0.70
Dist. Of Col.	0.50	1.00
Florida	0.90	0.40
Georgia	0.60	0.30
Hawaii	0.60	0.90
Idaho	0.50	0.00
Illinois	0.70	0.60
Indiana	0.60	0.10
Iowa	0.70	0.60
Kansas	0.50	0.00
Kentucky	0.80	0.30
Louisiana	0.80	0.30
Maine	0.70	0.60
Maryland	0.70	0.80
Massachusetts	0.70	0.80
Michigan	0.70	0.60
Minnesota	0.50	1.00
Mississippi	0.60	0.10
Missouri	0.80	0.30

State	Predictor	Indicator
Montana	0.60	0.10
Nebraska	0.50	0.00
Nevada	0.90	0.40
New Hampshire	0.80	0.50
New Jersey	0.70	0.60
New Mexico	0.80	0.50
New York	0.70	0.70
North Carolina	0.70	0.20
North Dakota	0.50	0.00
Ohio	1.00	0.50
Oklahoma	0.50	0.00
Oregon	0.60	0.70
Pennsylvania	0.80	0.70
Rhode Island	0.60	0.90
South Carolina	0.60	0.10
South Dakota	0.50	0.00
Tennessee	0.80	0.30
Texas	0.60	0.10
Utah	0.50	0.00
Vermont	0.70	0.60
Virginia	0.70	0.20
Washington	0.60	0.70
West Virginia	0.60	0.40
Wisconsin	0.70	0.90
Wyoming	0.50	0.00

For all other states, we used historical voting trends to predict the vote. We predicted Donald Trump to secure 228 electoral votes based on historical trends and Hillary Clinton to secure 123 electoral votes based on historical trends. This left 181 electoral votes to be decided by key states.

For elections in which the incumbent party changed, we found that the probability of a Republican candidate winning a key state when there was an increase in the growth rate of disposable income was 53.85 percent. We considered this margin too close to 0.5 to be able to conclusively say that a vote for a Republican candidate can be indicated by an increase in the growth rate. However, the probability that a Democratic candidate winning a key state when there was a decrease in the growth rate was 84.62 percent. We applied the data to the change in growth rate from 2012 to 2016 and found that this would push California, Connecticut, Iowa, Maine, Michigan, New Hampshire, New Jersey, and Vermont's votes blue. There was an increase in growth rate for Florida, Illinois, and Nevada, so we could not conclusively say which way this factor would influence these states' votes.

By using our system of weighted averages, we found that California, Connecticut, Illinois, Iowa, Maine, Michigan, New Hampshire, New Jersey, and Vermont would vote for a Democratic candidate. Florida, Nevada, New Mexico, and Ohio were predicted to vote for a Republican candidate. As seen from Table 2, not every factor could be considered for every state. This was when the indicator was 0.5 or the disposable income growth rate increased. In those cases, the weighted average was considered in reference to the total weight value rather than 1. No matter what, the GDP will affect a state's vote so there was no way for there to be a tie.

Table 2. Voting Predictions for Key States

		GDP	Income	Indicator			
	Weight	0.35	0.25	0.4	Average (1)	Average (0.75)	Average (0.6)
State	California	0	1	1	0.65	*	*
	Connecticut	1	1	1	1	*	*
	Florida	0	*	0	*	0	*
	Illinois	0	*	1	*	0.4	*
	Iowa	0	1	1	0.65	*	*
	Maine	1	1	1	1	*	*
	Michigan	1	1	1	1	*	*
	Nevada	0	*	0	*	0	*
	New Hampshire	1	1	*	*	*	0.6
	New Jersey	0	1	1	0.65	*	*
	New Mexico	0	1	*	*	*	0.25
	Ohio	0	1	*	*	*	0.25
	Vermont	1	1	1	1	*	*

* Indicates N/A

Using our method of prediction, we forecasted that Donald Trump would win the election with 286 electoral votes by a margin of 34 votes.

Discussion

There are many factors that influence a presidential election, many more than our model contains. Our model offers a simplistic view of a few key factors, but only for previously identified swing states. In the actual election, although we got the result right, we did incorrectly predict the voting behavior of six states: Colorado, Iowa, Michigan, New Mexico, Pennsylvania, and Wisconsin. Donald Trump actually won with 306 electoral votes and a margin of 74 votes. We were able to get the right outcome, but not every state was accurately accounted for, indicating that our model will not necessarily be successful in future elections. However, the polling models we first examined were generally unsuccessful in predicting this election last year when they had been successful in the past. This suggests that neither polling data nor historical data is enough to accurately predict the outcome of the election. Another thing that can be improved for our model to perform more accurate predictions may be to adjust our criteria for the indicators and predictors. Our cutoff for within one tenth of 0.5 for the indicator and 3 tenths from 1 for the predictor could be adjusted to maximize accuracy.

The biggest worry for our predictions is that the sample size maybe too small. From 1960 to 2012 there were only 14 elections. Because some data was unavailable before 1976, we needed to reduce our sample size to 10 elections. The patterns seen in the data may therefore change if more election years were included which could sway the predictions made. Additionally, there are many factors that influence how people, states, and the nation as a whole will vote. The influence of these factors may not be consistent election after election and no single factor has more than a moderate influence on voting behavior. Because there are so many factors and no one factor is strongly correlated with election data, there is the chance that we may have exaggerated patterns. If the model we used in this project were to be applied to future elections, we would like to compare the results gained from the key states of this election to random groups of states to ensure the patterns seen are not wholly reliant on the key state sample. Adding the effect of additional factors would also help make the model more accurate.

Conclusion

Our analysis of historical data allowed us to correctly predict the outcome of this year's presidential election. However, adding the actual voting behavior of states from this year's election could have an impact on the key states we identified previously. Our model is not robust, and therefore may not be accurate in future elections. The combination of historical and polling data may produce an even more accurate model. The strength of our model is that it does not rely on possibly biased sampling or polling, however, it dehumanizes the candidates for both parties. The public may favor one candidate over another for some reason that is not reflected in the economy, such as appearing trustworthy. For this year's election, this may

have benefitted our data out due to possible media bias. A model based on polling may be able to pick up where our model cannot cover and help steer a final prediction towards a more accurate choice.

Bibliography

"Historical Presidential Elections." *270towin.com.* Last modified 2016. http://www.270towin.com/historical-presidential-elections/timeline/margin-of-victory/.

Hoban, Brennan. "Why are swing states important?" *Brookings Institution*. Last modified September 28, 2016. https://www.brookings.edu/blog/fixgov/2016/09/28/why-are-swing-states-important/.

Newkirk II, Vann R. "What Went Wrong With the 2016 Polls?" *The Atlantic*. Last modified November 9, 2016. http://www.theatlantic.com/politics/archive/2016/11/what-went-wrong-polling-clinton-trump/507188/.

Peters, Gerhard and John T. Woolley. "The American Presidency Project." *University of California, Santa Barbara*. Last modified 2016. http://www.presidency.ucsb.edu/.

Rothschild, David. "Understanding How Polls Affect Voters." *The Huffington Post*. Last modified December 26, 2012. http://www.huffingtonpost.com/david-rothschild/understanding-how-polls-affect-voters_b_2009034.html.

Siebeneck, Todd and Catherine Wang. "Gross Domestic Product by State: First Quarter 2016." *Bureau of Economic Analysis.* Last modified July 27, 2016. http://www.bea.gov/newsreleases/regional/gdp_state/2016/pdf/qgsp0716.pdf.

Silver, Nate. "Which Economic Indicators Best Predict Presidential Elections?" *The New York Times*. Last modified November 18, 2011. http://fivethirtyeight.blogs.nytimes.com/2011/11/18/which-economic-indicators-best-predict-presidential-elections/.

"U.S. Real GDP by Year." *Multpl.* Last modified 2016. http://www.multpl.com/us-gdp-inflation-adjusted/table.

"YCharts." *YCharts.* Last modified 2016. https://ycharts.com/.

"2016 Election Forecast." *FiveThirtyEight*. Last modified 2016. http://projects.fivethirtyeight.com/2016-election-forecast/?ex_cid=rrpromo.

MATH 3320

Group 13: Melvin Du, Thomas Plantin, Rodrigo Zurita

Project's Report Feedback

Required Sections	Maximum Grade	Your grade	My comments
Introduction	15	15	Well written
Statement of the Problem	25	20	Well done in general, although there was an entire paragraph that belonged in the conclusion.
Methodology	25	25	Well explained. You cover all the key points in your statistical analysis. You did not explain clearly why a gaussian noise is preferred. You could also test a different distribution for noise just to see how things work.
Results	25	25	Well explained as well.
Bibliography	10	10	Well done.
Other comments			
Total	100	**95**	

Project's Presentation Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Be properly attired	5	5	Adequate for the subject chosen
Comments of another presentation	35	30	Well done
Chemistry among group members	20	20	It was very entertaining and shone a light on the project
Timely submission of the presentation	20	20	You submitted your milestones on time
Pass all 4 Milestones	20	20	Well done
Other comments			I like the effort put to edit the video to make it entertaining.
Total	100	**95**	

Kalman Filter GPS Tracking Algorithm

Autonomous Drone Navigation in 2D Cartesian Space

Thomas Plantin, tplantin@trinity.edu

Rodrigo Zurita, rzurita@trinity.edu

Mel Du, mdu@trinity.edu

Introduction:

Autonomous land vehicles/drones, such as self-driving cars, have entered the international spotlight as a viable means of transportation. This presents the challenge of 2-dimensional navigation in non-ideal conditions, where the vehicle GPS readings will be obstructed by interference noise. We have modeled both the drone's location and GPS readings as a Gaussian. This allows for signal processing via a Kalman Filter to predict and update the location of the drone. The implementation of Kalman filtering aims to best predict the future location of a moving object based on multiple readings of its previous placements. If more data is collected from the preliminary trajectory of the object, the prediction of the upcoming location will be more accurate. We have decided to focus on this project because we believe that the future of autonomous navigation lies in Kalman filtering. If we manage to optimize the functionalities of Kalman filters, we could enhance the accuracy of the readings. This means that we could potentially live in a society where only self-driving vehicles would exist, substantially decreasing the number of driving accidents and shortening the travel times. Our group's goal is to achieve the derivation of a GPS location-tracking model that will determine the precise location of the drone using a Kalman Filtering algorithm to eliminate interference noise.

Statement of the problem:

The Global Positioning System relies on readings of the time and location of multiple satellites in order to calculate the three dimensional Cartesian coordinate of a receiver device on Earth. These calculations are so sensitive to errors in measurement that they are only possible by taking into account general and special relativity. While Military GPS systems can theoretically calculate position within tens of centimeters, in practice, they suffer from interference noise. This can be caused by interference sources such as surface conditions, receiver quality, and the atmosphere.[1] For estimation of the true values, it can be assumed that the noise has a Gaussian distribution. Our goal is to estimate the true values of the x and y coordinates using a Kalman filter.

A Kalman filter is a mathematical algorithm that uses previous measurement data to estimate true values in real time. Also known as Linear Quadratic Estimation, this algorithm is used to correct statistical noise in measurements. These properties make it well suited for GPS signal processing. The resulting estimation is an output depicting where the GPS receiver is most likely to be. This is known as a Joint Probability Distribution.[1]

What we want to achieve:

In our project, both the noise and distribution are modeled as Gaussians. The location of the drone is modeled in a two dimensional Cartesian plane, with x and y coordinates. We are using MATLAB to model the kinematics of the drone, as well as to run the Kalman filter algorithm and output estimated coordinates as a function of time. Since this is a simulation, the estimated coordinates can be evaluated in comparison to the true location of the drone.

Methodology:

Kalman Filter

The Kalman filter determines the drone's location, whether the coordinates are detected or not. If the drone coordinates are detected, the Kalman filter first predicts its state at the current time. The filter then uses the newly detected location to correct the state, producing a filtered location. If the sensor data is missing, the Kalman filter solely relies on its previous state to predict the object's current location.

This process involves the use of probability and statistics in order to predict the new position of the moving object with a certain percentage of confidence level. The more values are read by the filter, the higher this percentage of confidence level becomes. However, if a measurement is missed, then the confidence level percentage will start dropping.

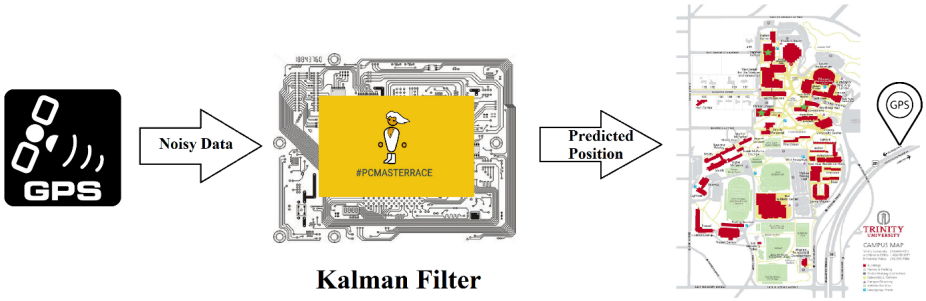


Figure 1: Noisy GPS signal is digitally filtered to produce an accurate estimate of true location.

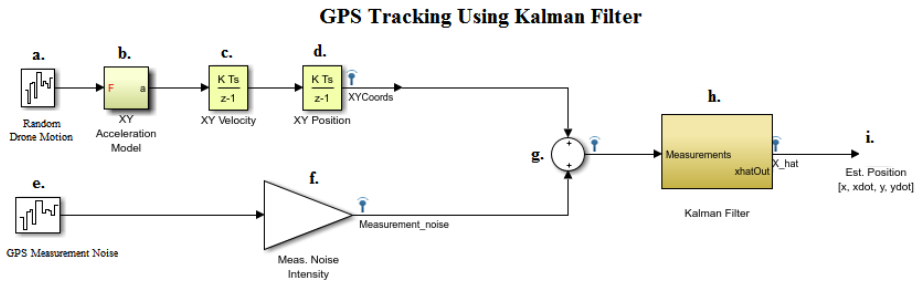


Figure 2: Flow chart depicting how the system is modeled [2].

- **a.** Drone starts at origin with 0 velocity and fixed acceleration
- **b.** Simulation begins modeling kinematics of drone
- **c.** True velocity value is recorded
- **d.** True position value is recorded

e. Noise is modeled with Gaussian distribution
f. Intensity of noise is set
g. Noise is introduced to true values and the sum is inputted to the Kalman filter
h. Kalman filter algorithm: Prior knowledge of state (drone location) is used to predict where the drone will be located. This is then updated with the current state measurement.
i. Kalman filter produces a Multivariate Normal Distribution of the drone's true position. Position of highest probability is graphed as the estimate.

The kinematics of the drone, the noisy GPS signal, and the Kalman-filter signal processing are all simulated in MATLAB. To find the true location of the drone, we used this discrete time linear dynamic system with the following state and measurements equations.

Required MATLAB Add-ons: Signal Processing Toolbox, DSP Toolbox

Step 1. Modeling of Drone Location:

The following states will be used to describe the position of the drone with respect to time:

- X coordinate (**x**)
- Rate of change of x-coordinate (**ẋ**)
- Y coordinate (**y**)
- Rate of change of y-coordinate (**ẏ**)

State Equation:

$$\begin{bmatrix} x(k) \\ \dot{x}(k) \\ y(k) \\ \dot{y}(k) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k-1) \\ \dot{x}(k-1) \\ y(k-1) \\ \dot{y}(k-1) \end{bmatrix} + \mathrm{w(k\text{-}1)}$$

Expands to:

$$x(k) = x(k\text{-}1) + \dot{x}(k\text{-}1) + w(k\text{-}1)$$

$$y(k) = y(k\text{-}1) + \dot{y}(k\text{-}1) + w(k\text{-}1)$$

$$\dot{x}(k) = \dot{x}(k\text{-}1) + w(k\text{-}1)$$

$$\dot{y}(k) = \dot{y}(k\text{-}1) + w(k\text{-}1)$$

The state equation describes the position and velocity of the GPS receiver using kinematics. k is the current time step, and (k-1) is the previous time step. The noise is additive to the position and velocity, and is represented by w(k). These equations use the state of the previous time step to predict the current time step state, taking noise into account. This allows the Kalman filter to estimate the GPS receiver's position, even if the data is blocked off by an obstruction.

Measurement equation: z(k) = H * x(k) + v(k)

$$\begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \\ y(k) \\ \dot{y}(k) \end{bmatrix} + \mathrm{v(k)}$$

Expands to:

$$z_1(k) = x(k) + v(k)$$

$$z_2(k) = y(k) + v(k)$$

In the observed x position, $z_1(k)$ is equal to position x(k) plus noise v(k). In the observed y position, $z_2(k)$ is equal to position y(k) plus noise v(k). In a system without noise, the state and the measurement equations are equal. These equations use measured values and update the current state estimate by modifying the predicted state values. By recursively predicting and updating, the Kalman filter can create an accurate model of the position of the GPS receiver. The longer this algorithm runs, the more accurate this model becomes.

In the MATLAB code for this model, the drone's initial conditions are at the origin with a velocity of [0, 0]ft/s and constant acceleration of [4, -2]ft/s^2. The simulation will then model the kinematics of the drone through position and velocity values.

Step 2. Define Noise Intensity

$$P(w) \sim \mathcal{N}(0, Q)$$
$$P(v) \sim \mathcal{N}(0, R)$$

Noise is summed with the signal, which function is taken as an approximate normal distribution, or Gaussian. Using this model, the mean (μ) is set to 0 for both disturbance equations. Q is set to $5*10^5$ for the variance of noise in x position divided by the number of samples from the

MATLAB simulation, and R is set to $5*10^5$ for the variance of noise in y position divided by the number of samples obtained.

Step 3. Kalman Filter

The Kalman filter algorithm computes the following two steps:

1. Prediction: Process parameters x (state) and P (state error covariance) are estimated using the previous state.[2]
2. Correction (or update): The state and error covariance are corrected using the current measurement.[2]

Results

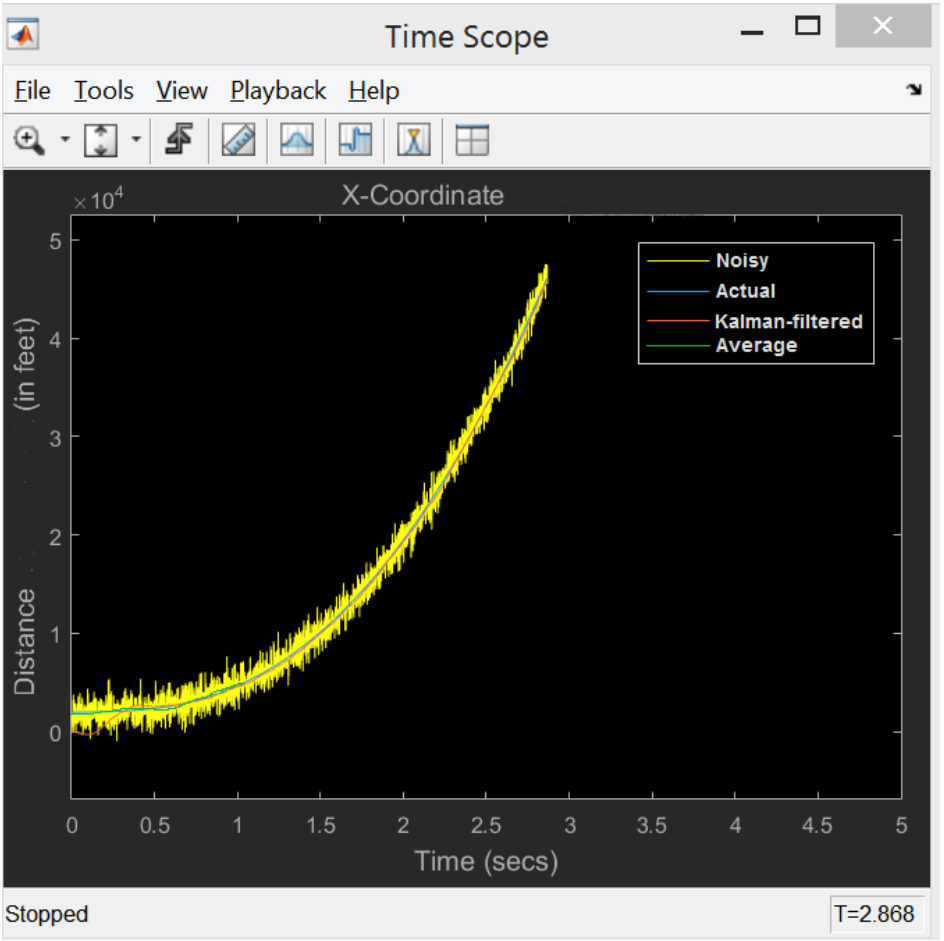


Figure 3: Graph of x position as a function of time. The true x position is shown in blue. The x position summed with noise is shown in yellow. The Kalman filter estimation IS shown in orange. It can be observed that the estimation becomes much more accurate as more previous data becomes available to the algorithm. The average of the noisy signal, shown in green, is very accurate due to the Gaussian distribution of the noise.

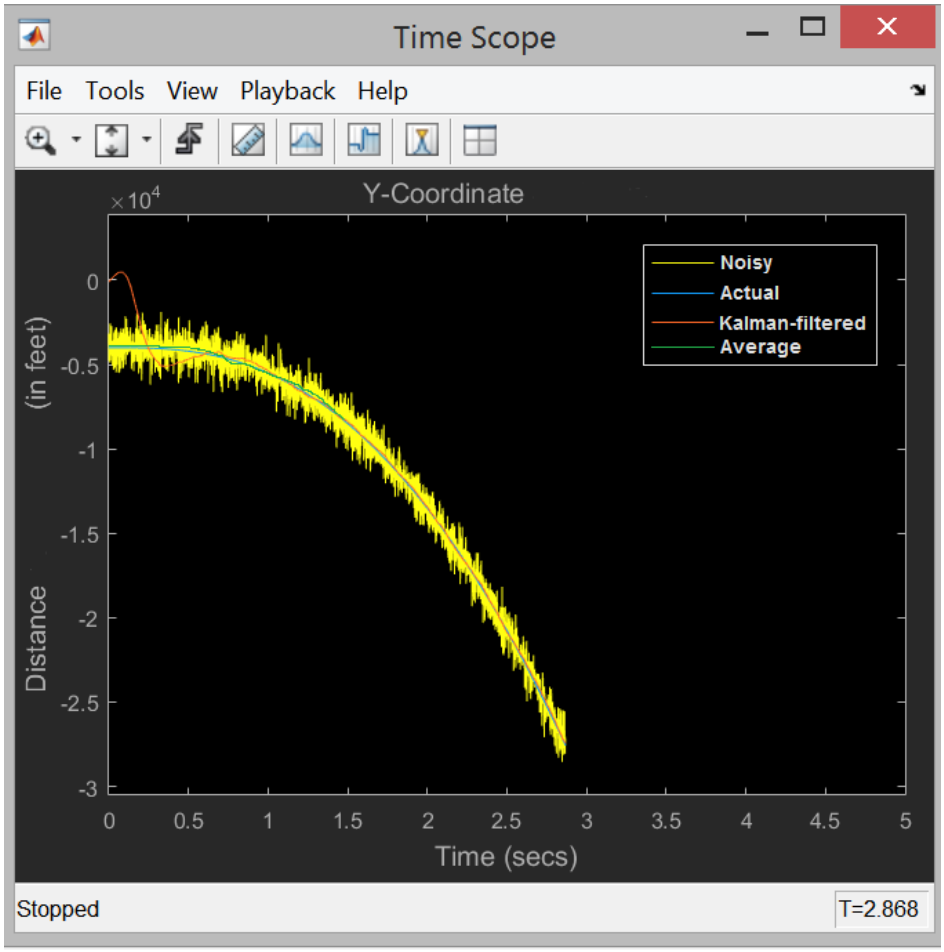


Figure 4: Graph of y position as a function of time. The true y position is shown in blue. The y position summed with noise is shown in yellow. The Kalman filter estimation is shown in orange. It can be observed that the estimation becomes much more accurate as more previous data becomes available to the algorithm. The average of the noisy signal, shown in green, is very accurate due to the Gaussian distribution of the noise.

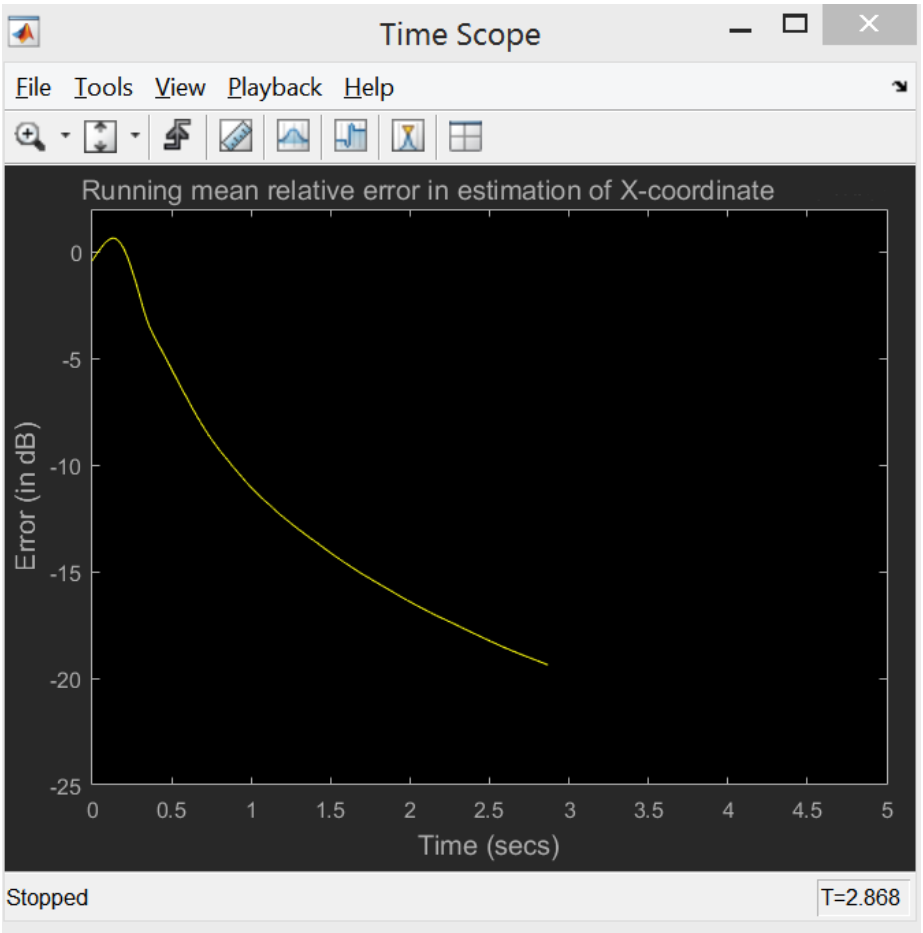


Figure 5: Logarithmic graph of mean relative error in y position as a function of time. Again, it can be seen that as more previous data becomes available, the model of location becomes vastly more accurate. This is because the Kalman filter relies on previous states in order to make accurate predictions.

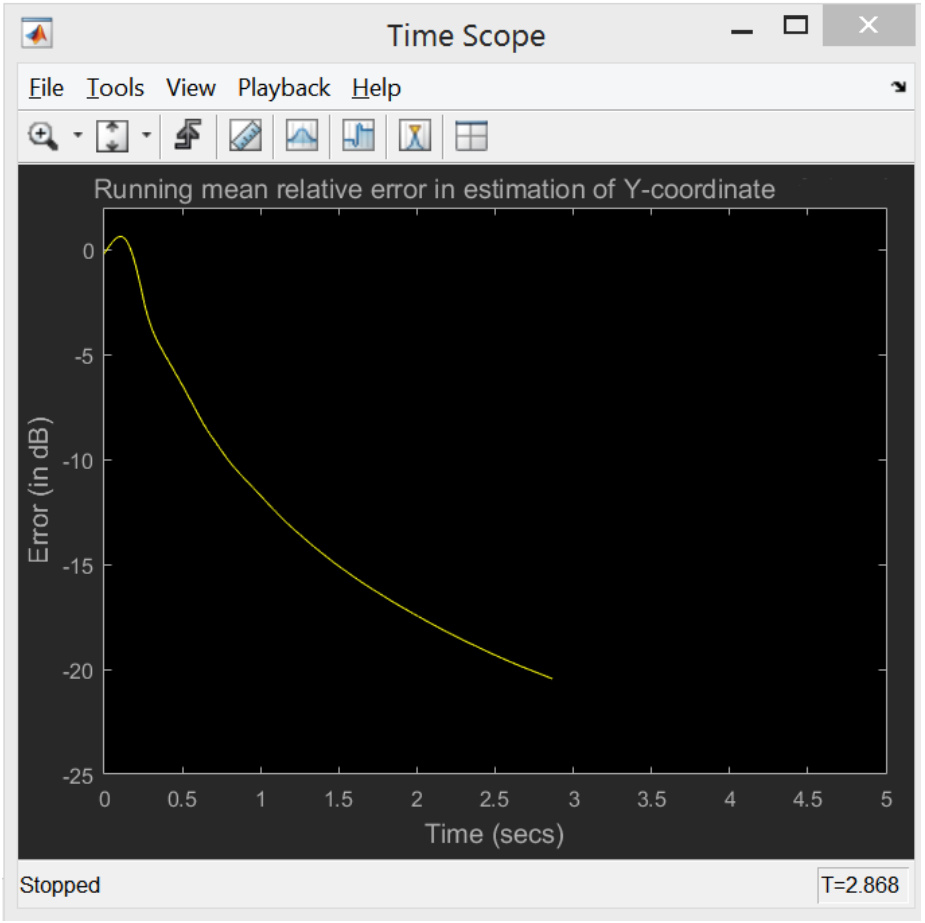


Figure 6: Logarithmic graph of mean relative error in y position as a function of time. Again, it can be seen that as more previous data becomes available, the model of location becomes vastly more accurate. This is because the Kalman filter relies on previous states in order to make accurate predictions.

One challenge is that our data was limited to simulation via mathematical models. An applied study would require a real drone and GPS. This would also mean that the noise isn't necessarily Gaussian, due to multiple interference sources. With an increase in our capital budget, we could have collected real measurements to compare to our simulations in MATLAB. Moreover, our results could improve by increasing the number of MATLAB generated samples. We could also have used different distributions of noise, such as Weibull or Lognormal distributions as additive disturbances to the collected data.

References:

[1] Borsa, A. A., Minster, J., Bills, B. G., & Fricker, H. A. (2006). Modeling long-period noise in kinematic GPS applications. Journal of Geodesy, 81(2), 157-170. doi:10.1007/s00190-006-0097-x

[2] Estimating Position of an Aircraft using Kalman Filter. (n.d.). Retrieved October 22, 2016 from https://www.mathworks.com/help/ Copyright 1995-2015 The MathWorks, Inc.

[3] Greg Welch and Gary Bishop, "An Introduction to the Kalman Filter"

TR95-041, University of North Carolina at Chapel Hill.

MATH 3320

Group 14: Parker Cormack, Christian Oakes, Samuel Neely

Project's Report Feedback

Required Sections	Maximum Grade	Your grade	My comments
Introduction	15	15	Well written
Statement of the Problem	25	20	Well done in general, although there was an entire paragraph that belonged in the conclusion.
Methodology	25	25	Well explained. You cover all the key points in your statistical analysis, except for that little blunder with the log-normal distribution.
Results	25	25	Well explained as well.
Bibliography	10	10	Well done.
Other comments			I like the depth of your analysis. I like your findings.
Total	100	**95**	You could mention attempts you made to deal with large values of n, like using the Sterling Formula etc.

Project's Presentation Feedback

Required Sections	**Maximum Grade**	**Your grade**	**My comments**
Be properly attired	5	5	Well done. Adequate for the subject chosen
Comments of another presentation	35	30	Well done
Chemistry among group members	20	20	Well coordinated
Timely submission of the presentation	20	20	You submitted your milestones on time
Pass all 4 Milestones	20	20	Well done
Other comments			Your video was a bit blurry but the point made was clear.
Total	100	**95**	

Efficiency Analysis for Gas versus All-Electric Vehicles

MATH 3320

Parker Cormack, Christian Oakes, and Sam Neely

Introduction:

Fuel and energy efficiency has become a major factor in both production and purchasing of automobiles. The goal of this analysis is to observe the factors that may affect efficiency and cost in both gas and electric vehicles. As we analyze fuel efficiency for the 1205 gas vehicles and 30 electric vehicles in our data sample, we hoped to find trends in our data suggesting possible distributions that define the data sample. In particular, we analyzed the data and statistics regarding the fuel efficiency of both gas and all-electric vehicles as a function of several different variables. These variables include the number of cylinders and displacement of the engines in gas cars, the retail price of both classifications of automobiles, and the maximum range a car can travel on full fuel capacity. For gas cars, we observed that fuel efficiency generally follows a lognormal distribution. As for the electric vehicles analyzed, we found the fuel efficiency across the sample to display the characteristics of a normal distribution. We then wanted to compare these relationships in gas and electric vehicles to evaluate which types of vehicles yield a better value for buyers in the 2016 production year.

With regard to the data used for this analysis, our group used the government published list of all 2016 production gas and electric vehicles in the United States. The government published this list along with the general specifications and EPA estimated values for each vehicle. Using such a wide variety of vehicles, we were able to maintain an element of randomness in our data since each model of a car that a company produces has different general specifications than other models that same company or other companies produce.

Our overall conclusion is that there are certain trends in gas car efficiency that correlate to both price and engine displacement. A model was built in which the price, fuel efficiency, or engine displacement could be found given one of the three values initially.

Methodologies:

We obtained the majority of our data from www.fueleconomy.gov, as well as several independent manufacturers' sites. Our data includes EPA estimated combined highway and city fuel efficiency, the number of cylinders in each gas vehicle engine, retail price of each vehicle, the engine displacement and battery size for gas and electric vehicles respectively; from this data we also calculated maximum range for several vehicles.

Prior to analyzing the effects of various factors on fuel efficiency, we wanted to observe the general behavior and trends of combined highway and city fuel efficiency using conventional fuel for both gas and electric vehicles. Using Minitab, we generated histograms displaying fuel efficiency for gas and electric vehicles separately, and used their respective distributions as a reference for the remainder of our analysis. We added distribution fits to each histogram, where applicable, to establish a general trend to each sample. We then found the median fuel efficiency for both gas and electric cars in order to find the best and worst 50 percent of all 2016 model year vehicles.

The first factor we considered that directly affects fuel efficiency in gas automobiles is the number of cylinders in an engine. Since there are mechanical limitations to the number of cylinders in most automobile engines, we separated the data according to the number of cylinders, giving us seven different classifications. Once we separated the data accordingly, we wanted to see if there was a general trend in efficiency according to the number of cylinders in the engines. To view this trend in an appropriate manner, we generated a series of boxplots side-by-side to show any general trend in the relationship.

The next influence on efficiency we considered for gas vehicles was the displacement of the engines in gas cars. Displacement relates to the total swept volume of all the pistons in an engine from their maximum to minimum volumes, but is independent of the number of cylinders in an engine. Our group wanted to show that there was a negative correlation between the number

of cylinders in gas vehicles and the combined fuel efficiency of the cars. Plotting the fuel efficiency based on the number of cylinders, we found a trend in the data—we then generated a non-linear fit to the data, which provided our group with the first of the governing equations that characterize our model for finding the expected efficiency of a gas vehicle given engine displacement. To test the validity of this governing equation, we also calculated the coefficient of determination for this non-linear fit using Minitab.

In further analysis of our data, we plotted the MSRP of gas and electric vehicles on separate histograms. We fit the histogram displaying gas vehicle price with a lognormal trend line and this fit gave the distribution of the different prices of all gas vehicles sampled. The histogram for electric vehicle did not have a distribution where a trend line would be applicable. We validated this claim with a Q-Q plot. Given a larger data set for electric vehicles, this may have been possible.

To find our second governing equation, we plotted the MSRP of gas vehicles and their efficiency. Applying a non-linear fit to the plot, we now have a way to calculate an expected efficiency based on the MSRP of a 2016 gas vehicle. This is applicable to a consumer purchasing a car and knowing what efficiency they can expect from their vehicle, given the consumer's purchasing power. We also plotted the MSRP of electric vehicles against their respective efficiencies, but the sample size was not large enough to show any correlation between the two.

In considering the relationship between efficiency and range of both electric vehicles and gas vehicles, we had to incorporate a method, which would accurately determine the predicted range for both types of vehicles. The government data provided in [3] provided the range for every type of electric vehicle used by consumers in the United States. However, the process for calculating the range for all of the gas engines was a more daunting task; [3] provided us with the combined highway and city fuel efficiency of every car we desired to analyze, but we had to research tank size individually to calculate maximum range. Using (1), we were able to compile data for range for gas automobiles.

$$\left(\frac{Miles}{Gallons}\right) * (Tank\ Size(Gallons)) = Max\ Range\ (Miles) \tag{1}$$

As we progressed with our analysis, we compared the range and fuel efficiency of both gas and electric vehicles to analyze any trends that were present.

Results:

After analyzing our data, we were able to derive a set of equations that act as a model to find the expected efficiency (in miles per gallon) for 2016 gas production vehicles, given either the price of the car or displacement of the engine. Vice-versa, if we know the efficiency of a vehicle, our model can also predict the values for price and engine displacement. Using the histogram in Figure 1, we can validate the general accuracy of our model. Displayed below are the governing equations for our gas car efficiency model:

Given MSRP:

$$E(x) = 304.207(price)^{-0.240397} \tag{2}$$

Given engine displacement:

$$E(x) = 43.38 - 10.57(disp) + 1.435(disp)^2 - 0.07219(disp)^3 \tag{3}$$

The range of a gas vehicle has a weak positive correlation to the fuel efficiency showing that more fuel-efficient cars do not necessarily travel farther than inefficient vehicles on one tank

of gas, because manufacturers can compensate for poor fuel efficiency by increasing the size of the gas tank. Based on our Kelly Blue Book's average dollar amount spent on new cars in 2016 of $33,340, we calculated the expected efficiency to be 27.825 MPG, which closely follows the behavior shown in Figure 1 [2]. We were unable to create an adequate model for finding the expected efficiency of electric vehicles because there were two very distinct groupings of data, which created challenges in finding an accurate fit for the data. The range of the electric vehicles seems to be independent of the vehicle's fuel efficiency as well, most likely due to varying battery charging capacities along with the varying battery capacities.

Discussion:

Having evaluated the effects of the aforementioned factors on efficiency, the results we have observed are not particularly surprising. The general trends in efficiency of gas cars as a function of price, engine displacement, and the number of cylinders in the engine are negative. We can validate this statement using our governing equations for finding expected value of efficiency in gas cars, and validate the results using the histogram in Figure 1.

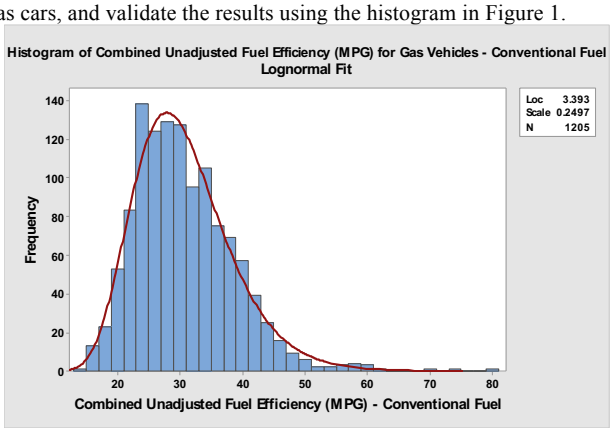


Figure 1: This figure shows the histogram of combined unadjusted fuel efficiency for 2016 production gas vehicles.

Since this histogram is unimodal, right-skewed, and there no significant outliers, we chose a lognormal distribution as a fit for the data characterized by $x \sim LN(3.393, 0.2497)$. Looking at the efficiency of electric cars, we found that the histogram is unimodal as well, but trends more towards a normal distribution than the gas cars. The rough normal fit to the data follows the distribution characterized by $x \sim N(100.6, 129.7321)$.

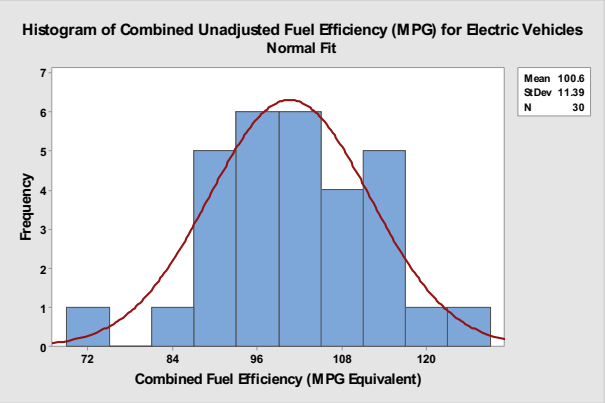


Figure 2: This figure shows the histogram of the miles per gallon efficiency equivalent for 2016 production electric vehicles.

When analyzing the effects the number of cylinders has on the efficiency of gas vehicles, we can see that as the number of cylinders in an engine, the efficiency generally decreases. We chose to represent this data using a series of boxplots to view the trend in the data as the number of cylinders in an engine increases. This group of boxplots shows the general negative trend in efficiency as the number of cylinders in a gas engine increases while also showing outlier behavior and median values for each cylinder classification.

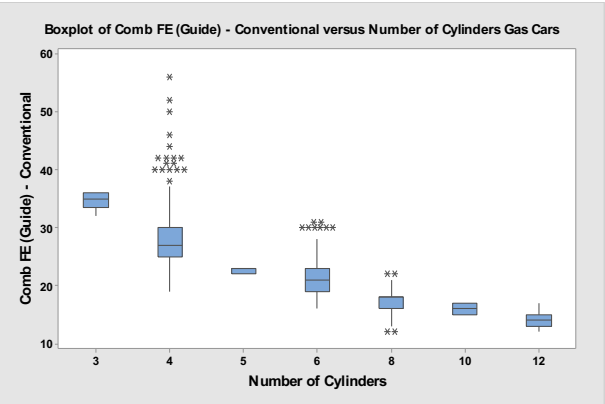


Figure 3: This figure shows a series of boxplots representing the efficiency data for each classification of number of cylinders in gas car engines.

In continuation of our analysis of the factors affecting fuel efficiency, we observed engine displacement versus fuel efficiency in gas cars. Figure 4 shows that there is a negative relationship with a moderately strong negative correlation between displacement and fuel efficiency, given by a coefficient of determination of 0.65.The general trend followed the pattern that as displacement of a gas engine increases, fuel efficiency decreases. We used this plot to derive (3), the second governing equation of our gas efficiency model.

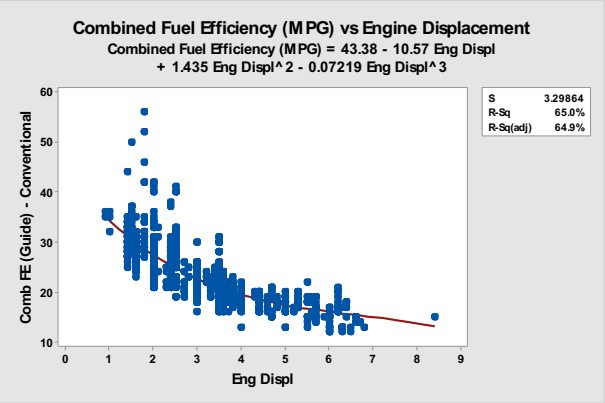


Figure 4: This figure shows the scatterplot of combined fuel efficiency as a function of engine displacement in liters.

Evaluating the range of the gas vehicle costs in the data set we were using we eliminated the high-end super and hyper cars since these cars are unrealistically expensive for the average consumer and the cost of these base models is not readily available without further inquiry. This data compiled in the histogram of frequency versus MSRP of gas vehicles shows a unimodal right skewed plot with the majority of the vehicles being in the $20,000 to $60,000 range shown in Figure 5. We then applied a lognormal fit to the histogram to show the trend of the MSRP of gas vehicles and found that the lognormal distribution fit the data more closely. We attempted to fit the data with a Weibull distribution, but the Weibull did not accurately encapsulate the height of the distribution. The lognormal distribution fails to capture the true mode of the data set but more accurately represents the mean.

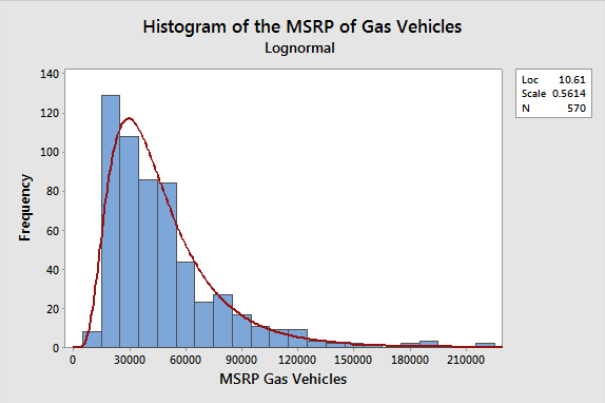


Figure 5: This figure shows the histogram of the MSRP for 2016 production gas vehicles.

The MSRP of electric vehicles appears to have a bimodal distribution with the modes centered on $30,000 and $75,000 plotted in Figure 6. This is a more reasonable price range for the average customer when compared to the higher end of the spectrum.

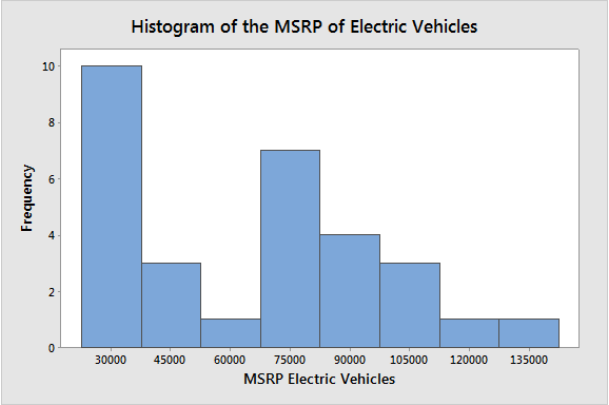


Figure 6: This figure shows the histogram of the MSRP for 2016 production electric vehicles.

Analyzing the MSRP of electric vehicles in a Q-Q plot, Figure 7, shows that the distribution is indeed bimodal characterized by the S shaped curve around the trend line. This makes it hard to give a single accurate mean value for a consumer to compare costs against unlike with the gas vehicles.

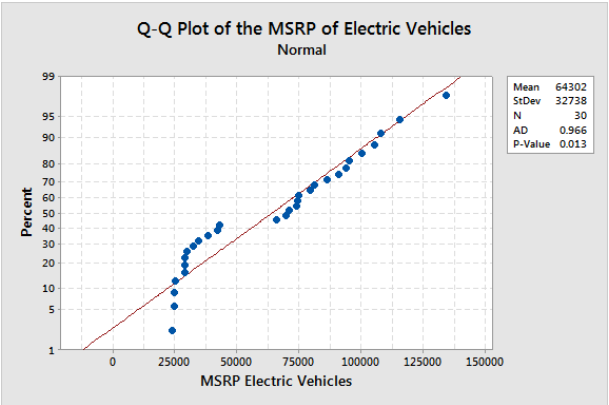


Figure 7: This figure shows the Q-Q plot for the MSRP of 2016 production electric vehicles.

The fuel efficiency as it compares to the MSRP of various gas powered vehicles, found in Figure 8, is right skewed and unimodal around $25,000. This shows that in general a more expensive vehicle does not necessarily correlate to a more efficient vehicle. This is very important for a consumer to know because it shows that if they want a very efficient vehicle they do not have to pay very much.

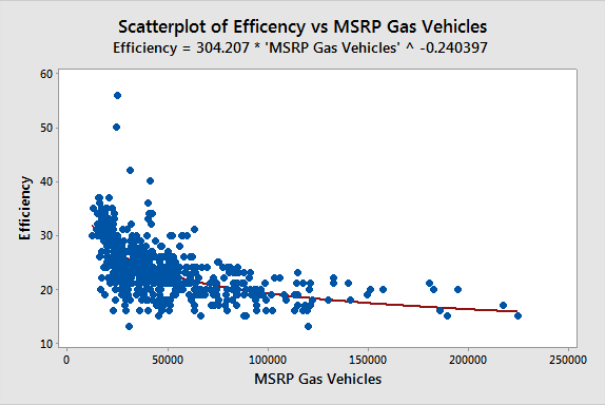


Figure 8: This figure shows the scatterplot of combined fuel efficiency as a function of MSRP for 2016 production gas vehicles.

Comparing the MSRP of electric vehicles to their efficiency in Figure 9, there is no direct correlation to an increase or decrease in price to an increase or decrease in efficiency. This means the only distinctions between electric vehicles are the features that they are equipped with; these features directly correlate to the prices of these vehicles. The efficiency in this case remains relatively constant and does not vary even when the cost of the vehicles ranges from $23800 to $134500.

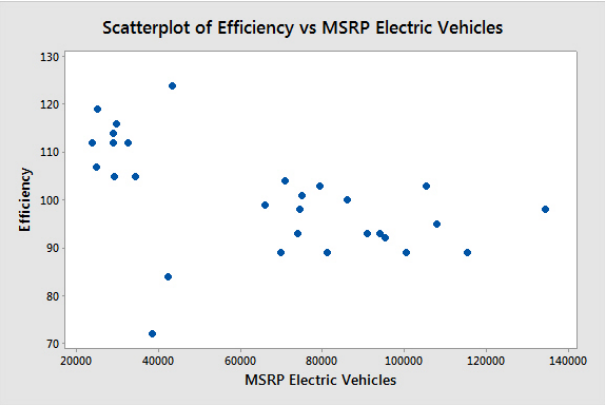


Figure 9: This figure shows the scatterplot of combined fuel efficiency as a function of MSRP for 2016 production electric vehicles.

For the gas vehicles, the data trends towards a slight increase in range as the efficiency increases as seen in Figure 10. This implies that the range of the gas vehicles may depends on the fuel efficiency of the car.

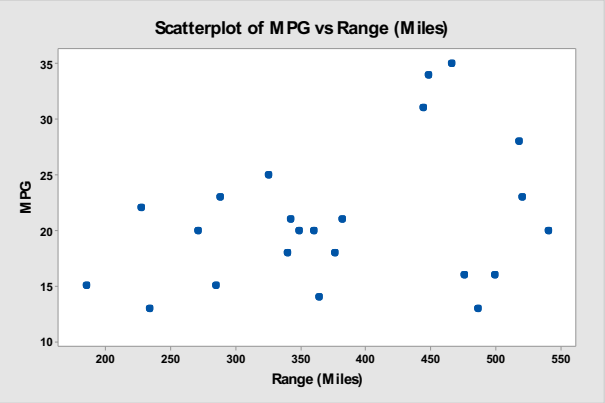


Figure 10: This figure shows the scatterplot of combined fuel efficiency as a function of range for 2016 production gas vehicles.

For the electric vehicles, the fuel efficiency of the cars is generally the same as displayed in Figure 11. The difference in range is most likely due to the battery that powers the car. There is a subtle difference in efficiency; however, the increase in efficiency does not correlate to a significantly larger range capability.

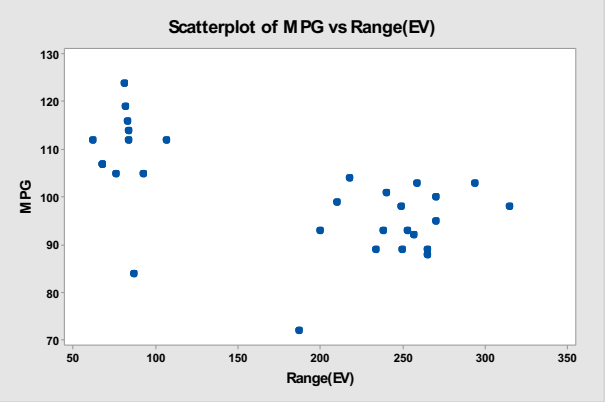


Figure 11: This figure shows the scatterplot of combined fuel efficiency as a function of range for 2016 production electric vehicles.

One of the main reasons behind that fact that fuel efficiency of electric vehicles is independent of range is due to the varying battery capacities of the cars, Figure 12. Vehicles with larger maximum capacity may not necessarily have a high efficiency but they may still obtain a greater range than more efficient vehicles with smaller battery sizes. In terms of the capacity of the battery, drivers never fully exhaust all of the power that may be available on a full charge. This means the estimated ranges of many vehicles is greater than what they are in practice. Additionally, much of the range data provided by manufacturers is idealized because it does not take into account an individual consumer's driving tendencies. Depending upon variables such as temperature and the driver's aggressiveness, the range of each electric vehicle changes drastically.

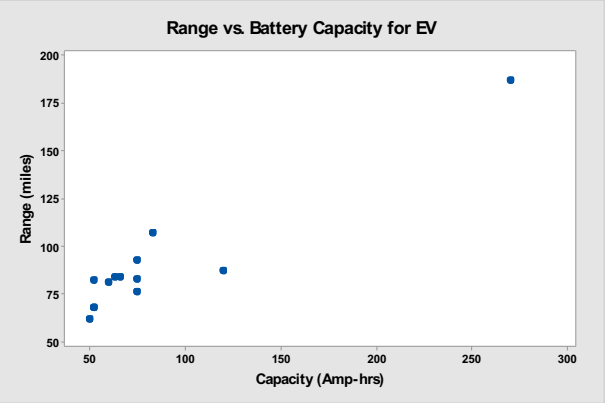


Figure 12: This figure shows the scatterplot of range as a function of battery capacity for 2016 production electric vehicles.

Using the data trends we were able to show an analysis of cars using the median U.S. annual income as the value for the MSRP [6].

We know the median U.S. annual income = $51939 (per Household). Given that *x* is the MSRP of a 2016 production gas vehicle. We want Pr(x < 51939) which will yield the proportion of cars that are priced less than the median U.S. annual income.

Given that:

$$x \sim Ln\ (10.61, 0.5614)$$

$$\Pr\ (x < 51939) = \Pr\ (\ln(x < 51939))$$

$$z_{\ln(51939)} = \frac{\ln(51939) - 10.61}{0.5614} = 0.44144$$

$$\Pr(z < 0.44144) = 0.67$$

This means that 67% of 2016 gas vehicles in production in the U.S. cost below the median U.S. annual income of $51939. This is to as predicted, because the majority of consumers are not willing to spend their entire annual income on a car so a majority or 67% of the cars would need to cost lower than median income for the manufacturers to stay in business.

Given the annual income we can also calculate the expected efficiency E(*f*) that a gas vehicle would have given *f* is the MSRP of gas vehicles.

$$E(f) = 304.207(f)^{-0.240397}$$

$$E(f) = 304.207(51939)^{-0.240397} = 22.3655\ MPG$$

We can validate this calculation by looking at the scatterplot of MSRP and efficiency for gas vehicles and, as expected, the value for an MSRP of $51939 falls around 22.4 MPG.

Conclusion:

Our goals for this project were to observe the factors that may affect efficiency and cost in both gas and electric vehicles. In our data analysis, we were able to construct a set of equations that act as a model to calculate the expected fuel efficiency of a gas vehicle based on MSRP or engine displacement. We were unable to attain equations that accurately predict the fuel efficiency of electric vehicles based on MSRP or any other factors. After further analysis, we determined that our model for gas vehicles could be used to tell a consumer interested in purchasing a new vehicle whether or not they are paying an appropriate value for their new

vehicle, given its specified parameters fitting our model. The values we find are expected values, or averages, so they are the boundary values for what makes a good deal for the consumer.

References:

[1] *Compare Prices From Multiple Dealers and get the Lowest Price!* (2016, November). http://dealers.car.com/prices

[2] Deaton, J.P. (2015, July 2). *How Much Should You Spend on a Car?* http://usnews.rankingsandreviews.com/cars-trucks/best-cars-blog/2015/07/How_Much_Should_You_Spend_on_a_Car/

[3] *Download Fuel Economy Data.* (2015, November). https://www.fueleconomy.gov/feg/download.shtml

[4] *New Car Research.* (2015, November). http://www.autoguide.com/new-cars/

[5] *Plugin Cars: Cars.* (2016). http://www.plugincars.com/cars?field_isphev_value_many_to_one=pure+electric

[6] *Real Median Household Income in the United States*. (2016, September 13). https://fred.stlouisfed.org/series/MEHOINUSA672N