BARUCH COLLEGE, CUNY

DEPARTMENT OF STATISTICS

Final Project: STA 9700
Semester: Fall, 2019

Case Study: Decoding the relationship between urban pedestrian walks in NYC

Submitted To:
Prof. Lawrence Tatum

Submitted By:
Tanay Mukherjee
23987468

TABLE OF CONTENTS

NOTE: All the code snippets have been shared with the respective outputs in 'blue' color for easy reading and identification.

Chapter 1: Introduction

1.1. Topic

The subject of my study is to see the relationship between pedestrians on a regular day who passes Brooklyn Bridge from Manhattan. How many of them are going towards Manhattan or coming towards Brooklyn?

Does the regular traffic of pedestrians have any co-relation at all who happen to come from different directions and meet at Brooklyn Bridge? Does it show changes when there is a public holiday or during a particular time of the day. How does a bad weather or a good weather affects the frequency of pedestrians on a given day?

This data set looks very interesting because it is random and unbiased, and we have different variables to compare our analysis.

It interests me because I feel a random walk of life for many random people can teach us so much and the idea that it can still have a relationship – strong or weak shows how profound the subject is, and its implementation is kind of everywhere just that we don't observe much.

1.2. Data Source

The source is https://opendata.cityofnewyork.us/. We have many such interesting datasets and the ones shared here are mostly verified by NYC state and can be used to do some interesting study about the city and its people.

1.3. Variables

Following are the variables and their description:

Column Name	Column Description
Hour_beginning	Date and time of hourly count
Location	Name of site where count was obtained
Pedestrians	Total count (sum of directions)
Towards Brooklyn	Pedestrian crossing towards Brooklyn
Towards Manhattan	Pedestrians crossing towards Manhattan
Weather_summary	Overall daily weather (cloudy, clear, rain, etc.)
Temperature	Hourly temperature, in Fahrenheit degrees
Precipitation	Hourly precipitation, in inches
Lat	latitude
Long	longitude
Events	Holidays

1.4. Total number of observations

The dataset has a total of 500 rows excluding the header.

I have sampled this data from a total of 7000 observations using the random function in excel and chose the top 500 rows selected on arranging the rows by this random number.

This was done to avoid 'ink blot' what we learnt in class and also because too many observations make the relationship between the y and x weak to understand.

1.5. Data View

Show a half-page of the data set, including the column headers.

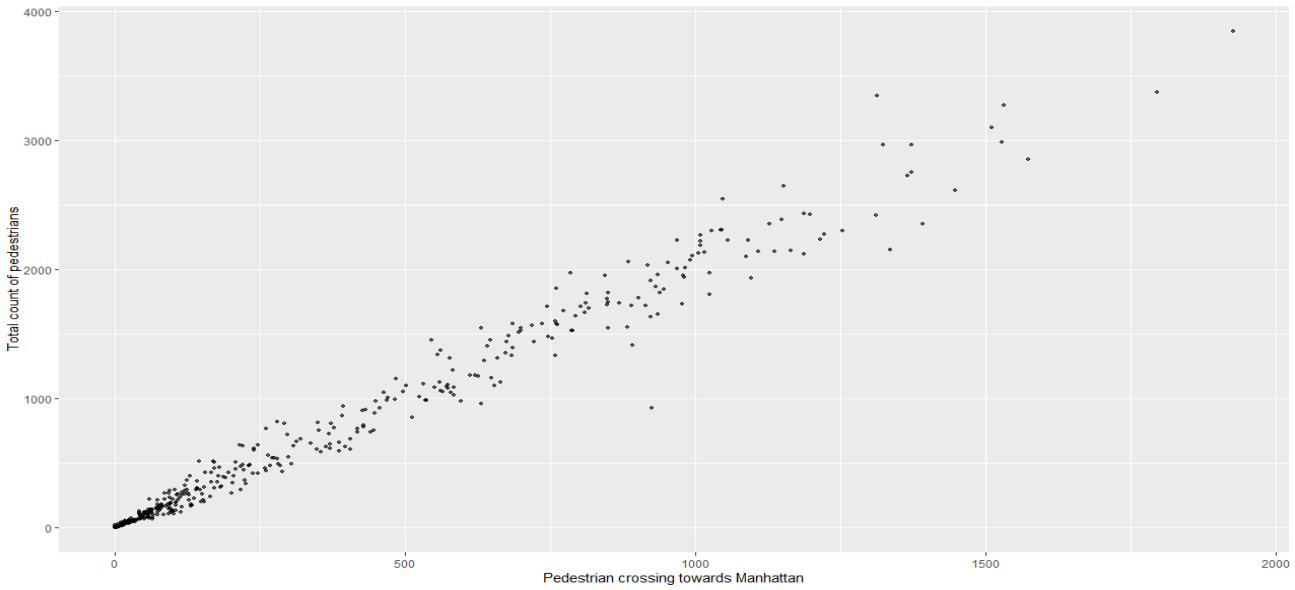
hour_beginning	location	Pedestrians	Towards Manhattan	Towards Brooklyn	weather_s	temperature	precipitation	lat	long	events
11/12/2017 6:00	Brooklyn Bridge	38	23	15	clear-night	30	0	40.70816	-73.9995	
11/29/2017 1:00	Brooklyn Bridge	11	7	4	clear-night	48	0	40.70816	-73.9995	
12/17/2017 22:00	Brooklyn Bridge	168	99	69	partly-clou	35	0	40.70816	-73.9995	
11/10/2017 12:00	Brooklyn Bridge	1213	506	707	clear-day	36	0.0001	40.70816	-73.9995	Veterans D
1/24/2018 11:00	Brooklyn Bridge	535	298	237	partly-clou	40	0	40.70816	-73.9995	
6/10/2018 19:00	Brooklyn Bridge	1027	491	536	partly-clou	64	0.0005	40.70816	-73.9995	
3/16/2018 18:00	Brooklyn Bridge	912	429	483	partly-clou	36	0	40.70816	-73.9995	
4/16/2018 6:00	Brooklyn Bridge	10	7	3	rain	41	0.3067	40.70816	-73.9995	
3/2/2018 21:00	Brooklyn Bridge	16	9	7	partly-clou	35	0.0024	40.70816	-73.9995	
12/30/2017 3:00	Brooklyn Bridge	0	0	0	clear-night	16	0	40.70816	-73.9995	
10/27/2017 21:00	Brooklyn Bridge	225	99	126	clear-night	54	0	40.70816	-73.9995	
11/7/2017 4:00	Brooklyn Bridge	2	1	1	clear-night	42	0	40.70816	-73.9995	
10/3/2017 21:00	Brooklyn Bridge	196	102	94	clear-night	58	0	40.70816	-73.9995	
12/31/2017 15:00	Brooklyn Bridge	1724	915	809	partly-clou	14	0	40.70816	-73.9995	New Year's
2/3/2018 21:00	Brooklyn Bridge	74	50	24	clear-night	28	0	40.70816	-73.9995	
11/25/2017 2:00	Brooklyn Bridge	17	3	14	clear-night	38	0	40.70816	-73.9995	
12/9/2017 9:00	Brooklyn Bridge	480	266	214	partly-clou	34	0	40.70816	-73.9995	
11/13/2017 4:00	Brooklyn Bridge	9	5	4	partly-clou	41	0.0001	40.70816	-73.9995	
1/26/2018 14:00	Brooklyn Bridge	966	455	511	clear-day	35	0	40.70816	-73.9995	
1/18/2018 14:00	Brooklyn Bridge	714	341	373	clear-day	31	0.002	40.70816	-73.9995	
12/6/2017 2:00	Brooklyn Bridge	4	0	4	partly-clou	56	0.001	40.70816	-73.9995	
12/2/2017 13:00	Brooklyn Bridge	2649	1152	1497	clear-day	47	0	40.70816	-73.9995	
1/11/2018 4:00	Brooklyn Bridge	2	0	2	partly-clou	37	0	40.70816	-73.9995	
11/3/2017 17:00	Brooklyn Bridge	2281	1232	1049	partly-clou	72	0	40.70816	-73.9995	
12/18/2017 7:00	Brooklyn Bridge	129	63	66	partly-clou	34	0.0004	40.70816	-73.9995	
11/25/2017 17:00	Brooklyn Bridge	1779	667	1112	clear-night	54	0	40.70816	-73.9995	
12/11/2017 14:00	Brooklyn Bridge	1674	792	882	partly-clou	40	0.0042	40.70816	-73.9995	
12/27/2017 6:00	Brooklyn Bridge	61	35	26	cloudy	19	0	40.70816	-73.9995	
11/28/2017 19:00	Brooklyn Bridge	361	237	124	clear-night	49	0.0005	40.70816	-73.9995	
10/27/2017 5:00	Brooklyn Bridge	22	13	9	clear-night	45	0	40.70816	-73.9995	
1/17/2018 20:00	Brooklyn Bridge	59	38	21	clear-night	25	0.0027	40.70816	-73.9995	

Chapter 2: A Multiple Regression Model with Two Regressors

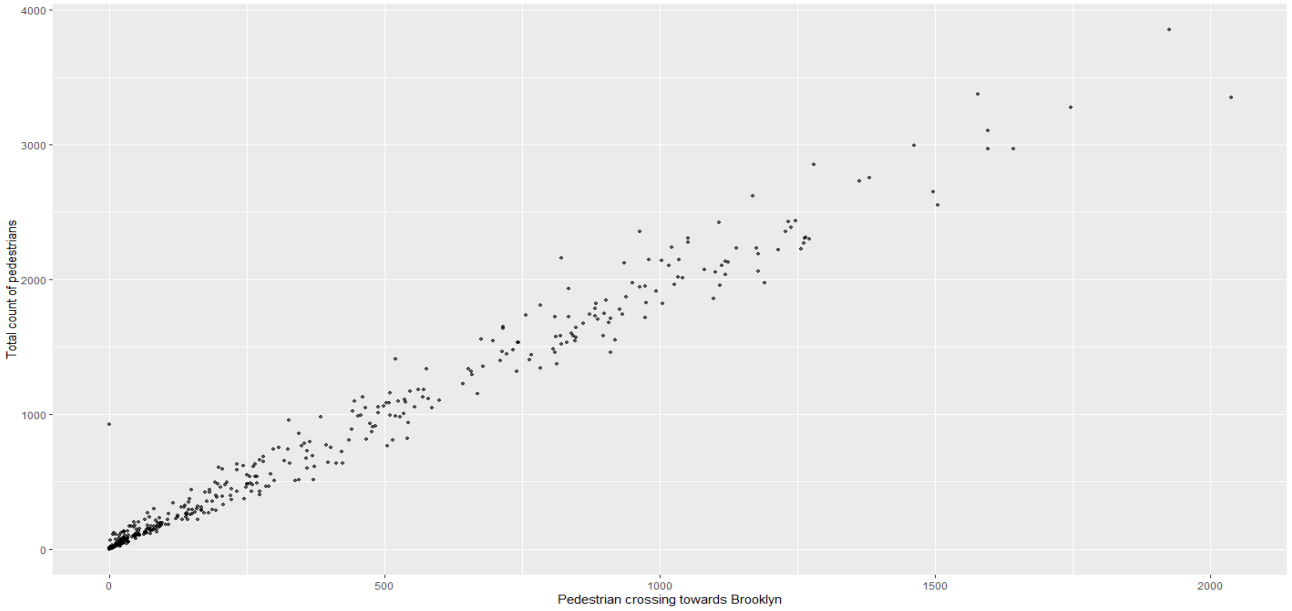
2.1. Scatterplots

Below are the two scatter plots for variables x_1 and x_2 with respect to our y-value which in this case is 'count of pedestrians'.

a) Scatter plot between y = total number of pedestrians and x_1 = pedestrians towards Manhattan



b) Scatter plot between y = total number of pedestrians and x_1 = pedestrians towards Brooklyn Bridge

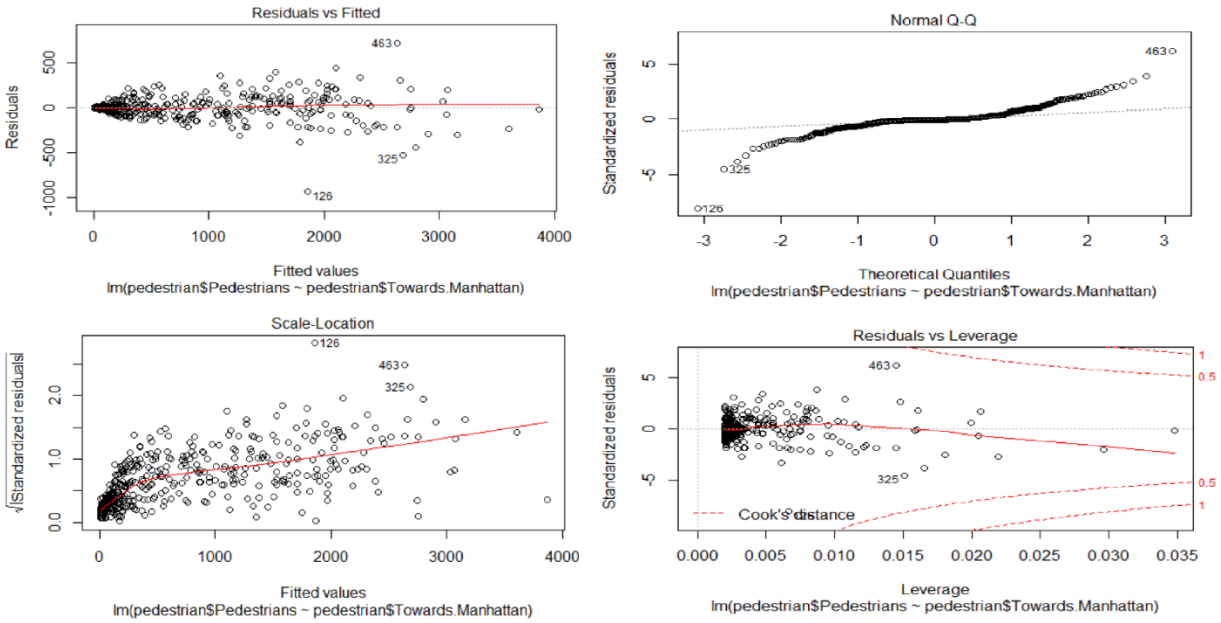


2.2. Analysis of Scatterplots

a) *Scatter plot between y = total number of pedestrians and x_1 = pedestrians towards Manhattan*

I tried to check the various expects by plotting the graph for liner model and it seems very interesting. We have outliers of course but the heteroscedasticity is not there as we see a much flatter line and an evenly distributed residuals in the first plot residuals and fitted values.

```
model<-lm(pedestrian$Pedestrians~pedestrian$Towards.Manhattan)
plot(model)
```



1. Residuals vs Fitted

This plot shows if residuals have non-linear patterns. There could be a non-linear relationship between predictor variables and an outcome variable and the pattern could show up in this plot if the model doesn't capture the non-linear relationship. If you find equally spread residuals around a horizontal line without distinct patterns, that is a good indication you don't have non-linear relationships.

2. Normal Q-Q

This plot shows if residuals are normally distributed. Do residuals follow a straight line well or do they deviate severely? It's good if residuals are lined well on the straight dashed line.

3. Scale-Location

It's also called Spread-Location plot. This plot shows if residuals are spread equally along the ranges of predictors. This is how you can check the assumption of equal variance (homoscedasticity). It's good if you see a horizontal line with equally (randomly) spread points.

4. Residuals vs Leverage

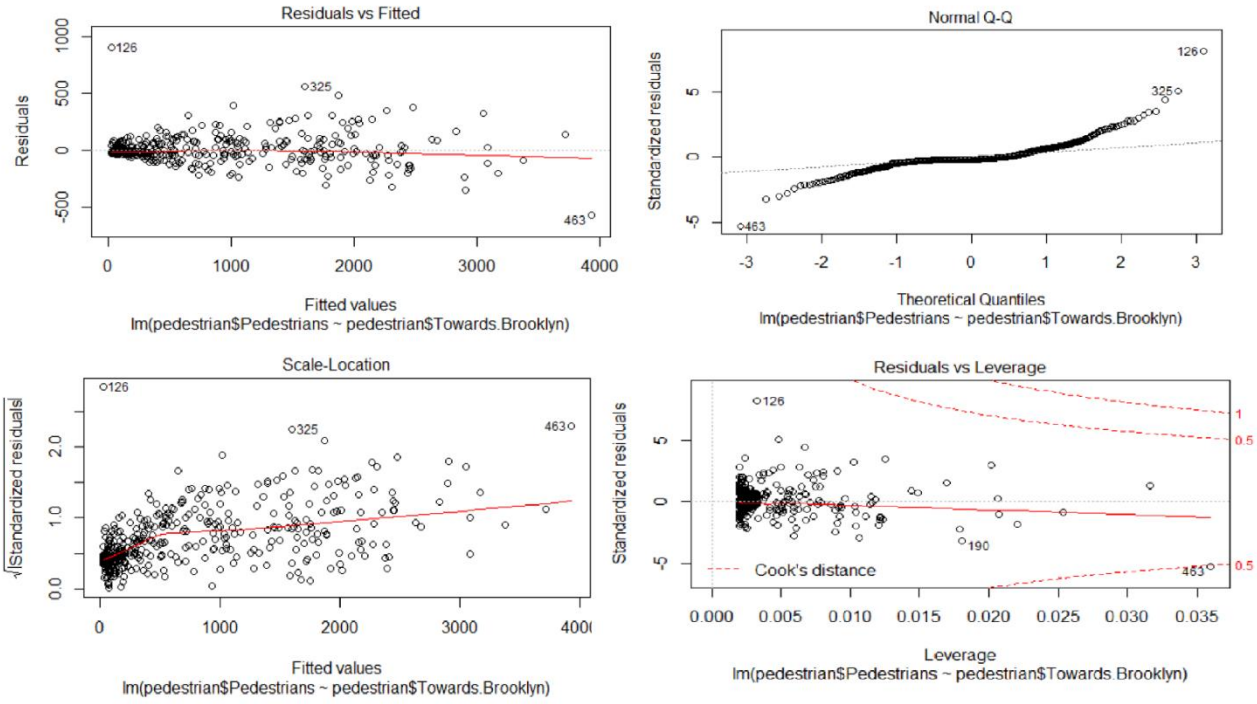
This plot helps us to find influential cases (i.e., subjects) if any. Not all outliers are influential in linear regression analysis (whatever outliers mean). Even though data have extreme values, they might not be influential to determine a regression line. That means, the results wouldn't be much different if we

either include or exclude them from analysis. They follow the trend in the majority of cases and they don't really matter; they are not influential. On the other hand, some cases could be very influential even if they look to be within a reasonable range of the values. They could be extreme cases against a regression line and can alter the results if we exclude them from analysis. Another way to put it is that they don't get along with the trend in the majority of the cases.

b) Scatter plot between y = total number of pedestrians and x_1 = pedestrians towards Brooklyn Bridge

We have fewer outliers here for pedestrians towards Brooklyn variable but there is no heteroscedasticity as we see a much flatter line and an evenly distributed residuals in the first plot residuals and fitted values.

```
model<-lm(pedestrian$Pedestrians~pedestrian$Towards.Brooklyn)
plot(model)
```



Definitions of those various important points as used above in our definition:

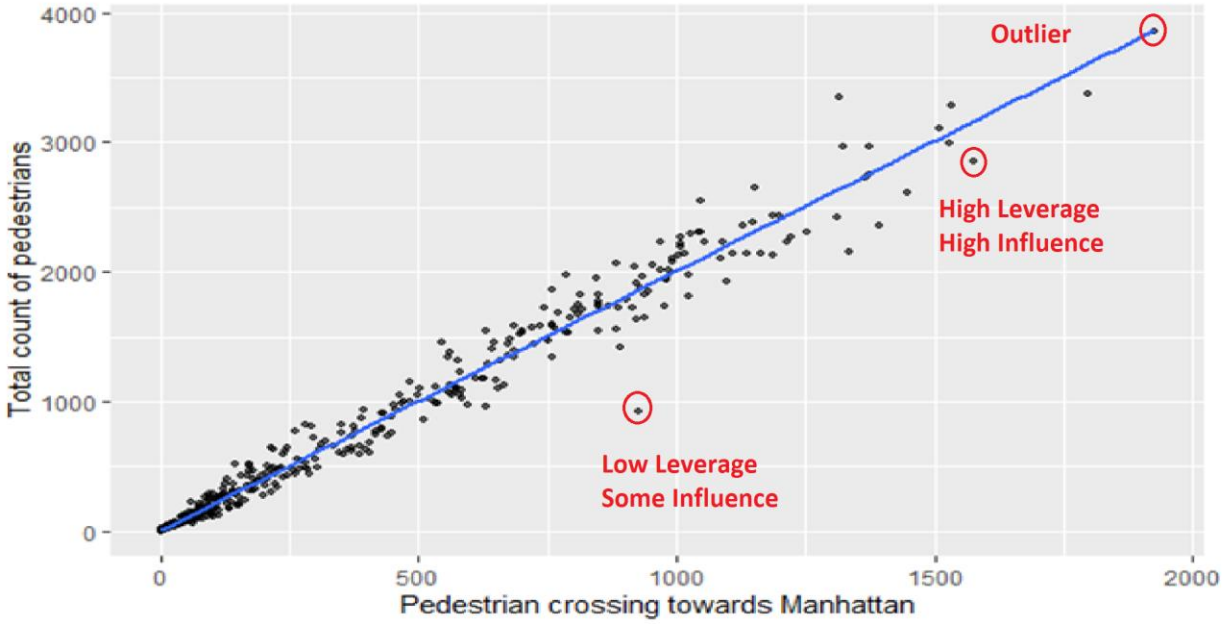
Leverage points are those observations, if any, made at extreme or outlying values of the independent variables such that the lack of neighboring observations means that the fitted regression model will pass close to that particular observation. In the scatter plot below we can see many of those case. I have tried to plot as many of them that I can identify.

Influential observations are those observations that have a relatively large effect on the regression model's predictions.

Although an **influential point** will typically have **high leverage**, a **high leverage point** is not necessarily **an influential point**.

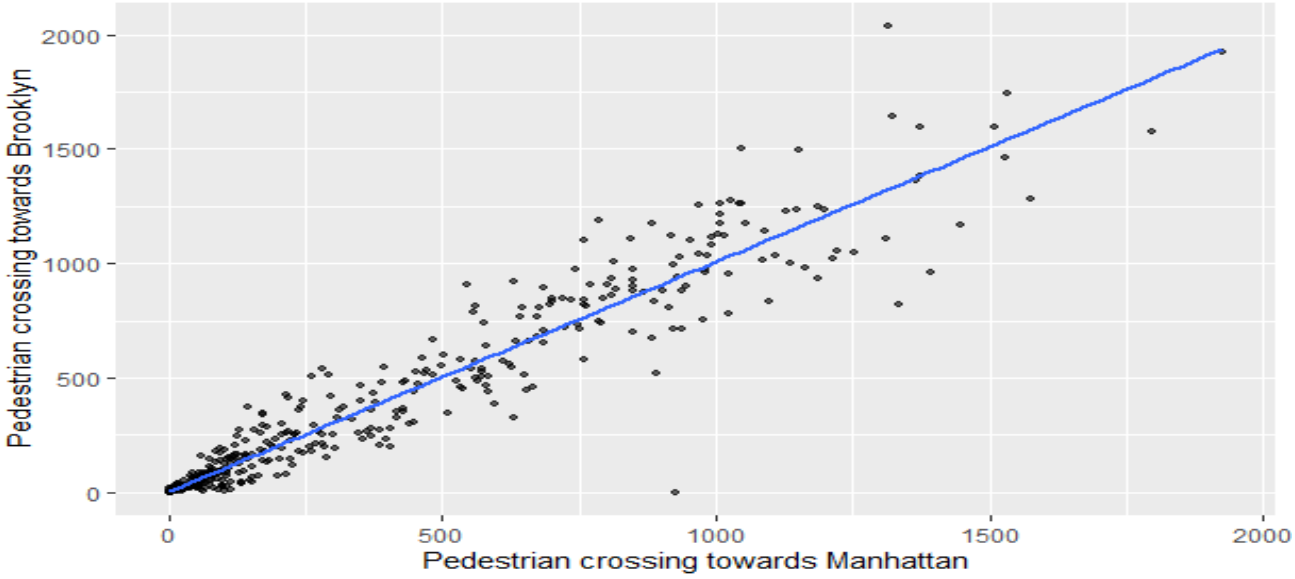
Now imagine an extra data point, an outlier some distance away from the main body of the data, but one which lies somewhere along that regression line. If the regression line were to be refitted, the coefficients would not change. Conversely, deleting the extra outlier would have zero influence on the coefficients. So, an outlier or leverage point would have zero influence if it were perfectly consistent with the rest of the data and the model that rest implies.

Identifying the outliers, leverage points and influential points:



2.3. Graphical Inspection for Collinearity

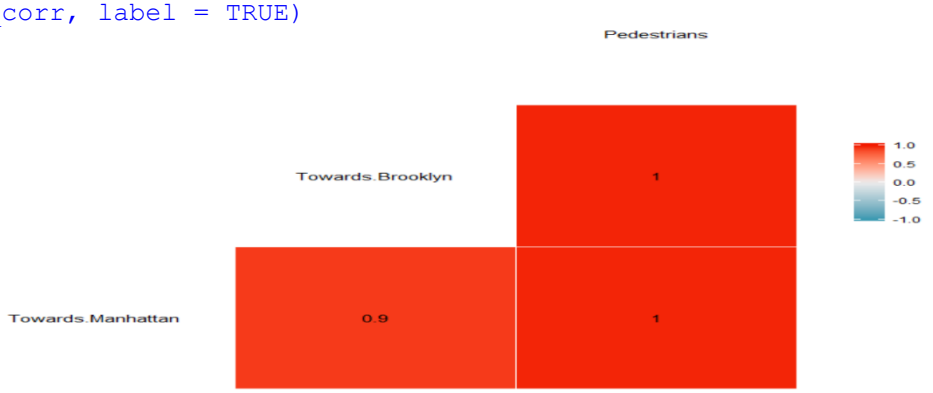
(a) Scatter plot for x_1 = pedestrians towards Manhattan and x_2 = pedestrians towards Brooklyn Bridge



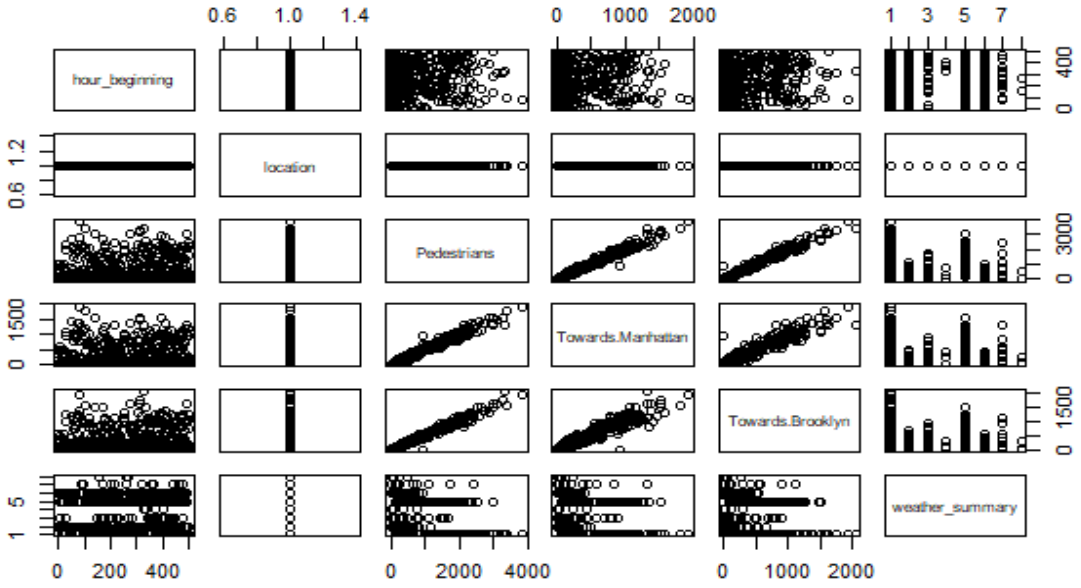
(b) Evidence of collinearity between x1 and x2:
Collinearity implies two variables are near perfect linear combinations of one another. Multicollinearity involves more than two variables. From the above graph we do observe quite a dependency between

Let's try and get the co-relation between these variables:

```
library(dplyr)
library(GGally)
ped_corr <- pedestrian %>% select(Towards.Manhattan,Towards.Brooklyn,
Pedestrians)
ggcorr(ped_corr, label = TRUE)
```




```
pairs(ped_pair)
```



We can see that the some of these variables in the pairs have collinearity in the dataset:

```
cor(ped_pair)
```

```
                  Pedestrians Towards.Manhattan Towards.Brooklyn
Pedestrians         1.0000000         0.9894647        0.9903916
Towards.Manhattan   0.9894647         1.0000000        0.9599366
Towards.Brooklyn    0.9903916         0.9599366        1.0000000
```

```
eigen(cor(ped_pair))$values
[1] 2.959930e+00 4.007046e-02 4.020611e-16
```

Therefore, to see high co-relation, we look into eigen values for the same and we do see some collinearity.

2.4. The Linear Regression Model

(a) The meaning of Y_x term in the model;

The regression equation is written as $Y_x = b_0 + b_1 X + \varepsilon$. Y_x is the value of the dependent variable (Y_x), which can be written in $Y_{x\ bar.}$ b_0 which is a constant, equals the value of $Y_{x\ bar}$ when the value of X=0. b_1 is the coefficient of X, which is the slope of the regression line. That means how much $Y_{x\ bar}$ changes for each one-unit change in X. X is the value of the independent variable (X), what is predicting or explaining the value of $Y_{x\ bar}$. e is the error term, which is the error in predicting the value of $Y_{x\ bar}$, given the value of X.

From the data set I have plotted, $Y_{x\ bar}$ is the total pedestrians and
b_0 =18.482 and b_1 = 1.970.

Therefore, for the above dataset if we use X=10:
$Y_{x\ bar}$ = 18.482+ 1.970 * 10 = 38.182

(b) Understanding the terms on the right-hand side of $E(Y_x)$;

The probability distribution of that random variable will be Normal, because both b_0 and b_1 are linear combinations of the Y's, which are themselves (assumed) to be Normally distributed conditional on knowing their respective x-values.

The expected value of $Y_{x\ bar}$ will be μ_{Yx} , or, $E(Y_{x\ bar}) = E(Yx) = \beta_0 + \beta_1 x$. This means that y-bar is unbiased estimator of μ_{Yx}, in much the same way as y-bar is an unbiased estimator of μ_y .

By substituting the values:

$E(Y_{x\ bar}) = E(b_0 + b_1 x)$
$= E(b_0) + E(b_1) * x$
$= \beta_0 + \beta_1 x$
$= E(Y_x)$
$= E(18.482 + 1.970 * x)$

(c) Understanding terms on the right-hand side of $V(Y_x)$.

As $V(\hat{Y}_x) = V[b_0 + b_1 x]$,

$= V[(\overline{Y} - b_1 \overline{x}) + b_1 x]$

$= V[(\overline{Y} + (b_1 x - b_1 \overline{x})]$

$= V[(\overline{Y} + b_1 (x - \overline{x})]$

$= V[(\overline{Y} + b_1 (x - \overline{x})]$

$= V(\overline{Y}) + V[b_1 (x - \overline{x})]$

$= \sigma^2_Y / n + (x - \overline{x})^2 V(b_1)$

$= \sigma^2_Y / n + (x - \overline{x})^2 \sigma^2_Y / SSx$

$$= \sigma_{Y}^{2}\left[\frac{1}{n}+\frac{(x-\bar{x})^{2}}{SSx}\right]$$

It can be shown that substituting MSE for the population variance results in an unbiased estimator. Thus, we have the sample standard error of $\hat{Y}_x$, defined as $s\sqrt{\frac{1}{n}+\frac{(x-\bar{x})^{2}}{SSx}}$, where s=√MSE. $V(\hat{Y}_x)$ is an unbiased estimator of $V(Y_x)$.

2.5. R output for the fitted model:

```
model<- lm(pedestrian$Pedestrians~pedestrian$Towards.Manhattan, data=pedestrian)
model

Call:
lm(formula = pedestrian$Pedestrians ~ pedestrian$Towards.Manhattan,
    data = pedestrian)

Coefficients:
(Intercept)  pedestrian$Towards.Manhattan
3.475                         2.005

P_conf <- predict(model, pedestrian, interval = "confidence")
P_conf %>% head()
     fit         lwr         upr
1 129.78626   117.54426   142.02827
2  45.57874    32.70923    58.44825
3 396.44341   385.69663   407.19019
4  51.59357    38.77090    64.41623
5  95.70227    83.21382   108.19071
6 189.93449   178.10014   201.76885

P_pred <- predict(model, pedestrian, interval = "prediction")
P_pred %>% head()
     fit           lwr        upr
1 129.78626    -99.33882   358.9113
2  45.57874   -183.58073   274.7382
3 396.44341    167.39335   625.4935
4  51.59357   -177.56328   280.7504
5  95.70227   -133.43612   324.8407
6 189.93449    -39.16917   419.0382
```

2.6. Analysis of Output

(i) Understanding what we are trying to achieve through t-test:

This t-statistic is computed by dividing the value of b_1 by the sample standard error of b_1,t -stat = b_1 / s_{b_1}. The goal of this t-test is to see if we have convincing evidence that β_1 is not zero. In the hypothesis testing framework, then, we would say that our null hypothesis is β_1=0 and our alternate hypothesis is $\beta_1 \neq 0$. In symbols, we write this as,

H_0: $\beta_1 = 0$

H_1: $\beta_1 \neq 0$

Test statistic: t-stat = $\dfrac{b_1 - 0}{s/\sqrt{SSx}}$ = | 8.192 |

Rejection region: |t-stat| > t-critical value, α=0.05
t-critical value with 2.000 d.f. = 728

Conclusion: Null hypothesis is rejected as |t-stat| = | 8.192 | is greater than t-critical value of 2.000

(ii) The results of the t-test as we see from R:

```
t.test(x, y, va.equal = TRUE)

Welch Two Sample t-test

data:  x and y
t = -8.1921, df = 728.46, p-value = 1.152e-15

alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -406.9671 -249.6169

sample estimates:
mean of x   mean of y
323.220      651.512

t.test(z, y, va.equal = TRUE)

Welch Two Sample t-test

data:  z and y
t = -7.9908, df = 747.68, p-value = 5.076e-15

alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-402.6276 -243.8124

sample estimates:
mean of x   mean of y
328.292      651.512
```

(b) The F-test

(i) In multiple regression, the F-test finally wakes up and takes a noble character of its own. Its null hypothesis will state in this application,

H_o: $\beta_1 = \beta_2 = 0$;
H_a: either $\beta_1 \neq 0$ or $\beta_2 \neq 0$ or both.

(ii) Here, the null hypothesis is rejected because the p-value was less than 0.05. The conclusion of the test, then, is the rather weak statement that either one or the other or both of the true partial slope coefficients is not zero.

(c) <u>The</u> $\hat{y}$ -<u>equation</u>

(i) $\hat{y} = b_0 + b_1 x$

$= 18.482 + 1.970 * x$

(ii) Relationship of $\hat{y}$ to that of $E(Y_x)$.

The particular value of $\hat{y}$ that would be computed depends on the sample from which b_0 and b_1 were computed. That sample consists of n pairs of (x, y) values. If we took a new sample of n pairs of (x,y) values, we would get a different value of b_0 and a different value of b_1. Therefore, we would get a different value of $\hat{y}$ for the same value of x. To be specific, suppose we took a sample of 50 pedestrians from NYC, regressed the pedestrians towards Mahattan, and then computed $\hat{y}$ for x=30. Imagine we were to continue repeating that experiment again and again until we have examined every possible sample of size 50. For each and every one of those zillions of possible samples, we compute the sample slope and sample intercept, along with the $\hat{y}$ values for x=30. The list of all of those $\hat{y}$ values for x=30, one for each possible sample, results in daughter population for Y hat by given x=30. The probability distribution of that random variable will be Normal, because both b_0 and b_1 are linear combinations of the Y's, which are themselves (assumed) to be Normally distributed conditional on knowing their respective x-values.

The expected value of $\hat{Y}_x$ will be μ_{Y_x}, or, $E(\hat{Y}_x) = E(Yx) = \beta_0 + \beta_1 x$. This means that $\hat{y}$ is unbiased estimator of μ_{Y_x}

2.7. <u>The Partial Regression Coefficient</u>

(a) The partial regression coefficient is also called regression coefficient. It is used in the context of multiple linear regression analysis and gives the amount by which the dependent variable (DV) increases when one independent variable (IV) is increased by one unit and all the other independent variables are held constant.

This coefficient is called partial because its value depends, in general, upon the other independent variables. Specifically, the value of the partial coefficient for one independent variable will vary, in general, depending upon the other independent variables included in the regression equation.

(b) R output for partial regression coefficient from R:

As you can see that the multiple regression of y on x_1 and y on both x_1 and x_2 is different from the regression summary below.

```
summary(lm(Total_ped_NYC ~ Ped_Mahatttan))

Residuals
Min            1Q  Median      3Q     Max
-933.05      -24.69   -3.49   25.23  714.04

Coefficients:
            Estimate   Std. Error     t value        Pr(>|t|)
(Intercept)  3.47498    6.72126       0.517          0.605
x            2.00494    0.01315 1     52.519         <2e-16 ***
```

```
Residual standard error: 116.5 on 498 degrees of freedom
Multiple R-squared:  0.979,    Adjusted R-squared:  0.979
F-statistic: 2.326e+04 on 1 and 498 DF,  p-value: < 2.2e-16
```

```
summary(lm(Total_ped_NYC ~ Ped_Brooklyn + Ped_Brooklyn))

Residuals:
 Min          1Q      Median          3Q         Max
-1.680e-11 -8.800e-15  2.500e-14  4.790e-14  1.446e-11

Coefficients:
             Estimate   Std. Error       t value        Pr(>|t|)
(Intercept) 4.067e-14    5.988e-14     6.790e-01       0.497
x           1.000e+00    4.178e-16     2.393e+15       <2e-16 ***
z           1.000e+00    3.991e-16     2.505e+15       <2e-16 ***

Residual standard error: 1.037e-12 on 497 degrees of freedom
Multiple R-squared:      1,    Adjusted R-squared:      1
F-statistic: 1.497e+32 on 2 and 497 DF,  p-value: < 2.2e-16
```

(c)

```
reg_Ped_Man <- lm(pedestrian$Pedestrians~pedestrian$Towards.Manhattan, data = pe
destrian)
reg_Ped_Man_res <- residuals(reg_Ped_Man)
reg_man_brook <- lm(pedestrian$Towards.Brooklyn~pedestrian$Pedestrians, data = p
edestrian)
reg_man_brook_res <- residuals(reg_man_brook)
summary(lm(reg_Ped_Man_res~reg_man_brook_res))

Call:
lm(formula = reg_Ped_Man_res ~ reg_man_brook_res)

Residuals:
Min      1Q  Median      3Q     Max
-13.655 -12.880  -8.342   9.138  67.081

Coefficients:
                    Estimate   Std. Error     t value        Pr(>|t|)
(Intercept)       -6.483e-15    7.540e-01       0.0             1
reg_man_brook_res  2.005e+00    1.315e-02       152.5        <2e-16 ***

Residual standard error: 16.86 on 498 degrees of freedom
Multiple R-squared:  0.979,    Adjusted R-squared:  0.979
F-statistic: 2.326e+04 on 1 and 498 DF,  p-value: < 2.2e-16
```

2.8. R-square

(a) The R-squared formula is calculated by dividing the sum of the first errors by the sum of the second errors and subtracting the derivation from 1 which comes out to be 0.9790

(b) summary(lm(y~x))$r.squared

(c) Understanding the meaning of R-square

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model.

R-squared = Explained variation / Total variation

R-squared is always between 0 and 100%:

a) 0% indicates that the model explains none of the variability of the response data around its mean.
b) 100% indicates that the model explains all the variability of the response data around its mean.

In general, the higher the R-squared, the better the model fits your data.
In practice, R square is not likely to be 0 or 1 but somewhere between these limits. The closer it is to 1, the greater is said to be the degree of linear association between X and Y.

(d) Interpreting R-square:
From the notes we understood that, R^2 can be interpreted as the proportionate reduction of total variation associated with the use of the predictor variable X.

2.9. <u>Adjusted R-square</u>
(a) 0.9789983

(b) summary(lm(y~x))$adj.r.squared

(c) Understanding the meaning of adjusted R-square:
The adjusted R-squared compares the descriptive power of regression models (two or more variables), that include a diverse number of independent variables—known as a predictor. Every predictor or independent variable added to a model increases the R-squared value and never decreases it. So, a model that includes several predictors will return higher R2 values and may seem to be a better fit. However, this result is due to it including more terms.

The adjusted R-squared compensates for the addition of variables and only increases if the new predictor enhances the model above what would be obtained by probability. Conversely, it will decrease when a predictor improves the model less than what is predicted by chance.

The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model. The adjusted R-squared increases only if the new term improves the model more than would be expected by chance. It decreases when a predictor improves the model by less than expected by chance. The adjusted R-squared can be negative, but it's usually not. It is always lower than the R-squared.

Chapter 3: Matrix Methods

3.1. Simple Linear Regression in Matrix Terms

Solution: (a) and (b)

I took the values from the same dataset I used in my project, that is, pedestrian data from sensors to identify number of pedestrians (source: NYC open source data) who came on a particular day across different hours and how many of them were going towards Brooklyn Bridge and how many of them were coming towards Manhattan.

From the dataset, I selected x-variable as total pedestrians and y-variable as pedestrians towards Manhattan. I randomly selected 8 rows from 7000+ observations using rand function in excel.

X		Y
1	111	63
1	33	21
1	428	196
1	39	24
1	72	46
1	265	93
1	31	25
1	123	114

(c) Computing the b-vector, Hat matrix, and the y-hat vector using R:

X		Y
1	111	63
1	33	21
1	428	196
1	39	24
1	72	46
1	265	93
1	31	25
1	123	114

X'X	
8	1102
1102	289614

(X'X)^-1	
0.2626865	-0.0009995
-0.0009995	0.0000073

X'							
1	1	1	1	1	1	1	1
111	33	428	39	72	265	31	123

H							
0.13019225	0.145332279	0.068662	0.144168	0.137762	0.1003	0.14572	0.127863
0.14533228	0.204618923	-0.09561	0.200058	0.174976	0.028279	0.206139	0.136211
-0.06866163	-0.095614726	0.736298	-0.08298	-0.01348	0.393002	-0.09983	0.093935
0.14416766	0.200058412	-0.08298	0.195759	0.172113	0.033819	0.201492	0.135569
0.13776227	0.174975601	-0.01348	0.172113	0.156369	0.06429	0.17593	0.132037
0.10030041	0.02827916	0.393002	0.033819	0.06429	0.242496	0.026432	0.111381
0.14572048	0.206139094	-0.09983	0.201492	0.17593	0.026432	0.207688	0.136425
0.12786302	0.136211257	0.093935	0.135569	0.132037	0.111381	0.136425	0.126579

Summarized solution:

	b-hat vector	y-hat column	
X'y	(X'X)^(-1) * X'y	X(X'X)^(-1)*X'y	Hy
582	17.6818853	62.05619188	62.05619188
135264	0.399768528	30.87424672	30.87424672
		188.7828152	188.7828152
		33.27285788	33.27285788
		46.4652193	46.4652193
		123.6205452	123.6205452
		30.07470966	30.07470966
		66.85341422	66.85341422

Explanation:
We finalize are x and y variables, then we transpose x and multiply it with x' to get our new XX' matrix. Now, we do an inverse in the XX' matrix. Next, we calculate the H matrix using our original X variable values against the inverse of XX' matrix. Lastly, we compute a **(X'X)^(-1) * X'y** to get our b-hat vector and **X(X'X)^(-1)*X'y** to get out y-hat column.

(d) Let's repeat the process of matrix operations on R:

```
Y = pedestrian[,2]
Xmat = cbind(rep(1,8),pedestrian[,1])

Y
[1]  63  21 196  24  46  93  25 114
Xmat
     [,1] [,2]
[1,]    1  111
[2,]    1   33
[3,]    1  428
[4,]    1   39
```

```
[5,]    1   72
[6,]    1  265
[7,]    1   31
[8,]    1  123

b_hat <- solve(t(Xmat) %*% Xmat) %*% t(Xmat) %*% Y
b_hat
           [,1]
[1,] 17.6818853
[2,]  0.3997685
```

We do see the same value as we got in excel and hence we are sure about our process on Excel. Now, let's repeat it using regression function in R.

```
> x <- pedestrian$Total.Pedestrians
> z <- pedestrian$Towards.Manhattan
> summary(lm(z~x))

Residuals:
    Min      1Q  Median      3Q     Max
-30.621  -9.423  -2.770   2.512  47.147

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 17.68189   12.24158   1.444 0.198742
x            0.39977    0.06434   6.213 0.000802 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 23.88 on 6 degrees of freedom
Multiple R-squared:  0.8655,     Adjusted R-squared:  0.8431
F-statistic: 38.61 on 1 and 6 DF,  p-value: 0.0008022
```

Conclusion: *The above answer for b_hat is same as b-hat vector we calculated in Excel earlier.*

(e) The interaction variable has a significant impact on the independent variable. It does not introduce a linear relationship with y. Instead it is a second-degree regression.

3.2. Multiple Linear Regression in Matrix Terms

(a) I added a new regressor variable that is temperature, as denoted by the notation: X_2.

Total Pedestrians	Temperature	Towards Manhattan
X1	X2	Y
111	53	63
33	44	21
428	71	196
39	36	24
72	39	46
265	66	93
31	36	25
123	34	114

(b) R output with the code:

```
> Y = pedestrian[,3]
> Xmat = cbind(rep(1,8),pedestrian[,1], pedestrian[,2])

> Y
[1]  63  21 196  24  46  93  25 114
> Xmat
     [,1] [,2] [,3]
[1,]    1  111   53
[2,]    1   33   44
[3,]    1  428   71
[4,]    1   39   36
[5,]    1   72   39
[6,]    1  265   66
[7,]    1   31   36
[8,]    1  123   34

> b_hat <- solve(t(Xmat) %*% Xmat) %*% t(Xmat) %*% Y
> b_hat
            [,1]
[1,] 107.7450606
[2,]   0.6343655
[3,]  -2.5831961

> Xmat %*% b_hat
          [,1]
[1,]  41.25024
[2,]  15.01850
[3,] 195.84659
[4,]  39.49026
[5,]  52.67473
[6,] 105.36099
[7,]  34.41533
[8,]  97.94336

> HatMat <- Xmat %*% solve(t(Xmat) %*% Xmat) %*% t(Xmat)
> HatMat %*% Y
          [,1]
[1,]  41.25024
[2,]  15.01850
[3,] 195.84659
[4,]  39.49026
[5,]  52.67473
[6,] 105.36099
[7,]  34.41533
[8,]  97.94336
```

(c) Verify the b-vector values:

```
> b_hat <- solve(t(Xmat) %*% Xmat) %*% t(Xmat) %*% Y
> b_hat
            [,1]
[1,] 107.7450606
[2,]   0.6343655
[3,]  -2.5831961
```

Comparing b_hat value and coeffecients from R code using 'lm' function:

```
> x <- pedestrian$Total.Pedestrians
> y <- pedestrian$temperature
> z <- pedestrian$Towards.Manhattan
> summary(lm(z~x+y))

Residuals:
       1        2        3        4        5        6        7        8
 21.7498   5.9815   0.1534 -15.4903  -6.6747 -12.3610  -9.4153  16.0566

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 107.74506   32.43908   3.321  0.02097 *
x             0.63437    0.09249   6.859  0.00101 **
y            -2.58320    0.89990  -2.871  0.03497 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.08 on 5 degrees of freedom
Multiple R-squared:  0.9492,     Adjusted R-squared:  0.9289
F-statistic: 46.72 on 2 and 5 DF,  p-value: 0.0005815
```

The above answer is derived using regression summary function in R and the values match with the computation we did over matrices earlier and thus we can collude that the answers are correct, and we have validated it.

(d) Step 1: Read the file into R console.
Step 2: Define the Y variable. We choose the last column from the file we just read.

```
> Y = pedestrian[,3]
> Xmat = cbind(rep(1,8),pedestrian[,1], pedestrian[,2])

> Y
[1]  63  21 196  24  46  93  25 114
```

Step 3: We convert the data for X variables from the file into a matrix for operations

```
> Xmat
     [,1] [,2] [,3]
[1,]    1  111   53
[2,]    1   33   44
[3,]    1  428   71
[4,]    1   39   36
[5,]    1   72   39
[6,]    1  265   66
[7,]    1   31   36
[8,]    1  123   34
```

Step 4: Solving for the sample slope

```
> b_hat <- solve(t(Xmat) %*% Xmat) %*% t(Xmat) %*% Y
> b_hat
             [,1]
[1,] 107.7450606
[2,]   0.6343655
[3,]  -2.5831961
```

Step 5: We calculate y-hat vector by multiplying it with the X matrix and b_hat column. And then continuing the operation to calculate the HatMat parameters.

```
> Xmat %*% b_hat
          [,1]
[1,]  41.25024
[2,]  15.01850
[3,] 195.84659
[4,]  39.49026
[5,]  52.67473
[6,] 105.36099
[7,]  34.41533
[8,]  97.94336
```

Step 6: Next we compute the hat matrix, HatMat. Then I show that the matrix product of the hat matrix with the y-vector yields the yhat-vector

```
> HatMat <- Xmat %*% solve(t(Xmat) %*% Xmat) %*% t(Xmat)
> HatMat %*% Y
          [,1]
[1,]  41.25024
[2,]  15.01850
[3,] 195.84659
[4,]  39.49026
[5,]  52.67473
[6,] 105.36099
[7,]  34.41533
[8,]  97.94336
```

Chapter 4: Polynomial Regression

4.1. Simple Polynomial Regression

(a) Compute x^2 in R:

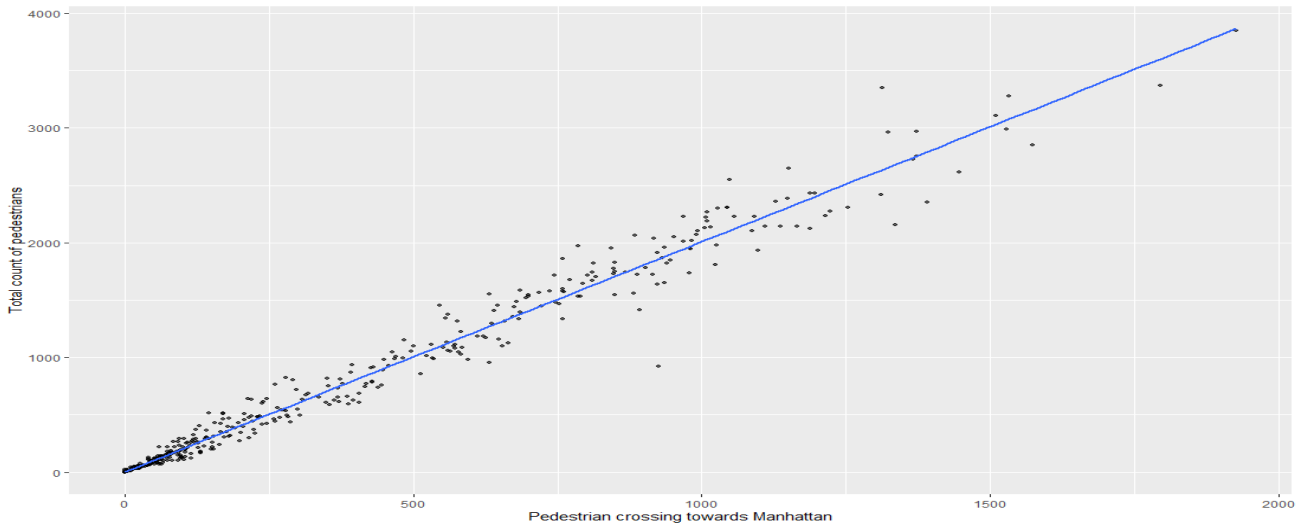
I am using the same dataset I have used earlier for the project. The pedestrian data on a regular day who are crossing the Brooklyn Bridge from Manhattan. We will plot the total pedestrians on a random day against the total pedestrians going towards Manhattan.

```
pedestrian <- pedestrian %>% mutate(square_x = TM * TM)
x_sq <- pedestrian$square_x
```

```
> pedestrian %>% select(2,3,4,16) %>% head(n=5)
   location            Pedestrians    Towards.Manhattan   square_x
1  Brooklyn Bridge         111                63             3969
2  Brooklyn Bridge          33                21              441
3  Brooklyn Bridge         428               196            38416
4  Brooklyn Bridge          39                24              576
5  Brooklyn Bridge          72                46             2116
```

(b) Simple polynomial regression in R:

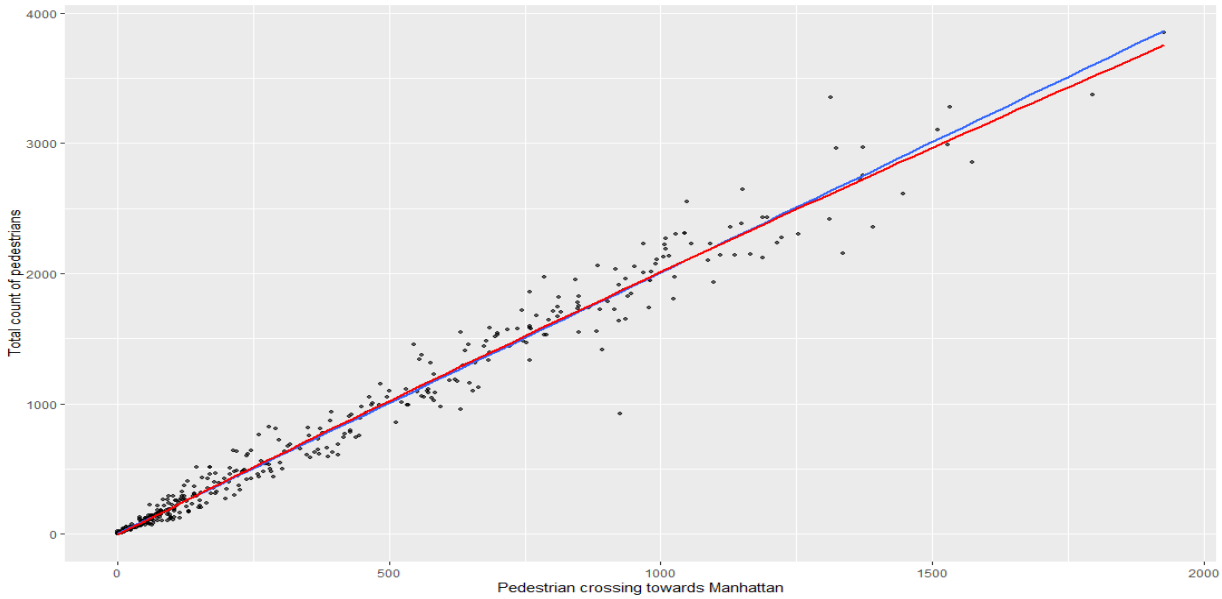
```
> # Creating scatter plot for the given data set
> library(ggplot2)
> ggplot(pedestrian, aes(x=x,y=y)) + geom_point(size=1, alpha=0.6) +
+   geom_smooth(method = 'lm', se = FALSE) + xlab("Pedestrian crossing towards M
anhattan") + ylab("Total count of pedestrians")
```



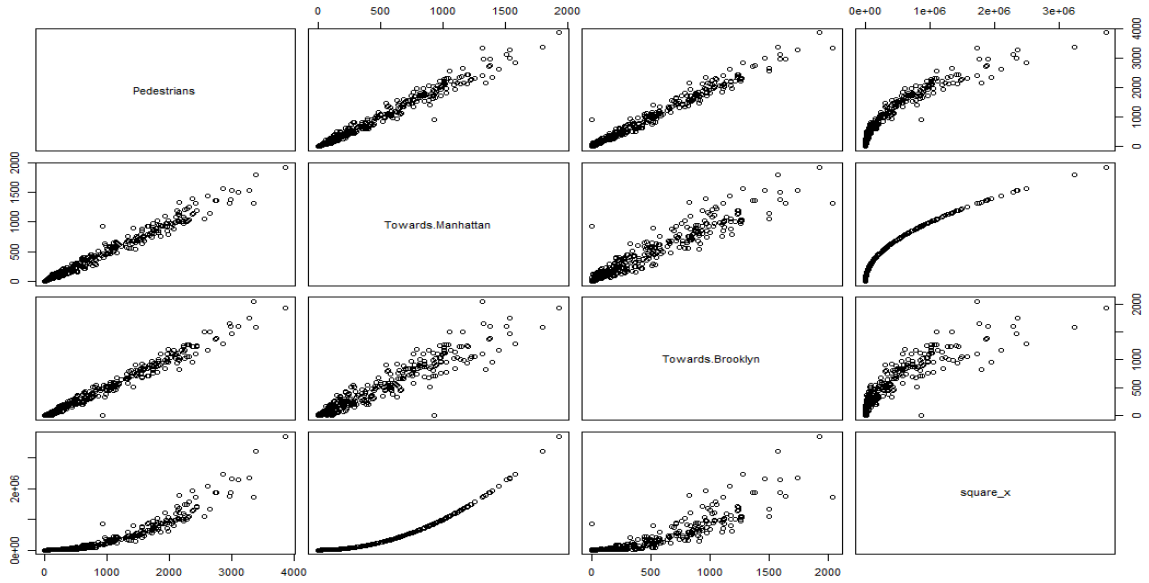

```
> model<-lm(pedestrian$Pedestrians~pedestrian$Towards.Manhattan)
> summary(model)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-933.05  -24.69   -3.49   25.23  714.04
```

```
Coefficients:
                              Estimate  Std. Error   t value   Pr(>|t|)
(Intercept)                    3.47498      6.72126   0.517    0.605
pedestrian$Towards.Manhattan   2.00494      0.01315  152.519   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 116.5 on 498 degrees of freedom
Multiple R-squared:  0.979,    Adjusted R-squared:  0.979
F-statistic: 2.326e+04 on 1 and 498 DF,  p-value: < 2.2e-16

> model_2 <- lm(pedestrian$Pedestrians~pedestrian$Towards.Manhattan + x_sq)
> summary(model_2)

Residuals:
    Min      1Q  Median      3Q     Max
-940.70  -28.98    3.06   26.16  735.91

Coefficients:
                               Estimate  Std. Error  t value  Pr(>|t|)
(Intercept)                  -4.738e+00   7.559e+00   -0.627    0.5311
pedestrian$Towards.Manhattan  2.088e+00   3.781e-02   55.223    <2e-16 ***
x_sq                         -7.103e-05   3.041e-05   -2.336    0.0199 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 115.9 on 497 degrees of freedom
Multiple R-squared:  0.9793,   Adjusted R-squared:  0.9792
F-statistic: 1.174e+04 on 2 and 497 DF,  p-value: < 2.2e-16

> ggplot(pedestrian, aes(x=x,y=y)) + geom_point(size=1, alpha=0.6) +
+   geom_smooth(method = 'lm', se = FALSE) +
+   stat_smooth(method="lm", se=TRUE, fill=NA, formula=y ~ poly(x, 2, raw=TRUE),
colour="red") +
+   xlab("Pedestrian crossing towards Manhattan") + ylab("Total count of pedestr
ians")
```

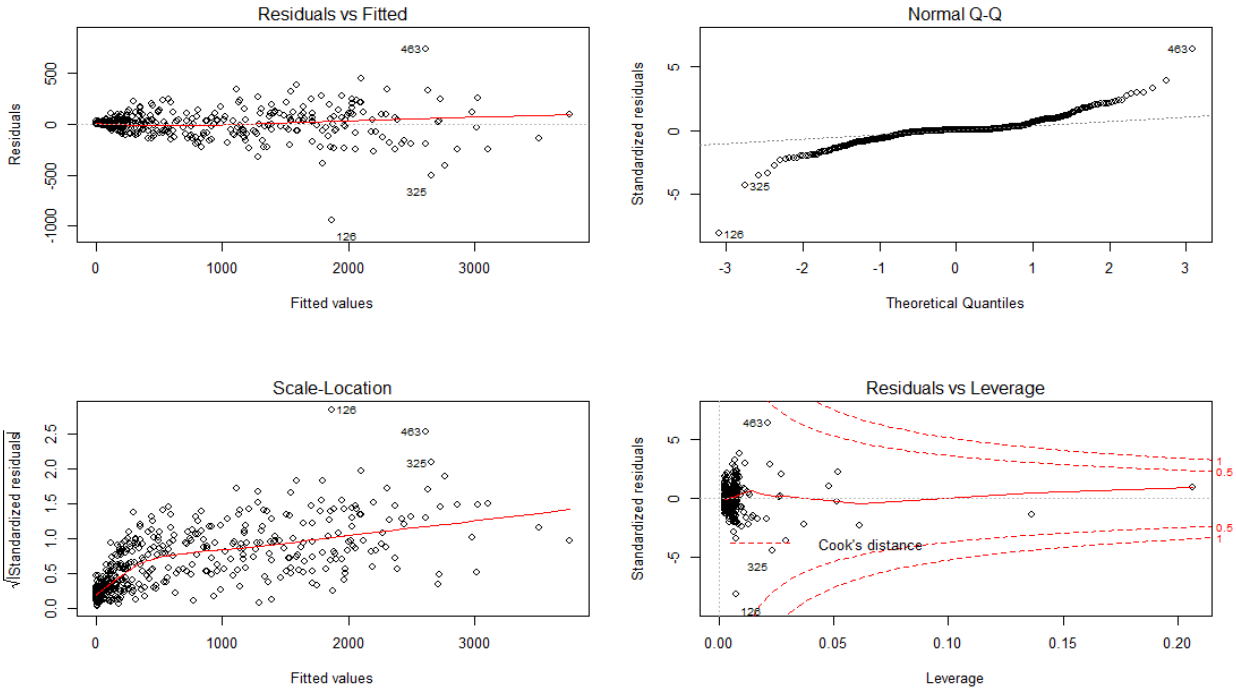


```
pairs(pedestrian %>% select(c(3,4,5,13)))
```



(c) The highest leverage point is the **observation 463** as seen from the below plots for Standardized Residuals v/s (Normal Q-Q and Leverage) and the leverage value is 0.22

```
> par(mfrow=c(2,2))
> plot(model)
```



(d) Explanation on the computation of the leverage points:
The location of points in x -space affects the model properties like parameter estimates, standard errors, predicted values, summary statistics etc.

The hat matrix ***H = X(X'X)$^{-1}$X'***
plays an important role is identifying the influential observations and leverage points. Since,

$$V(\hat{y}) = \sigma^2 H$$
$$V(e) = \sigma^2 (I - H),$$

(y-hat is fitted value and e is residual) the elements h_{ii} *of H matrix* may be interpreted as the amount of leverage excreted by the i^{th} observation y_i on the i^{th} fitted value y-hat$_i$.

The i^{th} diagonal element of H is given by

$$h_{ii} = x_i'(X'X)^{-1}x_i$$

where x_i' is the i^{th} row of X -matrix. The hat matrix diagonal is a standardized measure of the distance of i^{th} observation from the center (or centroid) of the x − space. Thus, large hat diagonals reveal observations that are potentially influential because they are remote in x -space from the rest of the sample.

So, we look into this diagonal variable and identify from the H – matrix, the value of the leverage points.

```
> Y = pedestrian[,2]
> Xmat = cbind(rep(1,8),pedestrian[,1])
> Xmat
> max(Xmat)
[1] 463

> > b_hat <- solve(t(Xmat) %*% Xmat) %*% t(Xmat) %*% Y
> b_hat

> max(diag(b_hat))
[1] 0.22
```

(e) (i) Hypothesis statements:

T-test is a univariate hypothesis test, that is applied when standard deviation is not known, and the sample size is small. It compares the means of two populations.

F-test is statistical test, that determines the equality of the variances of the two normal populations. It compares two population variances.

H_0: The null hypothesis: It is a statement about the population that either is believed to be true or is used to put forth an argument unless it can be shown to be incorrect beyond a reasonable doubt.

Ha: The alternative hypothesis: It is a claim about the population that is contradictory to H0 and what we conclude when we reject H_0. Since the null and alternative hypotheses are contradictory, you must examine evidence to decide if you have enough evidence to reject the null hypothesis or not. The evidence is in the form of sample data.

Evaluation of F-value using R is demonstrated below:

```
> anova(model, model_2)
Analysis of Variance Table

Model 1: pedestrian$Pedestrians ~ pedestrian$Towards.Manhattan
Model 2: pedestrian$Pedestrians ~ pedestrian$Towards.Manhattan + x_sq

  Res.Df     RSS     Df    Sum of Sq      F     Pr(>F)
1    498   6753414
2    497   6680063    1     73350       5.4573 0.01988 *

---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

(ii) Conditions for the null hypothesis to be rejected, for α=5%:

The significance level should be specified before any statistical calculations are performed. Then, when the probability (p) is computed from a significance test, it is compared with the significance level. The null hypothesis is rejected if p is at or below the significance level; it is not rejected if p is above the significance level. The degree to which p ends up being above or below the significance level does not matter. The null hypothesis either is or is not rejected at the previously stated significance level. Thus, if an experiment originally stated that he or she was using the 0.05 significance level and p was subsequently calculated to be 0.042, then the person would reject the null hypothesis at the 0.05 level or α=5%.

(iii) Conclusion of the test:

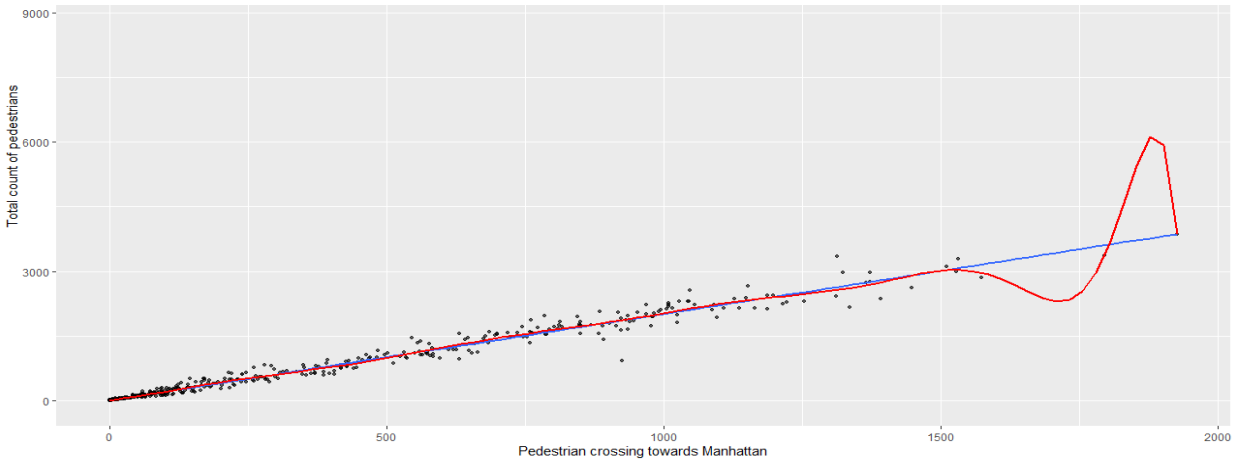
The conclusion is that the null hypothesis is rejected as the p-value for our predictor and regressor variable as shown below is less than 0.05 or α < 5%.

```
> mydata <- pedestrian %>% tbl_df()
> (chi2 <- chisq.test(table(mydata$Towards.Manhattan, mydata$Pedestrians), p = r
ep(1/12, 12)))

        Pearson's Chi-squared test

data:  table(mydata$Towards.Manhattan, mydata$Pedestrians)
X-squared = 138360, df = 113486, p-value < 2.2e-16
```

By looking at the p-Value: If the p-Value is less than 0.05, we fail to reject the null hypothesis that the x and y are independent. So for the example output above, (p-Value=2.2e-16), we reject the null hypothesis and conclude that x and y are not independent.

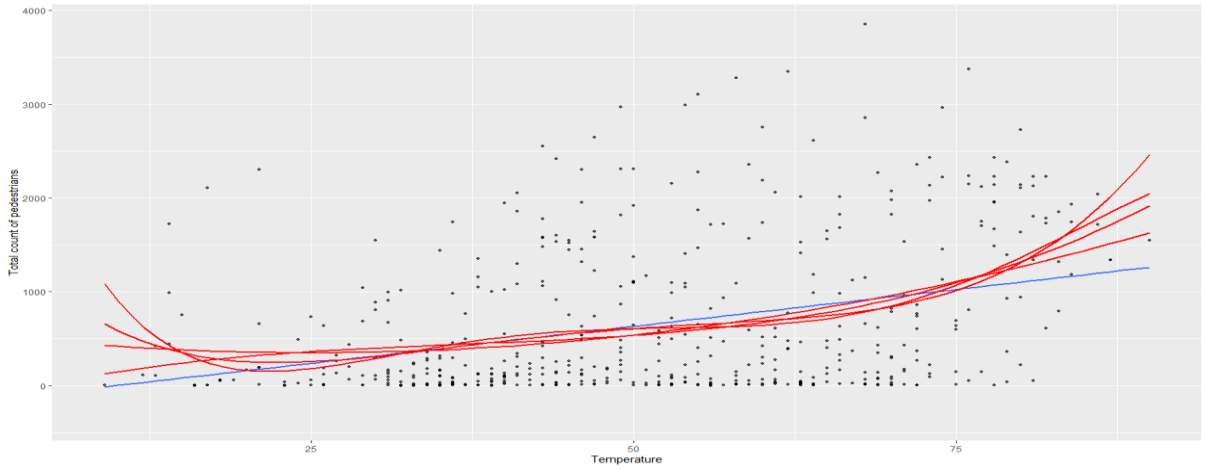
(f) From the same dataset I changed the x-variable to temperature of the day as the previously selected x-variable was working fine for x-line only and with any addition to degree of the x exponent was making no difference, which means we had a perfectly fitting line through x.

```
> ggplot(pedestrian, aes(x=x,y=y)) + geom_point(size=1, alpha=0.6) +
+   geom_smooth(method = 'lm', se = FALSE) +
+   stat_smooth(method="lm", se=TRUE, fill=NA, formula=y ~ poly(x, 14),colour="r
ed") +
+   xlab("Pedestrian crossing towards Manhattan") + ylab("Total count of pedestr
ians")
```



So, do make a more visually identifiable fun plot for polynomial, I chose a different variable as seen below and it works beautifully.

```
> ggplot(pedestrian, aes(x=x,y=y)) + geom_point(size=1, alpha=0.6) +
+   geom_smooth(method = 'lm', se = FALSE) +
+   stat_smooth(method="lm", se=TRUE, fill=NA, formula=y ~ poly(x, c(2,3,4,5),
+     xlab("Temperature") + ylab("Total count of pedestrians")
```



(g) If we look into the original x fitted line for total pedestrians walking towards Manhattan, we see that there is no need for a new x-value (for different exponents), as the original x-fitted line truly represents the slope of the dataset and it gives you the right standard error and adjusted R square values.

However, for temperature as x-value with no change in y value, we see a difference and it appears that x^3 gives a better representation of the curve as it reflects the change in temperature influencing the total pedestrians for the day and is not directly co-related.\

4.2. Multiple Regression with a Dummy Variable and an Interaction Term

(a) Defining a dummy variable:
I used the dataset I have used for que 1 above and for my two variables x1 and x2, I looked at the mean of the variables to see the number of pedestrians we have on average on a regular day. Using the means value, I define the dummy variables above mean as 1 and below mean as 0.

```
mean(pedestrian$Towards.Manhattan)
[1] 323.22
mean(pedestrian$Towards.Brooklyn)
[1] 328.292

pedestrian <- mutate(pedestrian,if (Towards.Manhattan > 323){x1_dummy = 1} else
{x1_dummy = 0})

pedestrian <- mutate(pedestrian,if (Towards.Brooklyn > 323){x2_dummy = 1} else {
x2_dummy = 0})
```

(b) Computing an interaction term for two of your x-variables:
In a regression equation, an interaction effect is represented as the product of two or more independent variables. For example, here is a typical regression equation without an interaction:

$$\hat{y} = b_0 + b_1X_1 + b_2X_2$$

where ŷ is the predicted value of a dependent variable, X1 and X2 are independent variables, and b0, b1, and b2 are regression coefficients. And here is the same regression equation with an interaction:

$$\hat{y} = b_0 + b_1X_1 + b_2X_2 + b_3X_1X_2$$

For our, dataset the interaction term will be defined as:

$\hat{y} = -8.99 + 2.00^* X_1 + 0.268 * X_2$

Here in the given equation, I have multiplied the dummy variable with our original x-value and following is the result:

```
> pedestrian %>% select(3,4,13,16,17) %>% head(n=5)

  Pedestrians Towards.Manhattan x1_dummy square_x interaction_term
1         111                63        0     3969                0
2          33                21        0      441                0
3         428               366        1    38416              366
4          39                24        0      576                0
5          72               416        1     2116              416
```

(c) Following code will represent it for all the variables and with their interaction term and dummy variables:

This code is of interest to me because it tells me the relation between my predictor variables, regressor variables and their respective dummy variables.

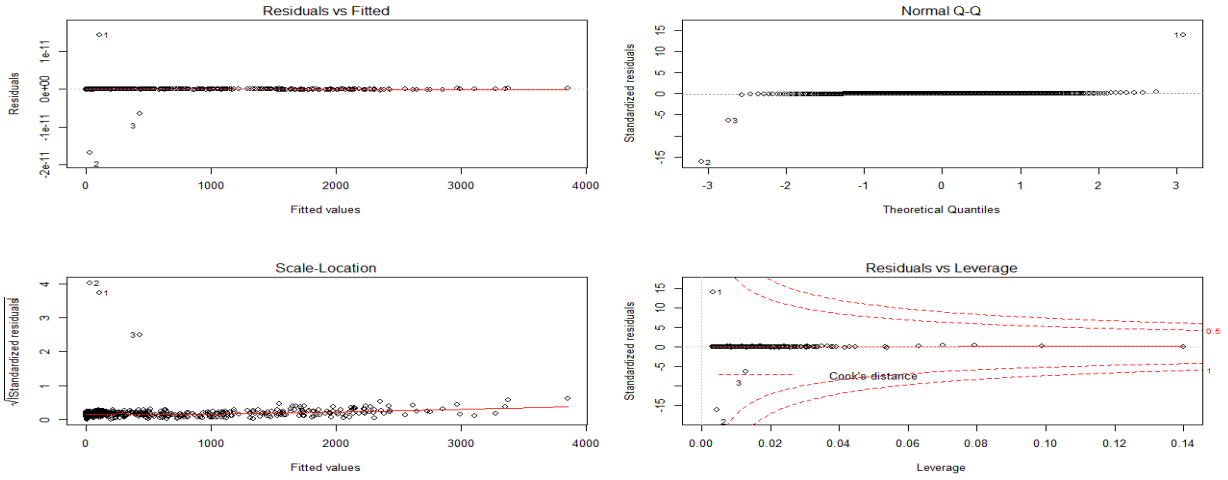
```
> x = pedestrian$Towards.Manhattan
> y = pedestrian$Pedestrians
> z = pedestrian$Towards.Brooklyn
```

```
> model_dummy <- lm(pedestrian$Pedestrians ~ pedestrian$Towards.Manhattan + pede
strian$Towards.Brooklyn + pedestrian$temperature + x*pedestrian$x1_dummy)
> summary(model_dummy)

Residuals:
       Min         1Q     Median         3Q        Max
-1.679e-11 -2.200e-15  1.930e-14  4.320e-14  1.447e-11

Coefficients: (1 not defined because of singularities)
                              Estimate  Std. Error      t value  Pr(>|t|)
(Intercept)                  0.000e+00   1.534e-13  0.000e+00     1.000
pedestrian$Towards.Manhattan  1.000e+00   8.169e-16   1.224e+15    <2e-16 ***
pedestrian$Towards.Brooklyn   1.000e+00   4.011e-16   2.493e+15    <2e-16 ***
pedestrian$temperature       -2.501e-16   2.840e-15  -8.800e-02     0.930
x                                    NA          NA          NA       NA
pedestrian$x1_dummy           1.047e-13   2.271e-13   4.610e-01     0.645
x:pedestrian$x1_dummy        -2.602e-16   7.344e-16  -3.540e-01     0.723
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.04e-12 on 494 degrees of freedom
Multiple R-squared:      1,    Adjusted R-squared:      1
F-statistic: 5.957e+31 on 5 and 494 DF,  p-value: < 2.2e-16
```

The p-value is way higher than what I expected and thus we can see that the null hypothesis can be neglected here.

I have also plotted the model summary with the interaction terms and the dummy variable using R:

```
> par(mfrow=c(2,2))
> plot(model_dummy)
```



(d) Understanding effect of a dummy variable on the fitted model:
It gives two intercepts for every y-value based on the chosen parameter and the category we are analyzing. That is for the x1_dummy here for values 1 and 0, the y-intercept will differ.

A Dummy variable or Indicator Variable is an artificial variable created to represent an attribute with two or more distinct categories/levels. It helps the fitted model to correctly analyze the attribute

variables. It helps us transform a continuous variable from a categorical variable and then we can use this new term to classify our attributes for the regression analysis.

Dummy variables assign the numbers ‘0’ and ‘1’ to indicate membership in any mutually exclusive and exhaustive category.

1. The number of dummy variables necessary to represent a single attribute variable is equal to the number of levels (categories) in that variable minus one.

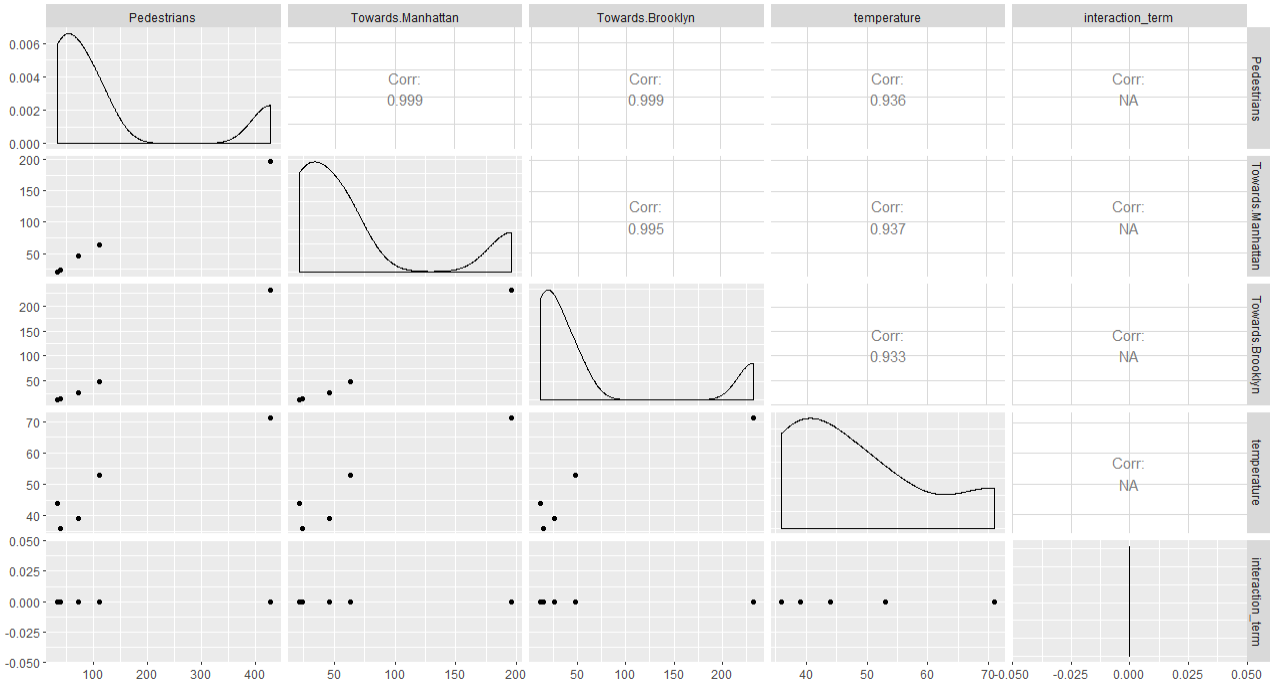
2. For a given attribute variable, none of the dummy variables constructed can be redundant. That is, one dummy variable can not be a constant multiple or a simple linear relation of another.

3. The interaction of two attribute variables is represented by a third dummy variable which is simply the product of the two individual dummy variables.

(e) Impact of an interaction variable:
The interaction variable here helps us identify the affect of categorical data on our predictor value. So, if we include interaction term with a dummy variable, the interaction term is equal to zero.

Now, we look into the data we know that the interaction variable is throwing a liner relationship with y. It doesn’t introduce a new liner relationship with y as seen from the co-relation chart below.

```
> library(GGally)
> pedestrian <- pedestrian %>% select(Pedestrians,Towards.Manhattan,Towards.Broo
klyn,temperature,interaction_term)

> ggpairs(pedestrian)
```



Chapter 5: Model Selection

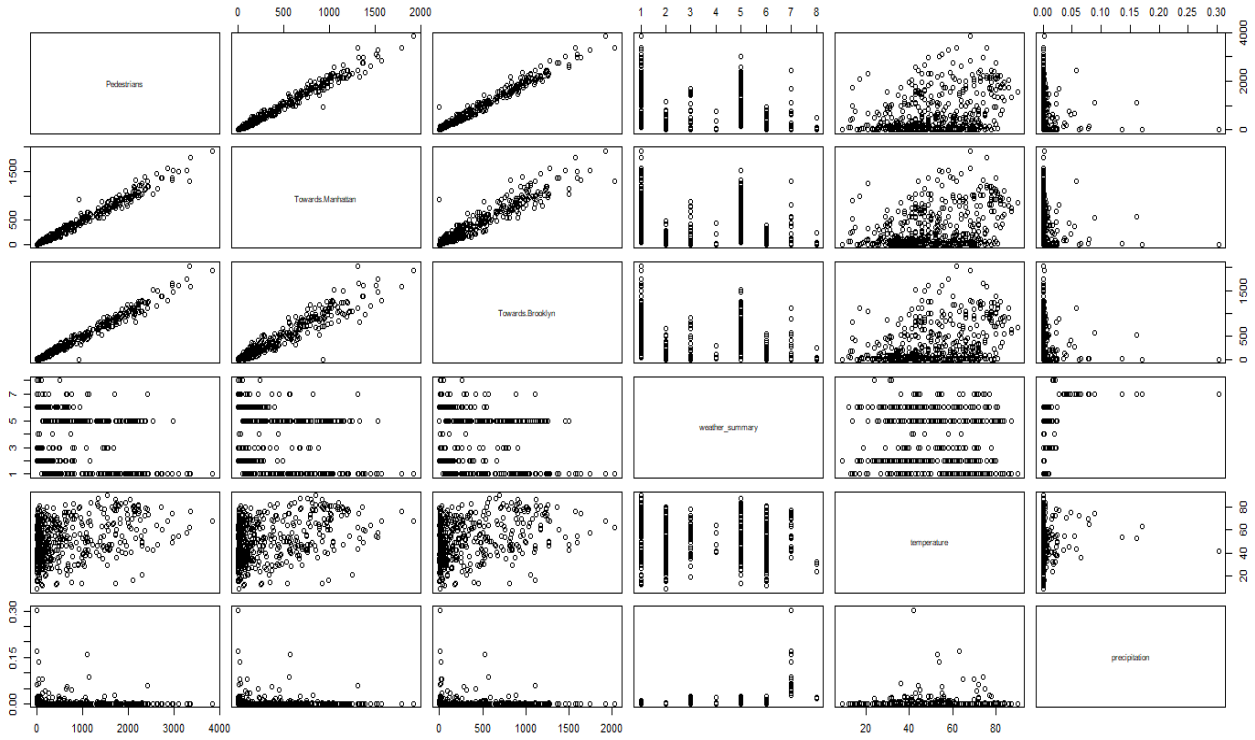
5.1. Best Subsets Model Selection

(a) Matrix scatterplot:

I continue to use the full data set that I have used for my project and other assignments and plot the matrix scatterplot using R.

Code:

```
pairs(pedestrian %>% select(c(3,4,5,6,7,8)))
```



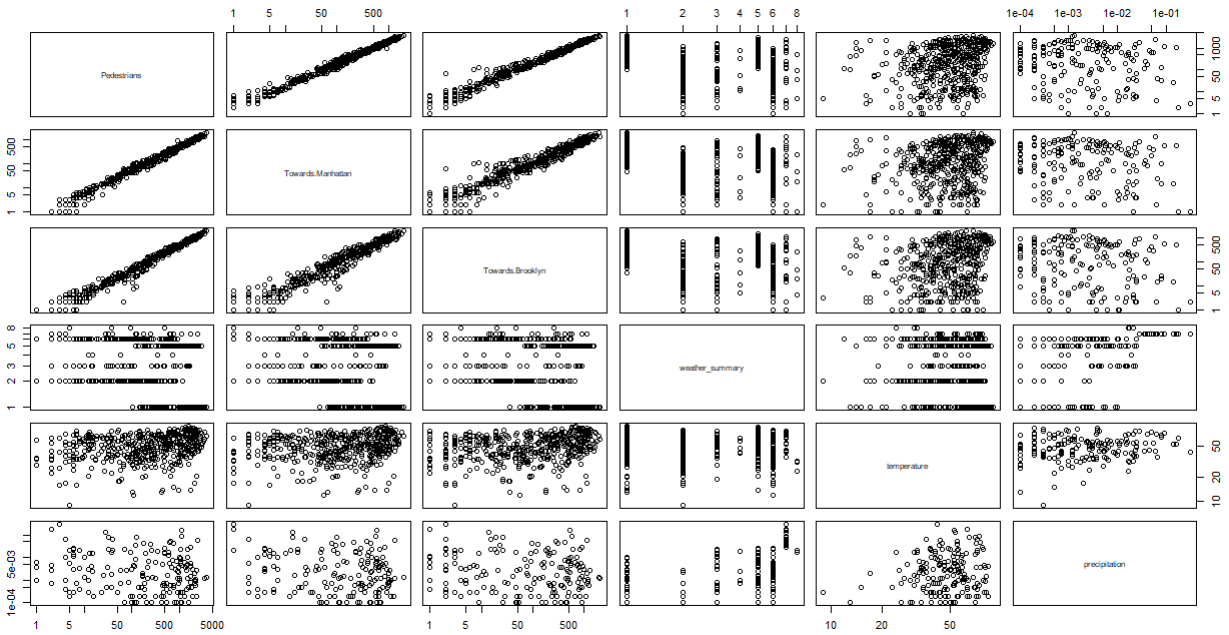
(b) Problems identified in the above matrix plot and the transformation one can do:

The only transformation I did was to clean the data to take a sample of 500 rows from an original set of 7000 observations. This was done to have a better understanding of the data and clean the scatter plot a bit to help easy interpretation.

I see that for temperature and precipitation, there are lot of issues with the underlying data as the points are mostly concentrated around x-axis or y-axis. I would like to plot the scatter plot using log of the regressor variables to see how it changes the model.

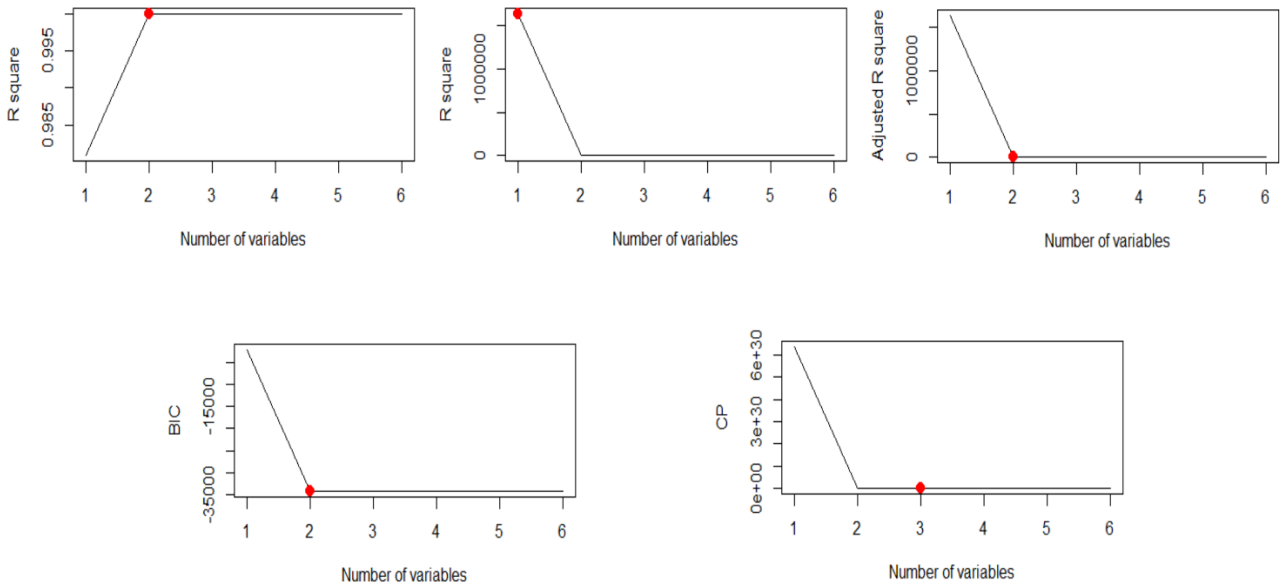
Plotting the logarithmic function of the regressor variables as a scatter plot in matrix plot:

```
pairs(pedestrian %>% select(c(3,4,5,6,7,8)), log ="xy")
```



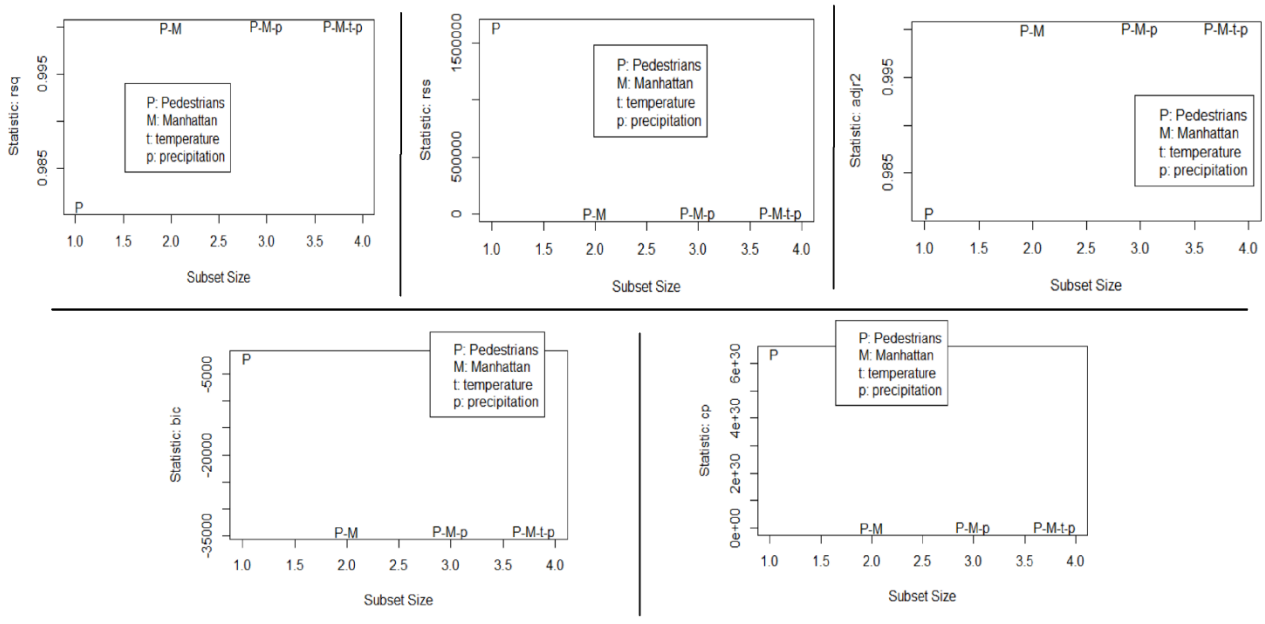
(c) Repeat it for all possible models in R with the code snippets:

```
leaps <- regsubsets(pedestrian_leaps$Brooklyn~.,data=pedestrian_leaps,nvmax = 6)

> reg.summary <- summary(leaps)
> names(reg.summary)
[1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"

> data.frame(RSQ = which.max(reg.summary$rsq),
+   RSS = which.max(reg.summary$rss), AdjR2 = which.max(reg.summary$adjr2), BIC
= which.min(reg.summary$bic), CP = which.min(reg.summary$cp))

RSQ RSS AdjR2 BIC CP
    2   1     2   2  3


> par(mfrow=c(3,2))
> plot(reg.summary$rsq, xlab = "Number of variables", ylab = "R square", type =
"l")
> points(2, reg.summary$rsq[2], col="red", cex = 2, pch=20)

> plot(reg.summary$rss, xlab = "Number of variables", ylab = "RSS", type = "l")
> points(1, reg.summary$rss[1], col="red", cex = 2, pch=20)

> plot(reg.summary$bic, xlab = "Number of variables", ylab = "BIC", type = "l")
> points(2, reg.summary$bic[2], col="red", cex = 2, pch=20)

> plot(reg.summary$cp, xlab = "Number of variables", ylab = "CP", type = "l")
> points(3, reg.summary$cp[3], col="red", cex = 2, pch=20)
```



Explanation: The lines plots correspond to the number of regressors in the models, which are given in the horizontal axis at the bottom of the figure as "Number of Variables," which range from 2 to 5 for this data set. By "variables," R means the number of β's, including the intercept. Thus, 2 is the smallest number of parameters for BIC while 3 is the smallest for CP. In SAS, the same plot will be put in a way that it shows for each model the best regressors marked via stars.

Following are all the subset plots for each variable to identify the best relationship for 2,3,4 and 5 order:



Explanation: We see in the above plots that for different models the best fit model has different output but largely we observe that P-M (that is pedestrian and Manhattan) seem to have the strongest relationship for two pairs but P-M-p has the best three step relations where in for 4 it is P-M-t-p (t and p are temperature and precipitation respectively)

5.2. Summary of forward and backward models:

If we have do it for all OLS models together then we do it using a ols_step_all_possible function:

```
> model <- lm(Pedestrians ~ Towards.Manhattan + Towards.Brooklyn + temperature,
data=pedestrian)
> ols_step_best_subset(model)
```

```
                Best Subsets Regression
-------------------------------------------------------------
Model Index    Predictors
-------------------------------------------------------------
     1         Towards.Brooklyn
     2         Towards.Manhattan Towards.Brooklyn
     3         Towards.Manhattan Towards.Brooklyn temperature
-------------------------------------------------------------
```

```
                                                Subsets Regression Summary
-----------------------------------------------------------------------------------------------------------------------------------------------------------
                    Adj.        Pred
Model    R-Square    R-Square    R-Square        C(p)           AIC          SBIC          SBC          MSEP          FPE          HSP          APC
-----------------------------------------------------------------------------------------------------------------------------------------------------------
  1        0.9809      0.9808      0.9806    5.716279e+30      6134.5970     4709.6584     6147.2408    12423.3487    12423.1496    24.8967     0.0193
  2        1.0000      1.0000           1          2.0092    -26170.5413   -27589.4316   -26153.6828        0.0000        0.0000     0.0000     0.0000
  3        1.0000      1.0000           1          4.0000    -26168.5505   -27587.4247   -26147.4775        0.0000        0.0000     0.0000     0.0000
-----------------------------------------------------------------------------------------------------------------------------------------------------------
AIC: Akaike Information Criteria
 SBIC: Sawa's Bayesian Information Criteria
 SBC: Schwarz Bayesian Criteria
 MSEP: Estimated error of prediction, assuming multivariate normality
 FPE: Final Prediction Error
 HSP: Hocking's Sp
 APC: Amemiya Prediction Criteria
```

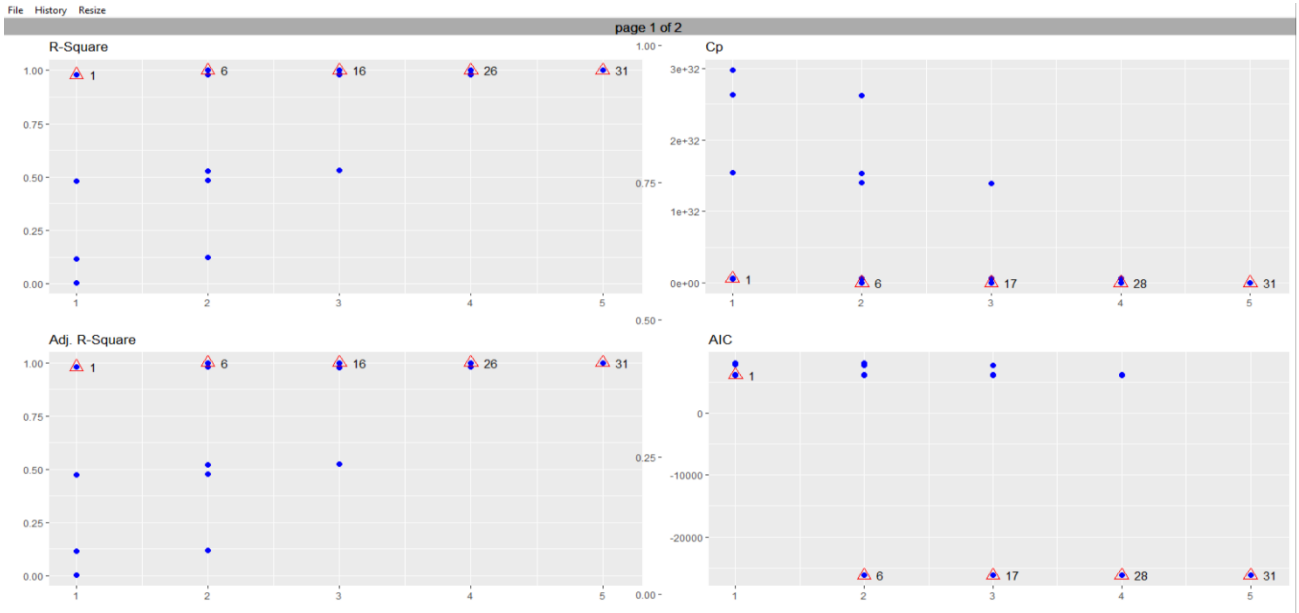
```
> library(olsrr)
> all_model <- lm(Pedestrians ~., data = pedestrian_leaps)
> ols_step_best_subset(all_model)
```

```
                    Best Subsets Regression
-------------------------------------------------------------------
Model Index    Predictors
-------------------------------------------------------------------
     1         Brooklyn
     2         Manhattan Brooklyn
     3         Manhattan Brooklyn WS
     4         Manhattan Brooklyn WS temperature
     5         Manhattan Brooklyn WS temperature precipitation
-------------------------------------------------------------------
```

n	predictors	rsquare	adjr	cp	aic	sbic
1	Brooklyn	0.98087562	0.980837217	5.71E+30	6134.597	4709.658
1	Manhattan	0.979040381	0.978998294	6.26E+30	6180.414	4755.475
1	WS	0.481753886	0.474380466	1.55E+32	7796.34	6359.402
1	temperature	0.117010048	0.115236976	2.64E+32	8050.772	6625.833
1	precipitation	0.004113619	0.002113847	2.97E+32	8110.932	6685.993
2	Manhattan Brooklyn	1	1	1.30E+00	-26170.541	-27589.423
2	Brooklyn WS	0.981791199	0.981494517	5.43E+30	6124.068	4685.129
2	Brooklyn temperature	0.980956556	0.980879923	5.68E+30	6134.476	4707.538

2	Brooklyn precipitation	0.980878711	0.980801765	5.71E+30	6136.516	4709.578
2	Manhattan WS	0.97931926	0.978982303	6.17E+30	6187.716	4748.778
2	Manhattan temperature	0.979070407	0.978986183	6.25E+30	6181.697	4754.758
2	Manhattan precipitation	0.979060365	0.978976101	6.25E+30	6181.937	4754.998
2	WS temperature	0.529970894	0.522312578	1.40E+32	7749.512	6310.574
2	WS precipitation	0.485117403	0.476728277	1.54E+32	7795.084	6356.146
2	temperature precipitation	0.122446467	0.118915065	2.62E+32	8049.684	6622.745
3	Manhattan Brooklyn WS	1	1	-3.94E+00	-26163.91	-27594.654
3	Manhattan Brooklyn temperature	1	1	3.29E+00	-26168.551	-27587.413
3	Manhattan Brooklyn precipitation	1	1	3.30E+00	-26168.549	-27587.412
3	Brooklyn WS temperature	0.981893285	0.981560713	5.40E+30	6123.256	4682.318
3	Brooklyn WS precipitation	0.981806459	0.981472292	5.43E+30	6125.648	4684.71
3	Brooklyn temperature precipitation	0.980958189	0.980843017	5.68E+30	6136.434	4707.495
3	Manhattan WS temperature	0.979368759	0.978989818	6.16E+30	6188.518	4747.58
3	Manhattan WS precipitation	0.979319377	0.978939528	6.17E+30	6189.714	4748.775
3	Manhattan temperature precipitation	0.979093063	0.97896661	6.24E+30	6183.155	4754.217
3	WS temperature precipitation	0.532352283	0.523762836	1.40E+32	7748.973	6308.034
4	Manhattan Brooklyn WS temperature	1	1	-1.95E+00	-26161.916	-27592.611
4	Manhattan Brooklyn WS precipitation	1	1	-1.99E+00	-26161.963	-27592.657
4	Manhattan Brooklyn temperature precipitation	1	1	5.29E+00	-26166.559	-27585.403
4	Brooklyn WS temperature precipitation	0.981906988	0.981536988	5.40E+30	6124.878	4681.939
4	Manhattan WS temperature precipitation	0.979368804	0.978946898	6.16E+30	6190.517	4747.579
5	Manhattan Brooklyn WS temperature precipitation	1	1	0.00E+00	-26159.97	-27590.614

```
final_model <- lm(Pedestrians ~ Towards.Manhattan + Towards.Brooklyn + temperatu
re, data=pedestrian_leaps)
plot(final_model,4)
```

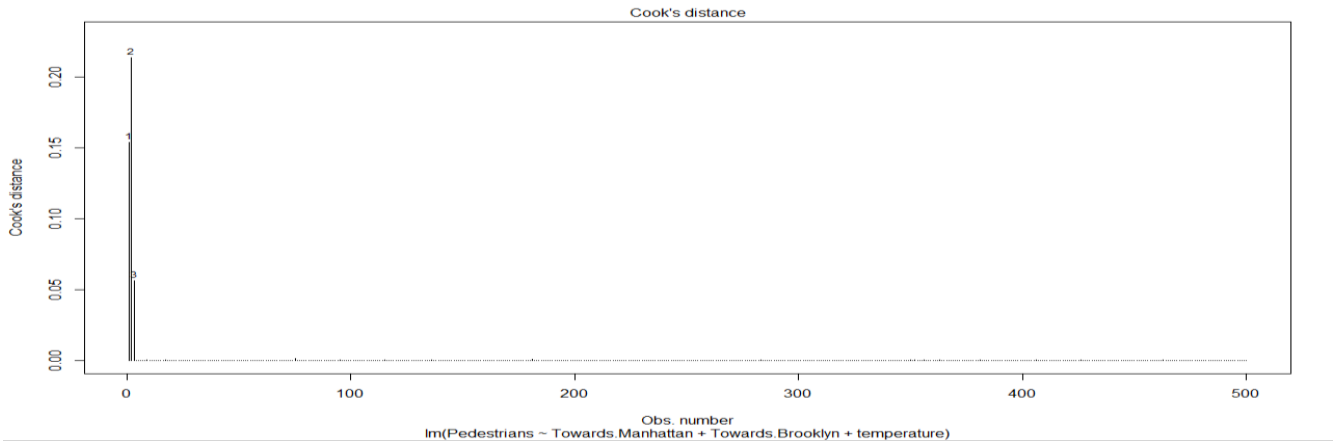


(d) Explanation to the criteria plot above and the final conclusion on the fitted model:

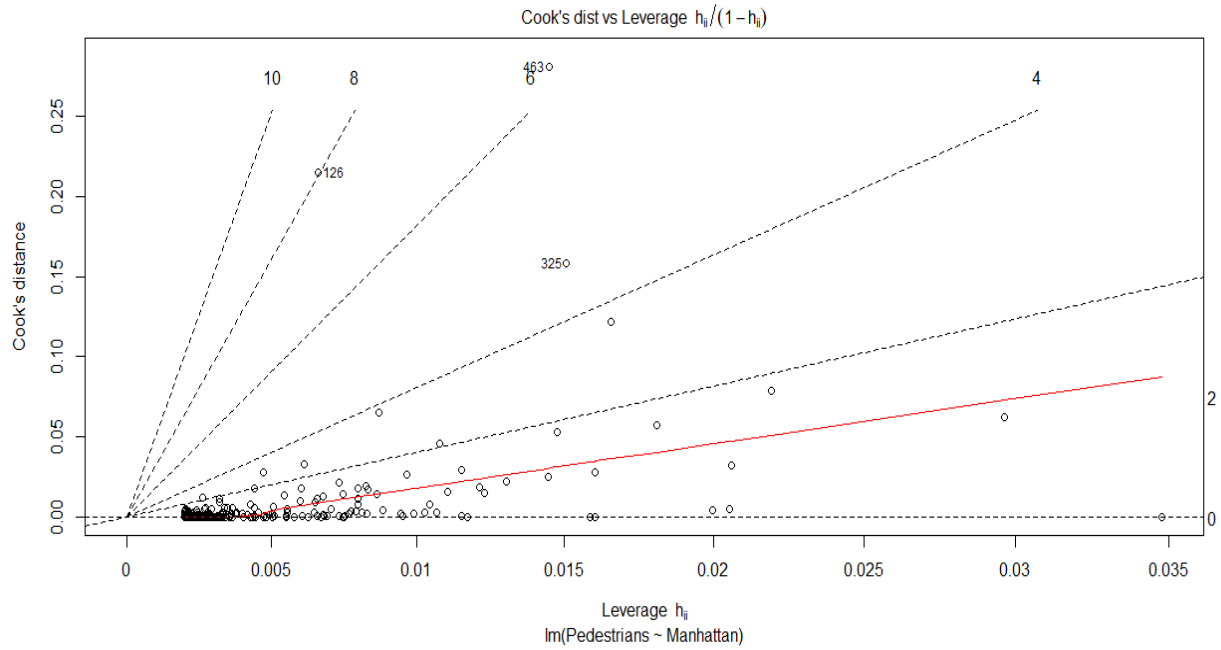
In the Criteria Plot and Summary Table, that Cp and BIC achieve their minimum criterion value for 6 parameters. Also, true for backward and forward selection model in the latter half of the question. The goal is select the model that has the smallest criterion value. I would prefer to choose the model fewer regressors, thus I would prefer the model with 5 variables, which are Pedestrian, Manhattan, Brooklyn, Temperature, Precipitation.

(e) For the model you selected, show the diagnostic plots with labels, as in *Program 8.6.3*, and interpret.

```
final_model <- lm(Pedestrians ~ Towards.Manhattan + Towards.Brooklyn + temperatu
re, data=pedestrian_leaps)
plot(final_model,4)
```



```
forward_model <- lm(Pedestrians ~ Manhattan, data = pedestrian_leaps)
plot(forward_model, which = 6)
```



5.3. Forward Stepwise Model Selection

(a) "Forward selection method" as executed on R :

```
ols_step_forward_p(model)
```

```
Forward Selection Method
---------------------------

Candidate Terms:

1. Towards.Manhattan
2. Towards.Brooklyn
3. weather_summary
4. temperature
5. precipitation
```

We are selecting variables based on p value...

```
Variables Entered:
```

✔ **Towards.Manhattan**
✔ **Towards.Brooklyn**
✔ **Temperature**

```
## No more variables to be added.

## Final Model Output
## ------------------
##
## Model Summary
## ------------------------------------------------------------------
## R 0.604 RMSE 89.724
## R-Squared 0.365 Coef. Var 51.913
```

```
## Adj. R-Squared 0.359 MSE 8050.401
## Pred R-Squared 0.341 MAE 63.130
## ----------------------------------------------------------------
## RMSE: Root Mean Square Error
## MSE: Mean Square Error
## MAE: Mean Absolute Error
##
## ANOVA
## ----------------------------------------------------------------------------
## Sum of
## Squares DF Mean Square F Sig.
## ----------------------------------------------------------------------------
## Regression 1512812.171 3 504270.724 62.639 0.0000
## Residual 2632481.019 327 8050.401
## Total 4145293.190 330
## ----------------------------------------------------------------------------
##
## Parameter Estimates
## ----------------------------------------------------------------------------
## model Beta Std. Error Std. Beta t Sig lower upper
## ----------------------------------------------------------------------------
## (Intercept) 29.557 14.467 2.043 0.042 1.097 58.016
## Manhattan   25.462 4.414 0.307 5.769 0.000 16.779 34.145
## Brooklyn     0.065 0.013 0.282 4.973 0.000 0.039 0.090
## Temperature 35.727 14.831 0.135 2.409 0.017 6.550 64.904
## ----------------------------------------------------------------------------
##
## Selection Summary
## ------------------------------------------------------------------------------
## Variable Adj.
## Step Entered R-Square R-Square C(p) AIC RMSE
## ------------------------------------------------------------------------------
## 1 Manhattan    0.2670 0.2647 48.6700 3965.6420 96.1033
## 2 Brooklyn     0.3537 0.3497 6.2362 3925.9759 90.3785
## 3 Temperature  0.3649 0.3591 2.4609 3922.1539 89.7240
## ------------------------------------------------------------------------------
```

As I increase the parameters I see a warning which says there is nothing needed more as the model selection on a perfectly fit data is a futile exercise.

(b) Comparing the results with the best subsets method:

After comparison, I see the results are consistent between forward selection model and the best fit model. The variables used for the models are same for both exercise.

5.4. Variance Inflation

(a) The VIF of regressor k is denoted as $VIF_k = (1 - R^2_k)^{-1}$, where, R^2_k is the R-square statistic resulting from the regression of k^{th} variable on the other x-variables in the model. If the correlation between that variable and the remaining x-variables is low, then the VIF will be close to 1.0. But, if the correlation is high, then the VIF can be enormous. If VIF value is absurdly high. It means we should get rid of one or the other of those variables.

(b) Evaluating VIF output and understanding any outrageous values:

```
> vif(final_model)
Towards.Manhattan  Towards.Brooklyn       temperature
          12.7890           12.7540            1.1328
```

I do see some outrageous values as the VIF is far greater than one for parameters like – Manhattan, and Brooklyn (which shows the no of pedestrians going each day towards Manhattan or Brooklyn from the bridge).

5.5. Cook's D

(a) Explanation of Cook's D and how is it computed:

Cook's distance is an aggregate influence measure, which shows the effect of the i^{th} case on all *n* fitted values." For the first case, i=1, compute $\hat{y}$ for all n cases, but without the first case in the fitting.

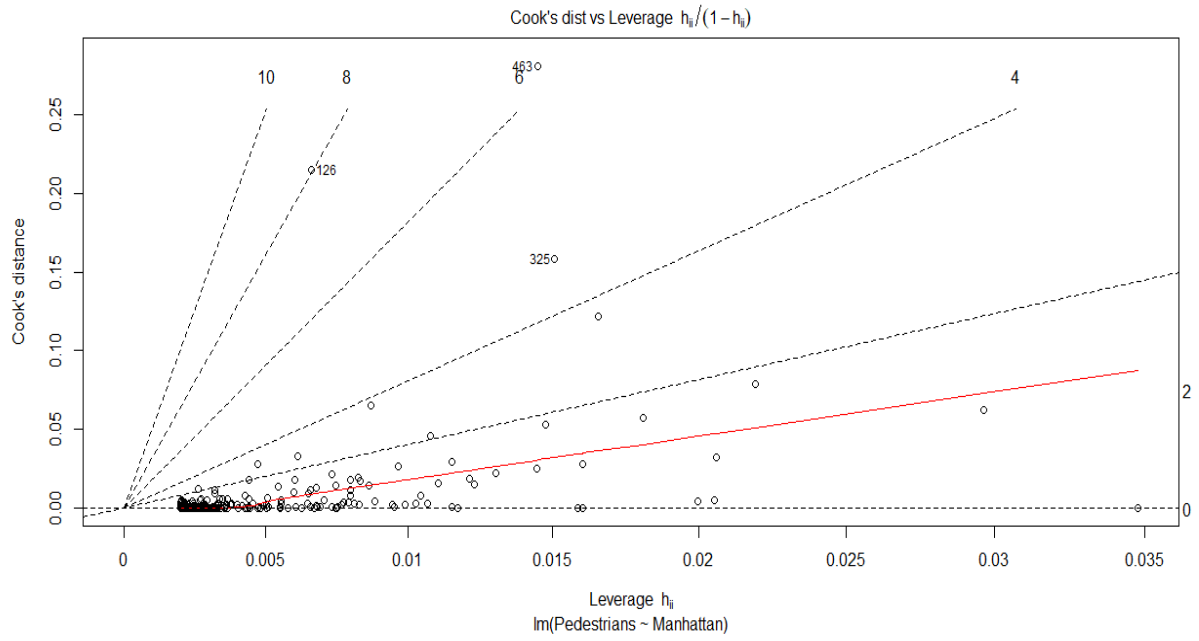
Cook's D is computed by taking the sum of the squares of the resulting residuals and divide by the MSE times p.

$$D_i = \frac{\sum_{j=1}^{n} (\hat{y}_j - \hat{y}_{j(i)})^2}{pMSE}$$

If n=4 and i=2, the numerator would be,

$(\hat{y}_1 - \hat{y}_{1(2)})^2 + (\hat{y}_2 - \hat{y}_{2(2)})^2 + (\hat{y}_3 - \hat{y}_{3(2)})^2 + (\hat{y}_4 - \hat{y}_{4(2)})^2$

(b) As from the figure below we can see that some of the observations form our data do have excessive influence. Like the observations 2, 8, 10. 4 and 463.



Chapter 6: Special Topic

6.1 Understanding Logistic Regression:

It is a statistical model that in its basic form uses a logistic function to model a binary dependent variable. The essential difference between Linear Regression and a Logistic regression is used when the dependent variable is binary in nature. In contrast, Linear regression is used when the dependent variable is continuous, and nature of the regression line is linear.

Types of Logistic Regression:

a) **Binary Logistic Regression**
   The categorical response has only two 2 possible outcomes. Example: Spam or Not

b) **Multinomial Logistic Regression**
   Three or more categories without ordering. Example: Predicting which food is preferred more (Veg, Non-Veg, Vegan)

c) **Ordinal Logistic Regression**
   Three or more categories with ordering. Example: Movie rating from 1 to 5

Logistic Model: Sigmoid Function:

Let us try to understand logistic regression by understanding the logistic model. As in linear regression let's represent our hypothesis(Prediction Of Dependent Variable) in classification. In classification our hypothesis representation which tries to predict the binary outcome of either o or 1, will look like,

hθ(x) = g(θ T x) = 1/ 1 + e −θ T x ,
Here g(z) = 1/(1 + e ^−z), is called the l***ogistic function or the sigmoid function:***

g(z): is a representation of logistic function, which we also call a Sigmoid function. From the above visual representation of sigmoid function, we can easily decipher how this curve describes many real-world situations, like population growth. In the initial stages it shows an exponential growth, but after some time, due to the competition for certain resources (bottle neck), the growth rate decreases until it gets to a stalemate and there is no growth.

6.2 Implementing Logistic Regression in R:

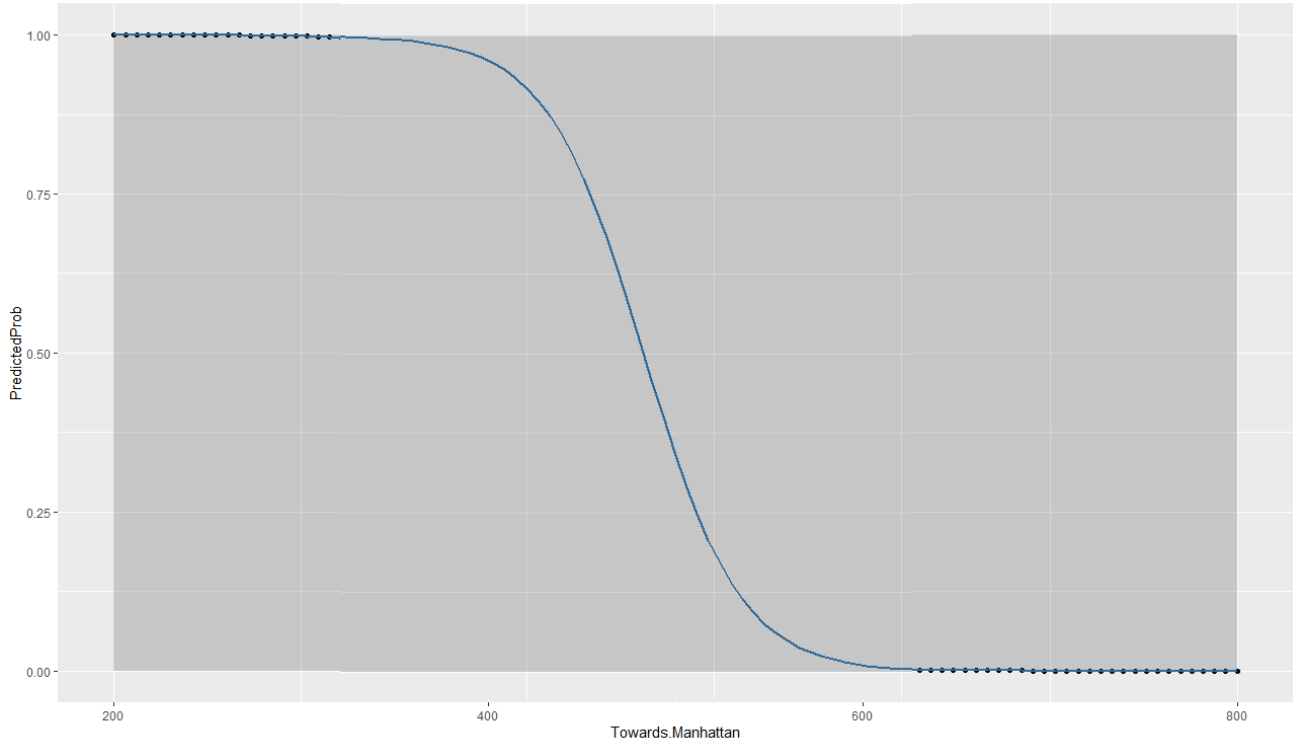
I am using the weather summary data to get the logistic regression on the 'total pedestrians' recorded from all directions coming to Brooklyn Bridge. I sue weather summary because I want to know how the affect of whether is consistent throughout and it does affect the frequency but not the overall random normal distribution or urban walking population of NYC.

```
# Inspect the data
sample_n(pedestrian_log, 3)

# Split the data into training and test set
set.seed(123)
training.samples <- pedestrian_log$x2_dummy %>%
  createDataPartition(p = 0.8, list = FALSE)

train.data  <- pedestrian_log[training.samples, ]
test.data <- pedestrian_log[-training.samples, ]

# Check the summary of the data
summary(pedestrian)

xtabs(~Towards.Manhattan + Towards.Brooklyn, data = pedestrian)

#Defining the logic of the logistic regression
mylogit <- glm(x2_dummy ~ Towards.Manhattan + Towards.Brooklyn + weather_dummy,
data = pedestrian, family = "binomial")
summary(mylogit)

confint(mylogit)
confint.default(mylogit)

library(aod)
wald.test(b = coef(mylogit), Sigma = vcov(mylogit), Terms = 1:4)

l <- cbind(0, 0, 0, 1, -1, 0)
wald.test(b = coef(mylogit), Sigma = vcov(mylogit), L = l)

exp(coef(mylogit))

newdata1 <- with(pedestrian, data.frame(Towards.Manhattan = mean(Towards.Manhatt
an), Towards.Brooklyn = mean(Towards.Brooklyn), weather_dummy = factor(1:4)))
newdata1

newdata1$weather_dummy <- as.numeric(newdata1$weather_dummy)
str(newdata1)
pedestrian$weather_dummy <- as.numeric(pedestrian$weather_dummy)
str(pedestrian)

newdata1$weather_dummy <- predict(mylogit, newdata = newdata1, type = "response"
)
newdata1

newdata2 <- with(pedestrian, data.frame(Towards.Manhattan = rep(seq(from = 200,
to = 800, length.out = 100), 4), Towards.Brooklyn = mean(Towards.Brooklyn), weat
her_dummy = rep(1:1, each = 100)))
```

```
newdata3 <- cbind(newdata2, predict(mylogit, newdata = newdata2, type = "link",
se = TRUE))

newdata3 <- within(newdata3, {
  PredictedProb <- plogis(fit)
  LL <- plogis(fit - (1.96 * se.fit))
  UL <- plogis(fit + (1.96 * se.fit))
})

head(newdata3)
```

#Plotting the output of the logistic regression:

```
library(ggplot2)

ggplot(newdata3, aes(x = Towards.Manhattan, y = PredictedProb)) +
  geom_ribbon(aes(ymin = LL, ymax = UL), alpha = 0.2) +
  geom_line(aes(colour = weather_dummy), size = 1)
```



Conclusion: We see our model gives a near perfect logistic regression for total pedestrians crossing Manhattan on a given day and how our best fit model scores against the untrained test data using the underlying logistic function (***sigmoid function***), we discussed above.

#Pass the test data and see how our best fit model scores against

```
themlogmodel_score = model.score(X_test, y_test)
print("This is how my Model Scored:\n\n", logmodel_score)
```

Answer: This is how my Model scored: 0.962756263. That is, 96% accuracy in logistic regression model for the pedestrians coming from Manhattan in NYC for different weather conditions.