Big Data Assignment 1: Benjamin Terrill

15622143@students.lincoln.ac.uk

Table of Contents

Section 1: Data Loading and Pre-Processing (30%) 2
Task 1: Loading Dataset 2
Task 2: Removing NaN Data from Dataset 2
Task 3: Writing New File 2
Task 4: Step 1-3 for New Dataset 2
Section 2: Statistical Analysis and Data Visualisation (30%) 3
Task 5: Table of data 3
Task 6: Creating plots 3
A box plot of 'mpg': 3
A box plot of 'Acceleration': 4
A box plot of 'Horsepower': 4
A box plot of 'Weight' 4
A scatter plot of 'acceleration' vs 'mpg': 5
A scatter plot of 'Horsepower' and 'mpg': 5
A scatter plot of 'Weight' and 'Horsepower': 6
..... 6
Section 3: Regression Analysis (40%) 7
Task 7: Linear regression of 'acceleration' vs 'mpg' 7
Task 8: 7
Task 9: Linear regression of 'horsepower' vs 'mpg' 8
Task 10: 9
Task 11: Linear regression of 'weight' vs 'horsepower' 9
Task 12 10

Section 1: Data Loading and Pre-Processing (30%)

Task 1: Loading Dataset

Loaded the file using xlsread('data_train.xls') and I used [rownum, colnum]=size(A) to find the number of columns and rows in the file.

Below it the number of columns and rows, which do match the original excel file.

```
Displays 8 Columns and 305 Rows
```

Task 2: Removing NaN Data from Dataset

I used the following code to remove find and then remove any NaN cells;

```
r = find(isnan(A(:,1)));
A(r,:)=[]
```

Task 3: Writing New File

After removing the NaN cells from the spreadsheet the size of the new file was.

```
Displays 8 columns and 296 rows
```

I then used the following to write it to a new file names 'data_train2.xls'.

```
filenameWrite = 'data_train2.xls'
xlswrite(filenameWrite,A)
```

Task 4: Step 1-3 for New Dataset

I repeated the above tasks for the 'data_test.xls' file.

The original size was.

```
Displays 101 rows and 8 columns
```

And after removing the NaN cells in the file it displayed.

```
Displays 96 rows and 8 columns
```

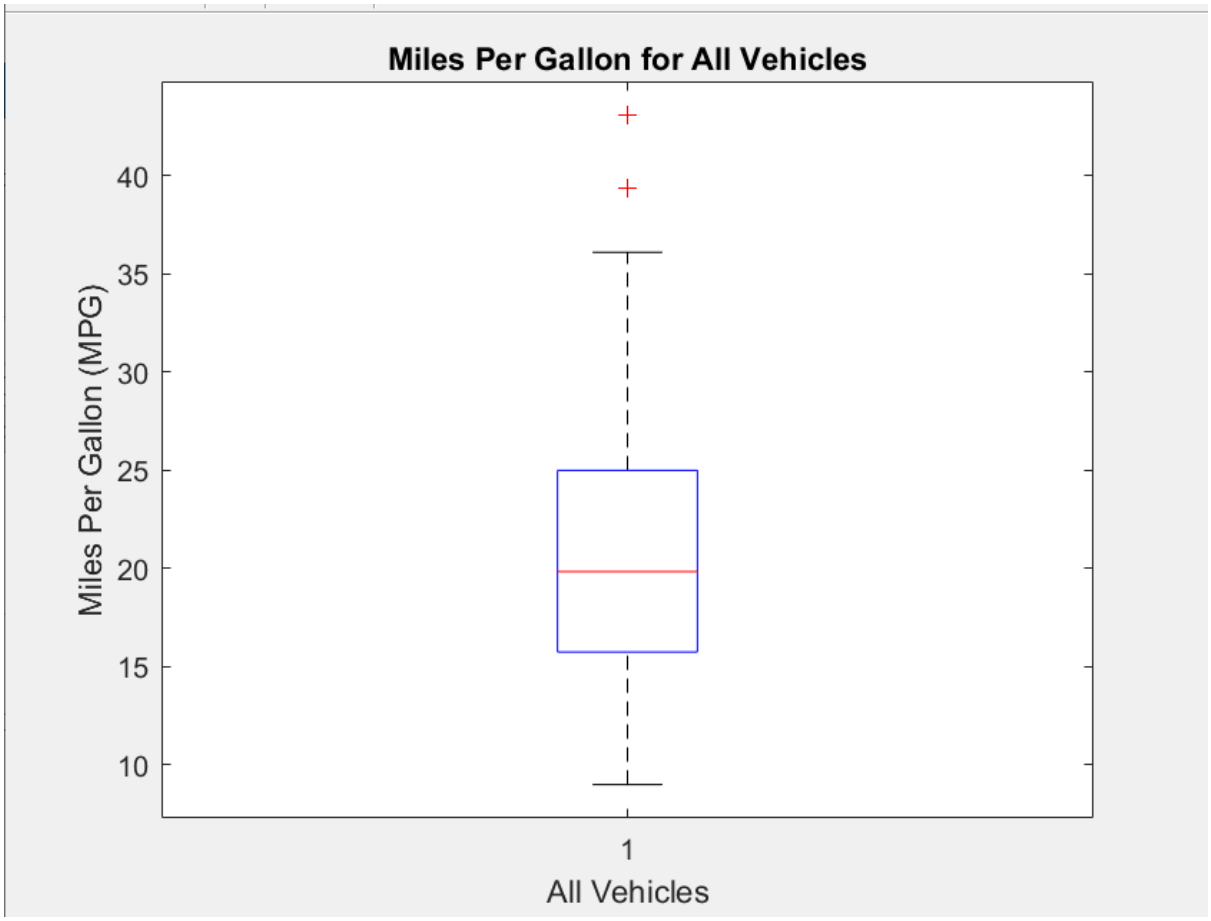
Section 2: Statistical Analysis and Data Visualisation (30%)

Task 5: Table of data

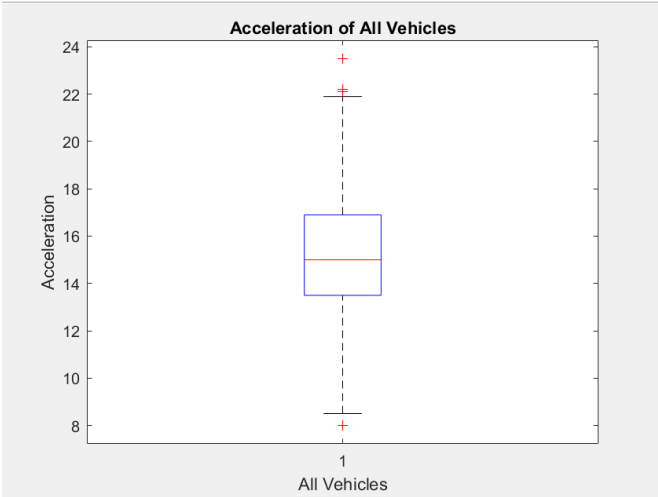
	Variable	Mean	Median	Min	Max	Standard Deviation
MPG	**0.00039**	**20.8**	**19.9**	**9**	**43.1**	**6.31**
Acceleration	**0.000072**	**15.2**	**15.0**	**8**	**23.5**	**2.68**
Horsepower	**0.01630**	**112.1**	**100.0**	**46**	**230.0**	**40.38**
Weight	**7.86644**	**3134.8**	**3094.0**	**1613**	**5140**	**889.92**

Task 6: Creating plots

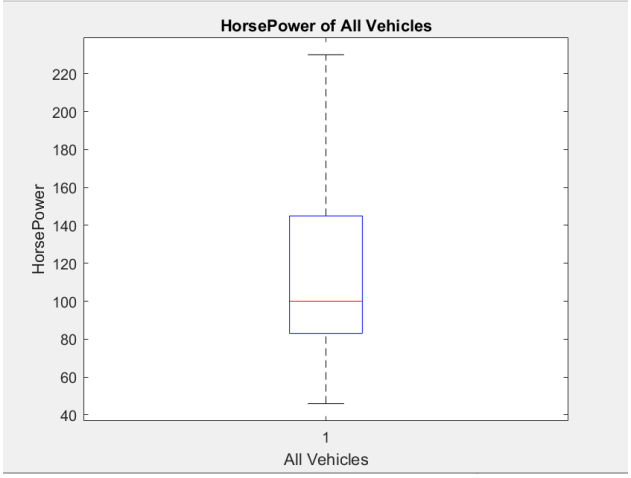
A box plot of 'mpg':



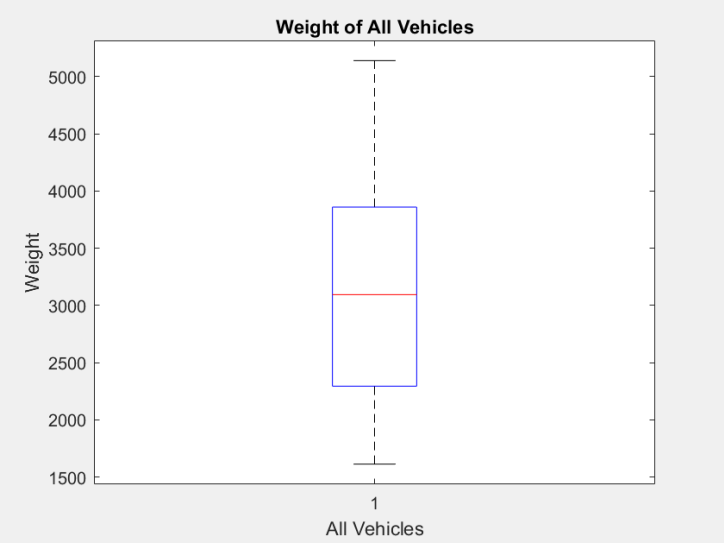
A box plot of 'Acceleration':



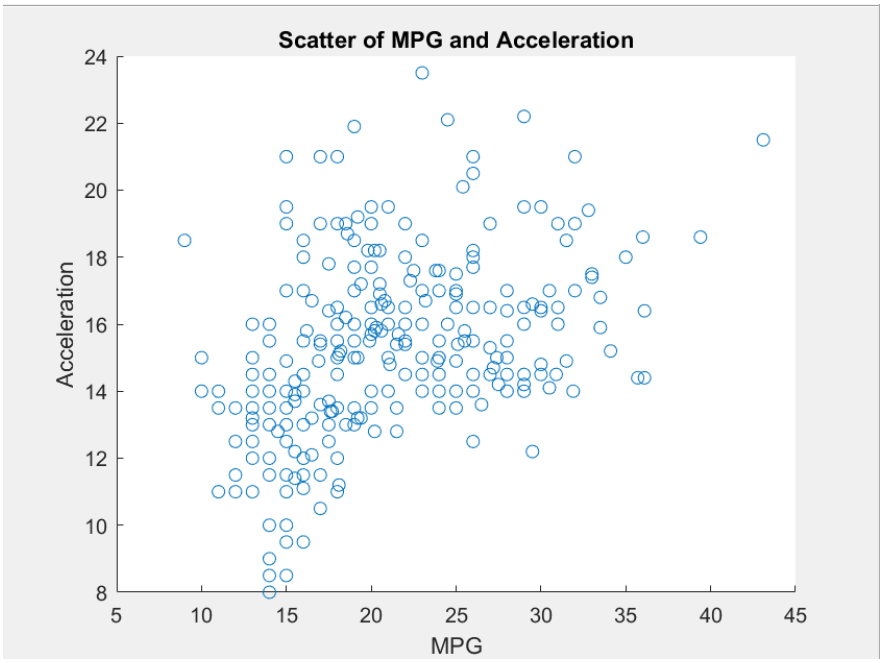
A box plot of 'Horsepower':



A box plot of 'Weight':

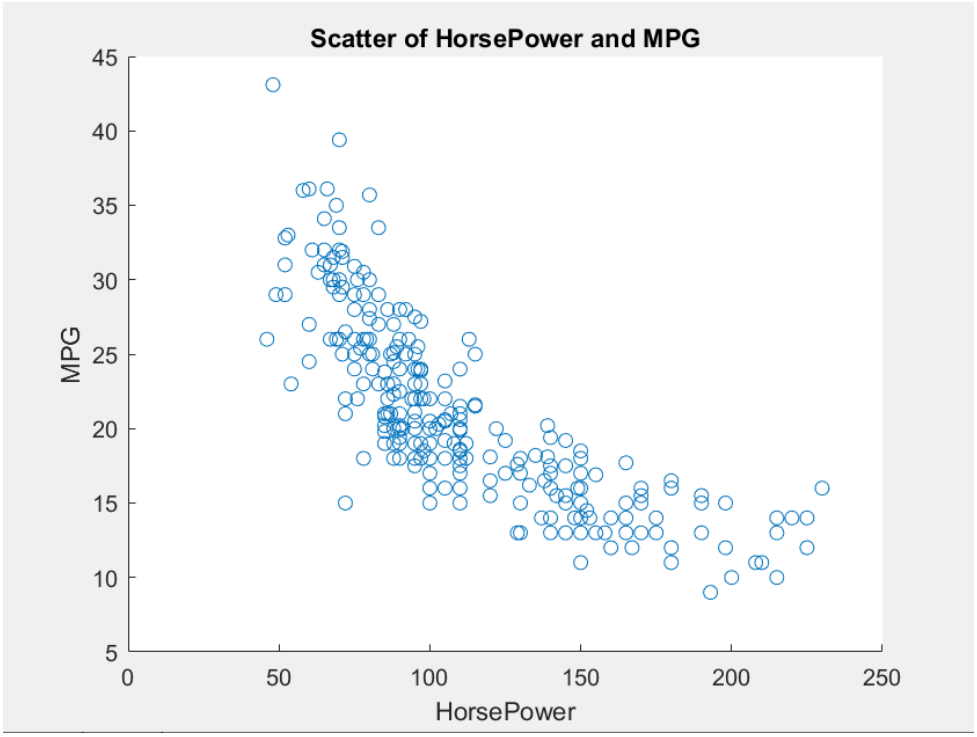


A scatter plot of 'acceleration' vs 'mpg':



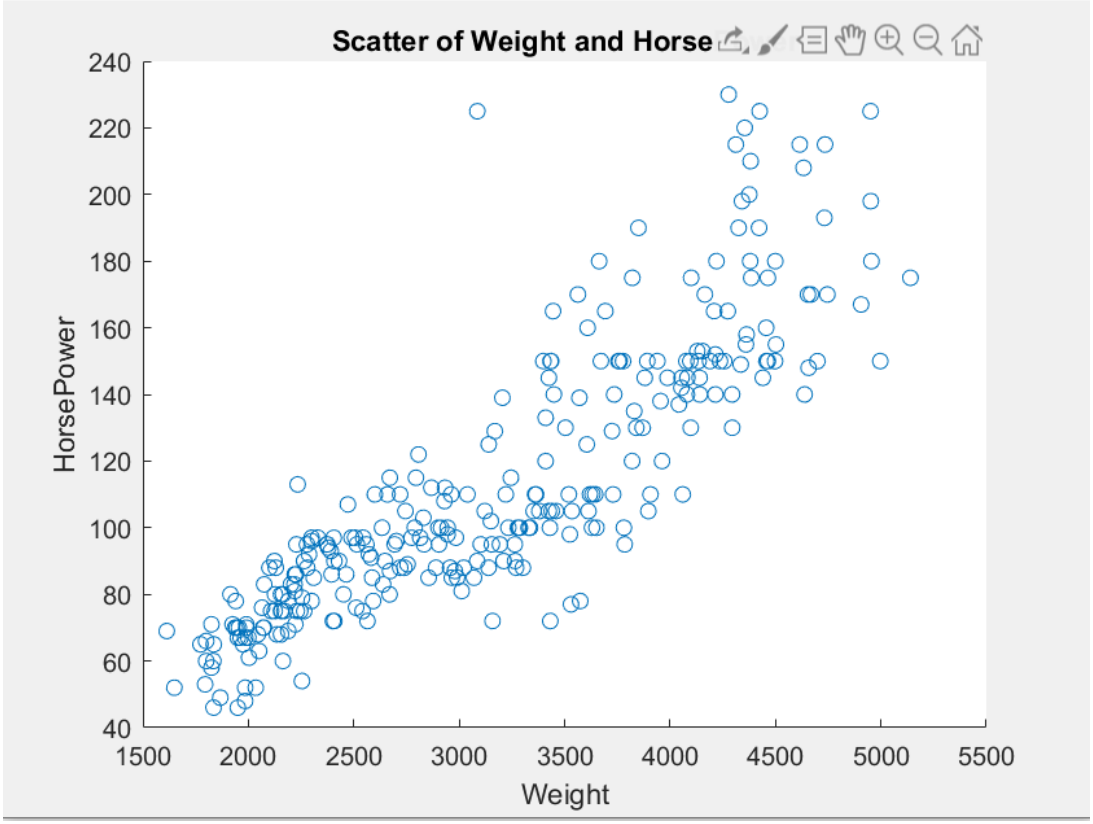
The scatter graph above shows acceleration and mpg. From the graph we can determine that there is a positive correlation between the two, however, it is a low correlation as there are too many outliers where acceleration is high while mpg has stayed low.

A scatter plot of 'Horsepower' and 'mpg':



The above scatter graph shows horsepower and mpg. It has a strong negative correlation as when horsepower of the cars increase the mpg decrease rapidly. From this we can infer that faster cars with more horsepower and less efficient that cars with lower horsepower. There are also much fewer outliers in this graph which shows that it is a more consistent trend.

A scatter plot of 'Weight' and 'Horsepower':

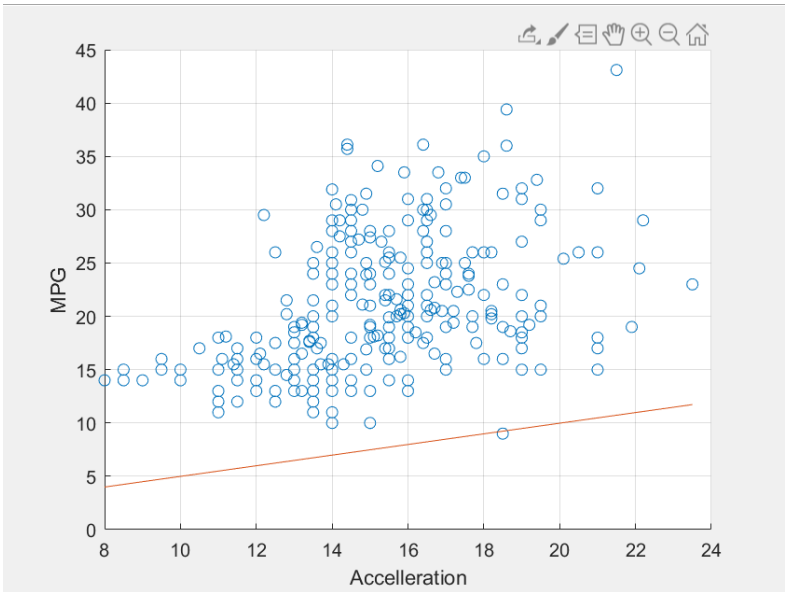


The scatter graph above shows weight vs horsepower of each car. This scatter graph shows a strong positive correlation. As the horsepower of the car increases the weight of the car also increases at a steady rate.

From this we can see that the more horsepower is needed for heavier cars to run. However, there are outliers in this graph as well and the data is not so consistent. This means that it can be hard to predict the exact increase in horsepower and the weight of the car goes up.

Section 3: Regression Analysis (40%)

Task 7: Linear regression of 'acceleration' vs 'mpg'



To perform linear regression you need two variables. The X-axis will contain the independent variable and the Y-axis contains the dependant variable.

I then used the following code to get the slope of the line of best fit.

```
b1 = x/y %Slope of the line%
```

This is then used to plot the graph with a line of best fit. There is no correlation in this graph so the line of best fit is not ideal.

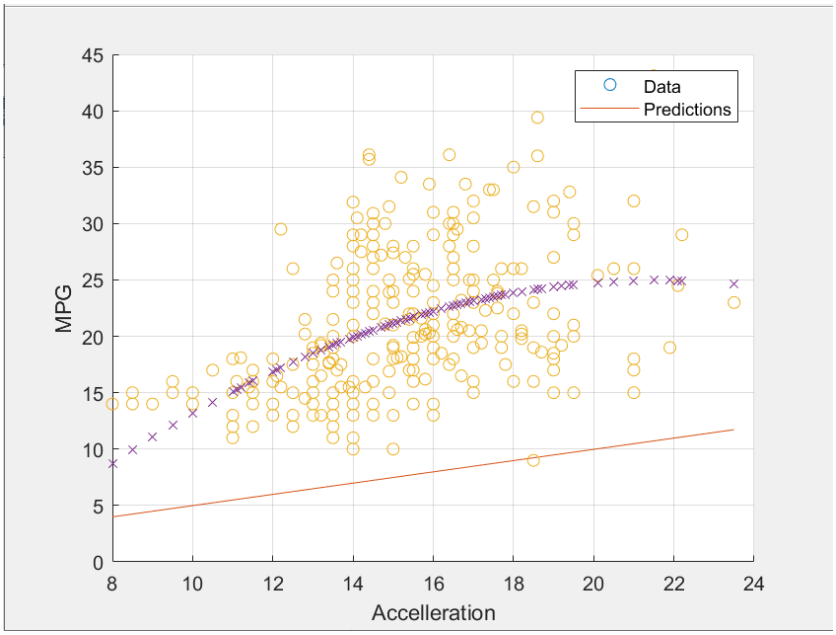
The average mean of squared errors is: mse = 31.252087209146946

The predicted values for the linear model are:

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	4.30401097102173	1.87505012293116	2.29541115641939	0.0224135267862434
x	1.0819937234924	0.121444301949017	8.90938237634757	5.49158922339111e-17

Task 8:



The mean of squared errors for the predicted values for 'mpg' are:

mse2 =

30.409906359710078

This is different to the ground truth mse I got in task 7.

Task 9: Linear regression of 'horsepower' vs 'mpg'

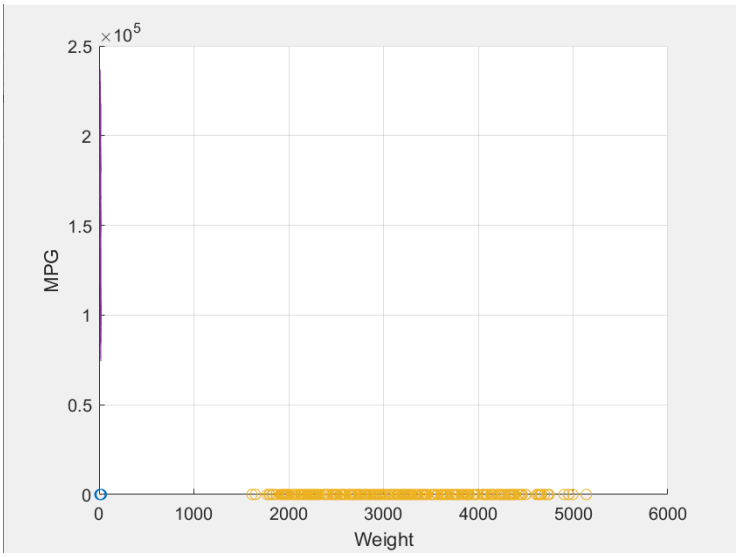
For this linear regression model, I used Weight as the independent variable and MPG as the dependant variable.

I created a slope value using; b2 = x2/y2

I then used the equation yCalc = b2*x2

And I created a scatter graph that plots the data with the line of best fit.

Unfortunately, for this data set the values have an extremely different range so the values appear all across the bottom of the graph. I was not able to find a solution for this to display the data so it can be read.




```
%Task 9%

x2=data4 %Independant variable - Weight%
y2=data1 %Dependant variable - MPG%

format long
b2 = x2/y2 %Slope of the line%

yCalc3 = b2*x2
scatter(x2,y2)
hold on
plot(x,yCalc3)
xlabel('Weight')
ylabel('MPG')
grid on


tbl3 = table(x2,y2);
lm = fitlm(tbl3, 'linear')

y_pred = -0.0062*x+40.4754
mse = mean((y_pred-y).^2)
```

The average mean of squared errors is: mse = 4.249692509834800e+02

The predicted values for the linear model are:

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	40.4754697524832	0.631522473915159	64.0918913012759	8.24390065472448e-175
x2	-0.00629088074838157	0.000193872687441387	-32.4485147000579	3.5886695631699e-99

Task 10:

The mean of squared errors for the predicted values for ‘mpg’ are:

mse3 =

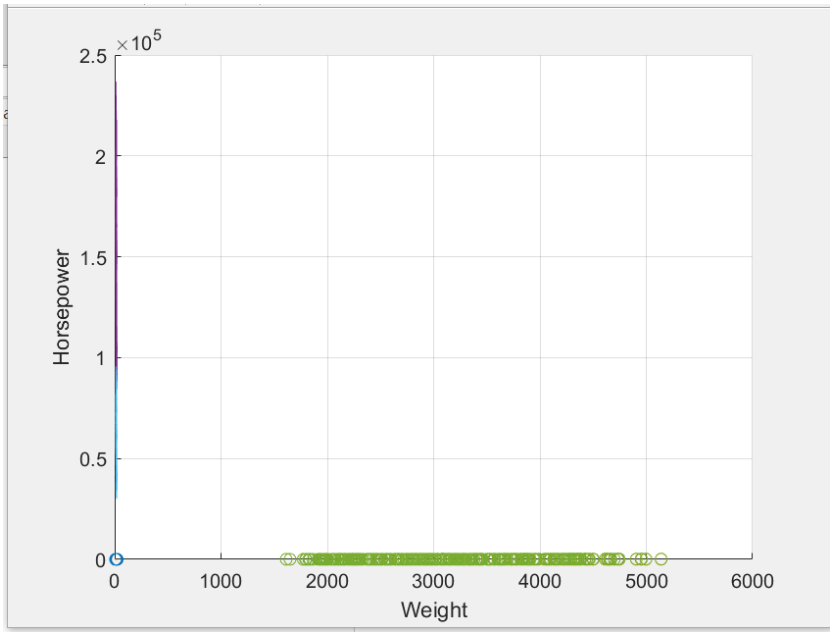
31.252085083798882

This is much different from the 4.2496 I got in the previous task.

This means that there is less correlation between the two variables so the data has more erroneous values that affect the prediction compared with the actual graph.

Task 11: Linear regression of ‘weight’ vs ‘horsepower’

For this final linear regression model I used Weight as the independent variable and Horsepower as the dependant variable.



Again I created a scatter graph and then a slope value using; b3=x3/y3

I had the same problem as previous tasks where I could not get the two y axis to work to display the data in a more clear way to that the data is not all along the bottom due to the high difference between Horsepower and Weight.

The average mean of squared errors for this is: mse =

1.001258915197838e+03

The predicted values for the linear model are:

Estimated Coefficients:

	Estimate	SE	tStat	pValue
(Intercept)	-10.8598203523361	4.39347978149278	-2.471803875844	0.0140095749244897
x3	0.0392196769487422	0.00134876551134716	29.0782027111364	1.73536653557852e-88

Task 12

The average mean of squared errors for this section is

mse3 =

31.252085083798882

```
%Task 10%

mdl = fitlm(x,y,'linear');
y_pred6 = predict(mdl,x);

plot(x,y,'o',x,y_pred6,'x')
legend('Data', 'Predictions')

mse5 = mean((y_pred6-y).^2)
```