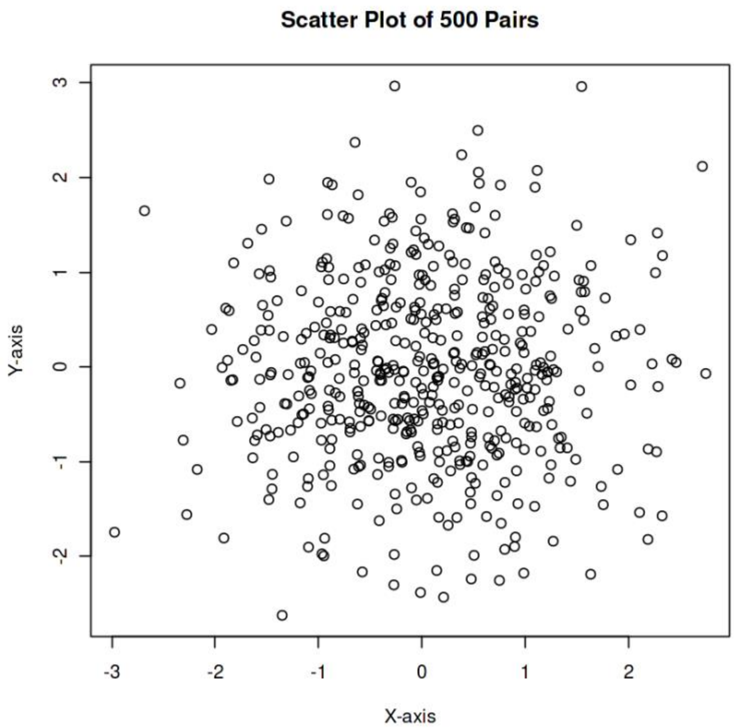
Filippo Riva
MT4420

Assignment 2 Report

1. Use set.seed(7) for the following.
a. Generate a plot of 500 pairs using the rnorm command. Print out the 20th pair.

```
set.seed(7)
# Generating 500 random pairs using matrix
data = matrix(rnorm(1000), ncol = 2)
cat("20th pair:", data[20, 1], data[20, 2], "\n")
plot(data, main = "Scatter Plot of 500 Pairs", xlab = "X-axis",
ylab = "Y-axis")
```

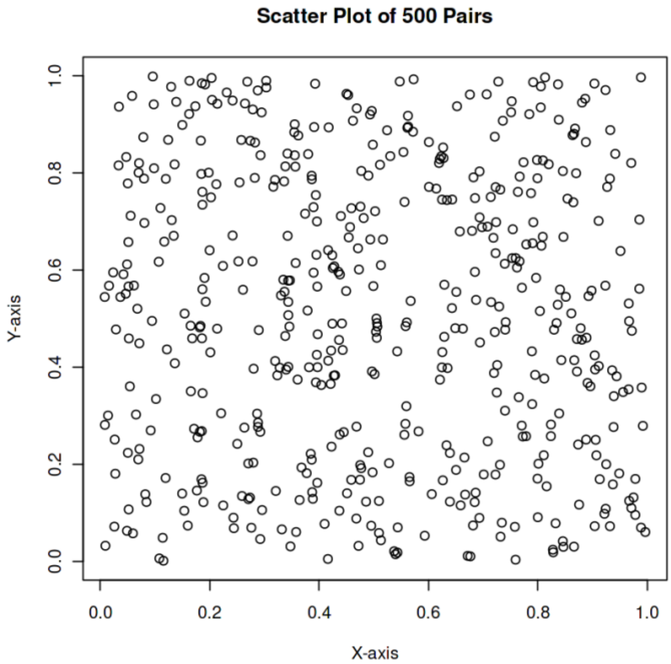
```
20th pair: 0.9881641 -2.176228
```



b. Generate a plot of 500 pairs using the runif command. Print out the 30th pair.

```
set.seed(7)
# Generating 500 randomn pairs using matrix
Data=matrix(runif(1000), ncol = 2)
cat("30th pair:", data[30, 1], data[30, 2], "\n")
plot(data, main = "Scatter Plot of 500 Pairs", xlab = "X-axis",
ylab = "Y-axis")
```

```
30th pair: 0.3622208 0.8772236
```



c. Explain how the two plots are different and explain why the two plots are different.

The two plots are different because I used two different commands “rnorm” and “runif”.
The big difference between those two commands is the numbers that they can produce.
“Rnorm” generates numbers from a normal distribution with a mean of 0 and a standard deviation of 1, while “runif” generates numbers from a uniform distribution between 0 and 1.
The first plot will show points concentrated around the mean, so the shape of the graph will be more circular, while the second plot show a uniform distribution of points with a minimum of 0 and a maximum of 1

2. Load the Life Expectancy data set.
a. Generate a list of all countries whose female lifespan is greater than 80. Your list should pair each country with its corresponding female lifespan.

```
lifexp = read.csv(file="LifeExpectancy.csv",head=TRUE)
selected_countries = lifexp[lifexp$FemaleLE > 80, c("Country",
"FemaleLE")]
print(selected_countries)
Country FemaleLE
1 Japan 86.8
2 Switzerland 85.3
3 Singapore 86.1
4 Australia 84.8
5 Spain 85.5
6 Iceland 84.1
7 Italy 84.8
8 Israel 84.3
9 Sweden 84.0
10 France 85.4
11 South Korea 85.5
12 Canada 84.1
13 Luxembourg 84.0
14 Netherlands 83.6
15 Norway 83.7
```

```
16 Malta 83.7
17 New Zealand 83.3
18 Austria 83.9
19 Ireland 83.4
20 United Kingdom 83.0
21 Belgium 83.5
22 Finland 83.8
23 Portugal 83.9
24 Germany 83.4
25 Greece 83.6
26 Slovenia 83.7
27 Denmark 82.5
28 Cyprus 82.7
29 Chile 83.4
30 Costa Rica 82.2
31 United States 81.6
32 Cuba 81.4
33 Czechia 81.7
34 Maldives 80.2
36 Croatia 81.2
37 Albania 80.7
38 Panama 81.1
40 Estonia 82.0
41 Poland 81.3
44 Uruguay 80.4
47 Slovakia 80.2
56 Vietnam 80.7
```

b. Generate a list of all countries whose male lifespan is greater than 75. Your list should pair each country with its corresponding male lifespan.

```
lifexp = read.csv(file="LifeExpectancy.csv",head=TRUE)
selected_countries=lifexp[lifexp$MaleLE > 75, c("Country",
"MaleLE")]
print(selected_countries)
```

```
Country MaleLE
1 Japan 80.5
2 Switzerland 81.3
3 Singapore 80.0
4 Australia 80.9
5 Spain 80.1
6 Iceland 81.2
7 Italy 80.5
8 Israel 80.6
9 Sweden 80.7
10 France 79.4
11 South Korea 78.8
12 Canada 80.2
13 Luxembourg 79.8
14 Netherlands 80.0
15 Norway 79.8
16 Malta 79.7
17 New Zealand 80.0
18 Austria 79.0
19 Ireland 79.4
20 United Kingdom 79.4
```

```
21 Belgium 78.6
22 Finland 78.3
23 Portugal 78.2
24 Germany 78.7
25 Greece 78.3
26 Slovenia 77.9
27 Denmark 78.6
28 Cyprus 78.3
29 Chile 77.4
30 Costa Rica 77.1
31 United States 76.9
32 Cuba 76.9
33 Czechia 75.9
34 Maldives 76.9
35 Qatar 77.4
37 Albania 75.1
39 Brunei 76.3
43 United Arab Emirates 76.4
45 Bahrain 76.2
```

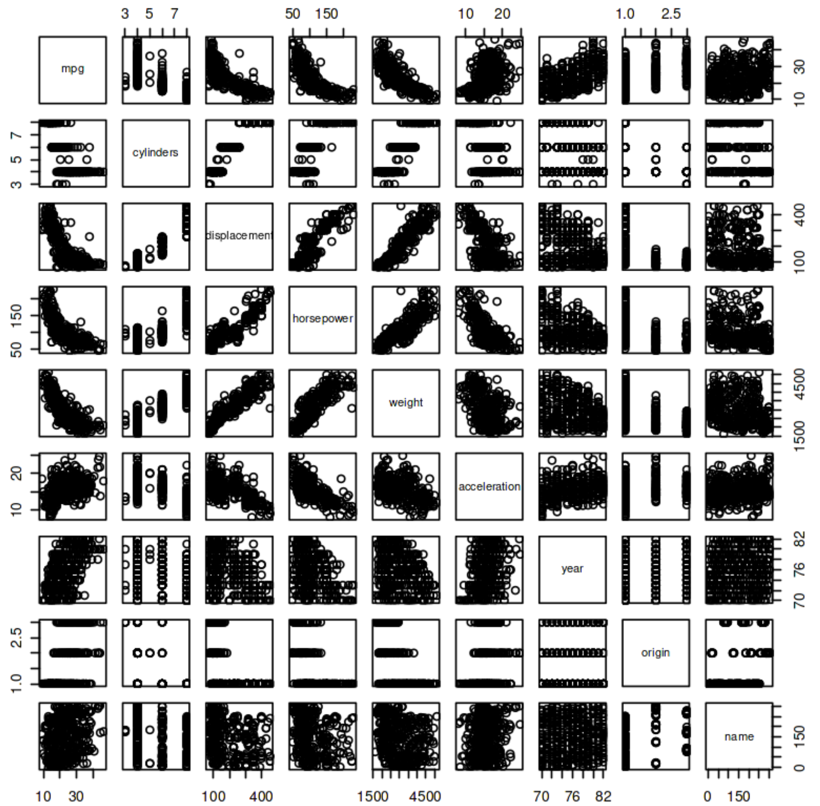
3. Load the Auto data set (start with the read.csv command at the bottom of page 49 and remove the rows that contain missing data as shown on page 50).

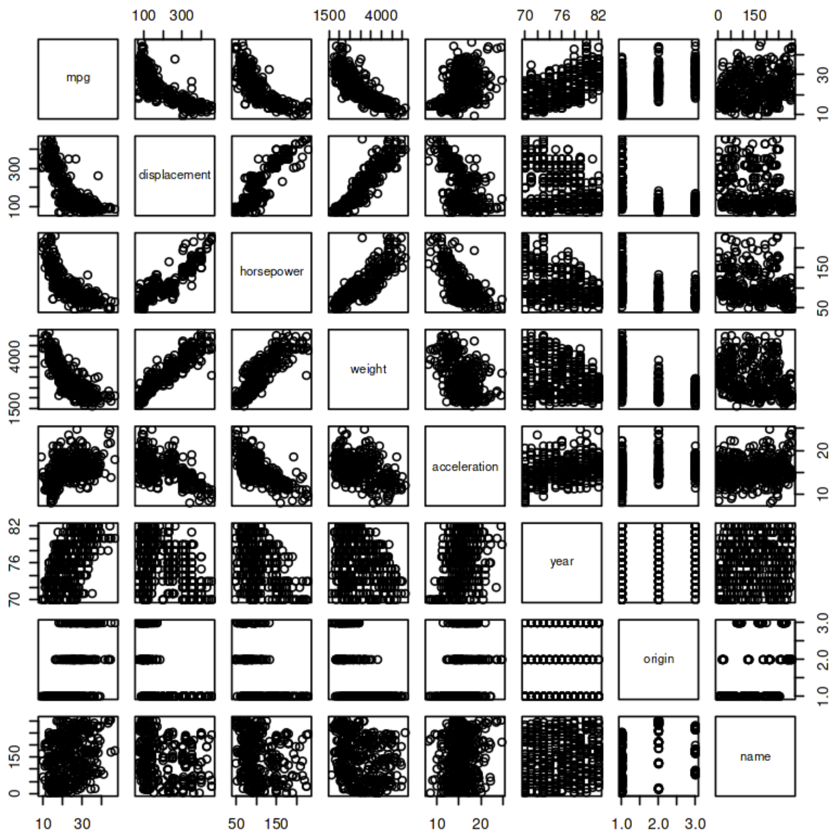
```
Using the command on page 49:
Auto = read.csv(file="Auto.csv",na.strings="?", stringsAsFactors = T)
#print(Auto)
#dim(Auto)
#summary(Auto)

Auto= na.omit(Auto) #remove the rows that have missing entries now we have 392
cars remaining
dim(Auto)
summary(Auto)
      mpg          cylinders      displacement     horsepower        weight    
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613  
 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225  
 Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804  
 Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978  
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615  
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140  
                                                                               
  acceleration        year           origin                      name    
 Min.   : 8.00   Min.   :70.00   Min.   :1.000   amc matador       :  5  
 1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   ford pinto        :  5  
 Median :15.50   Median :76.00   Median :1.000   toyota corolla    :  5  
 Mean   :15.54   Mean   :75.98   Mean   :1.577   amc gremlin       :  4  
 3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :  4  
 Max.   :24.80   Max.   :82.00   Max.   :3.000   chevrolet chevette:  4  
                                                 (Other)           :365  
```

a. Use the pairs function (p. 51) to look at pairwise plots of all the variables.

```
pairs(Auto)
pairs(
~ mpg + displacement + horsepower + weight + acceleration + year +
origin + name,
data = Auto
```

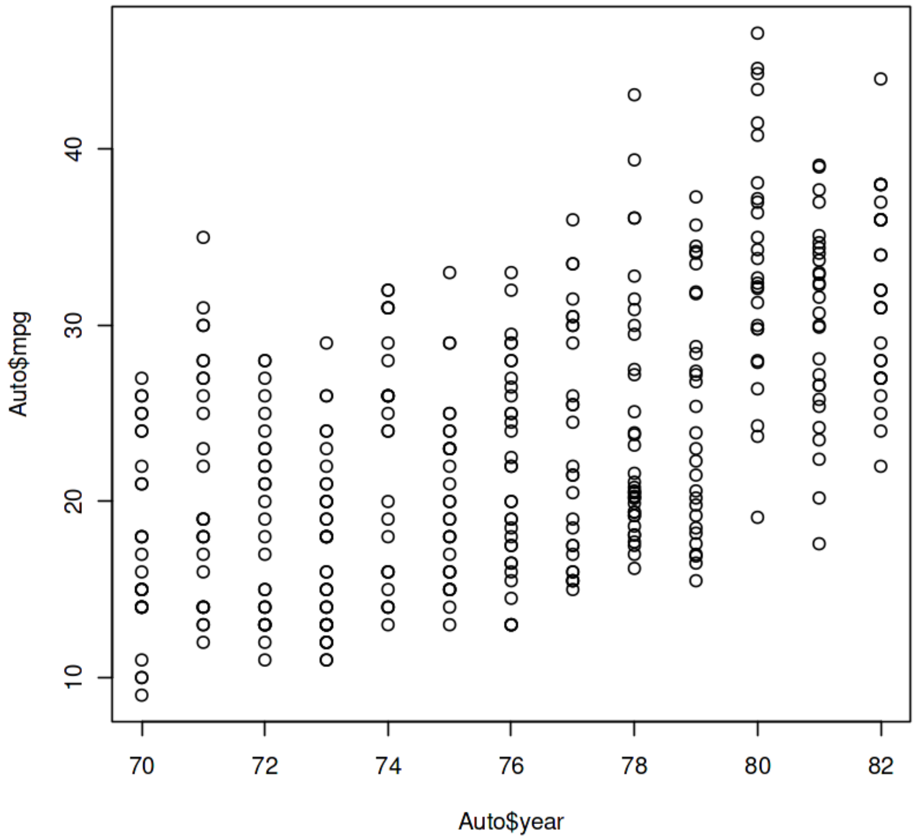
)





b. Make a single plot of year (horiz axis) vs mpg (vert. axis), does it appear that gas mileage improved over time? Justify your answer by referring to the plot.

```
plot( Auto$year, Auto$mpg )
```

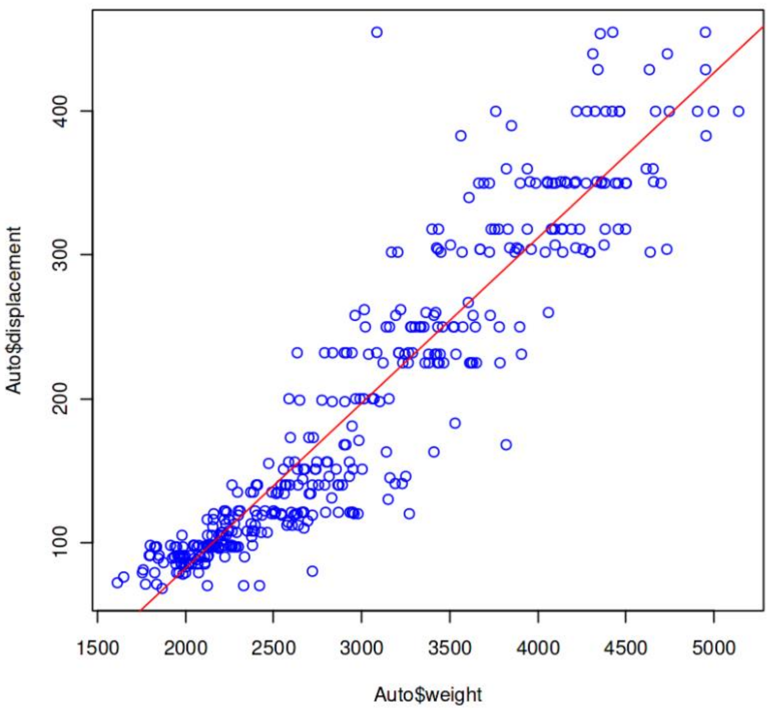


According to the graph, as we move on to year of production the average mpg for every year is increasing creating almost a linear regression line with a positive slope.

c. It looks like there is a linear relationship between displacement and weight. Plot weight (horiz axis) vs displacement, and use the lm function to find the least squares linear model that predicts displacement as a function of weight. In your report, state your linear model in the form d = mw + b where d and w are the variables and m and b are the slope and intercept respectively. Finally, use your model to predict the displacement of a car that weighs 4000 lbs. (You can make your prediction any way you like do it by hand, on your calculator, etc. If you like, you can try to use the predict function in R, but it takes a bit of setup.)

```
myline=lm(Auto$displacement ~ Auto$weight) #Using the lm
command to find the least squares line
summary(myline)
plot(Auto$weight,Auto$displacement,col="blue")
abline(myline,col="red")
```

```
Call:
lm(formula = Auto$displacement ~ Auto$weight)

Residuals:
     Min       1Q   Median       3Q      Max 
-123.241  -18.851   -0.316   16.776  248.126 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.478e+02  6.951e+00  -21.27   <2e-16 ***
Auto$weight  1.149e-01  2.245e-03   51.20   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 37.71 on 390 degrees of freedom
Multiple R-squared:  0.8705,    Adjusted R-squared:  0.8701 
F-statistic:  2621 on 1 and 390 DF,  p-value: < 2.2e-16
```

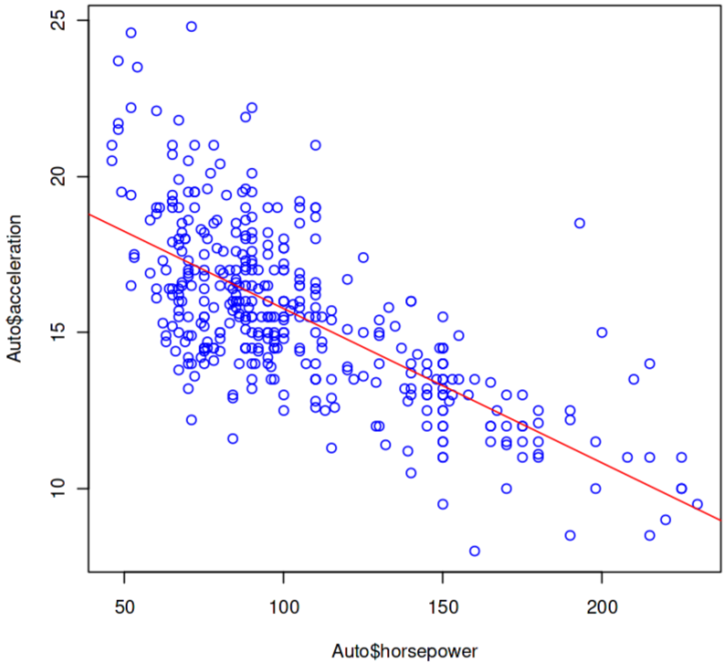


Line: d = 0.1149w -147.8
d= 0.1149(4000)-147.8
If a car would weigh 4000 lbs the displacement would be equal to 311.8 m

d. Repeat part c. with horsepower and acceleration. Plot horsepower (horiz. axis) vs acceleration (vert. axis), find the least squares linear model that predicts acceleration as a function of horsepower, and plot your model along with the data. State your model in the form a = mh + b, and use your model to predict the acceleration of a car with 425 horsepower.

```
myline=lm(Auto$acceleration ~ Auto$horsepower) #Using the lm
command to find the least squares line
summary(myline)
plot(Auto$horsepower,Auto$acceleration,col="blue")
abline(myline,col="red")
```

```
Call:
lm(formula = Auto$acceleration ~ Auto$horsepower)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.9947 -1.2913 -0.1748  1.1229  7.6053 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)     20.70193    0.29274   70.72   <2e-16 ***
Auto$horsepower -0.04940    0.00263  -18.78   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.002 on 390 degrees of freedom
Multiple R-squared:  0.475,	Adjusted R-squared:  0.4736 
F-statistic: 352.8 on 1 and 390 DF,  p-value: < 2.2e-16
```



Line: a = -0.04940h + 20.70193
A= -0.04940(425) +20.70193
If a car has horsepower equal to 425 the resultant acceleration will be equal to -0.29307 m/s^2