Assignment 1

The Old Republic

2023-06-21

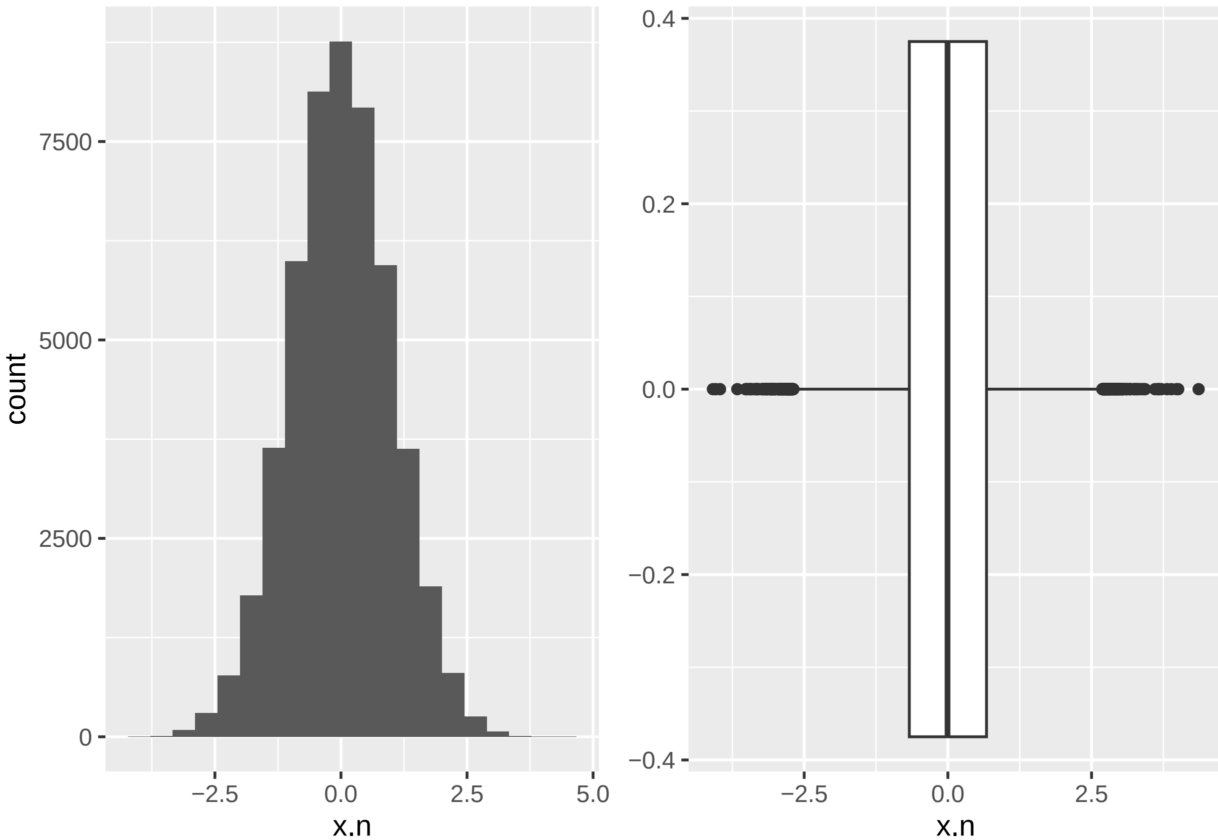
```r
library(Pareto)
set.seed(100)
Data = data.frame(x.n = rnorm(50000), x.p = rPareto(50000, t=1, alpha=2))
summary(Data)
```

```
##       x.n                 x.p
##  Min.   :-4.087893   Min.   :  1.000
##  1st Qu.:-0.671144   1st Qu.:  1.154
##  Median :-0.005919   Median :  1.412
##  Mean   :-0.000208   Mean   :  1.994
##  3rd Qu.: 0.672466   3rd Qu.:  1.992
##  Max.   : 4.363243   Max.   :159.275
```

Question 1

1. Histogram and Boxplot of the Variable x.n

```r
library(ggplot2)
library(grid)
library(gridExtra)
hist = ggplot(Data, aes(x= x.n)) + geom_histogram(bins = 20)
box = ggplot(Data, aes(x= x.n)) + geom_boxplot()
grid.arrange(hist, box, ncol = 2)
```



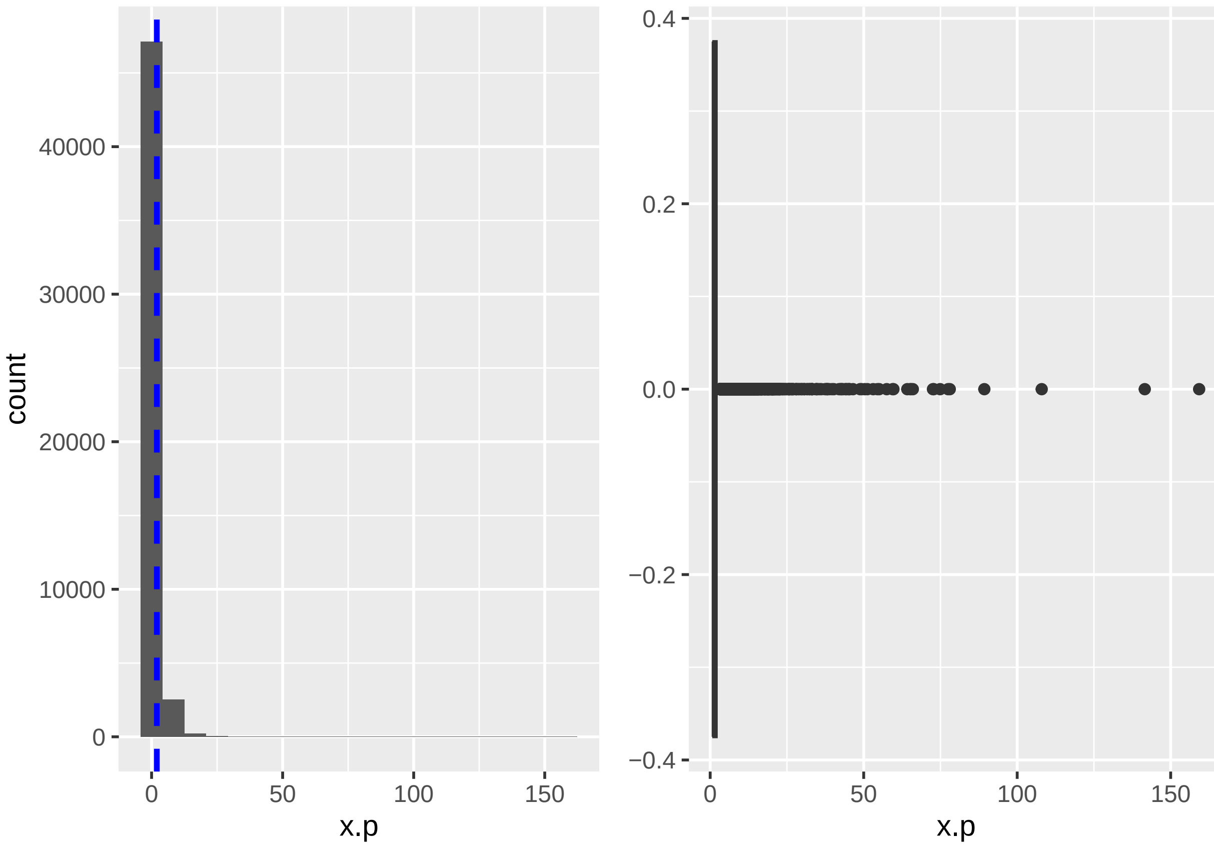
2.

The sample mean, and standard deviation of x.n are $-2.0849558 \times 10^{-4}$ and 0.9989658 respectively, we see that these parameters are approximately the same as the standard normal distribution. In fact, as we increase the sample size to infinity, the mean and standard deviation will also approach 0 and 1.

3. The sample mean and sample standard deviation are close to the distribution parameters. The sample mean can be used as a predictor for new observations. Including the fact that there are not many outliers, the sample mean is a reasonable indicator to predict new observations. Further, the sample standard deviation of course describes the spread of the data.

4. let's plot the median along with the mean and say there's a difference. And we can also mention the high proportion of outliers. We can add a pie chart maybe cause they seem to like visuals

```
phist = ggplot(Data, aes(x= x.p)) + geom_histogram(bins = 20)+
  geom_vline(aes(xintercept=mean(x.p)),
            color="blue", linetype="dashed", size=1)

pbox = ggplot(Data, aes(x= x.p)) + geom_boxplot()

grid.arrange(phist, pbox, ncol = 2)
```




```
mean(Data$x.p)
```

```
## [1] 1.993904
```

```
sd(Data$x.p)
```

```
## [1] 2.601173
```

```
quantile(Data$x.p, prob=c(.25,.5,.75), type=1)
```

```
##      25%      50%      75%
## 1.154338 1.411988 1.991697
```

```
q1  <- quantile(Data$x.p, prob=.25)
q3  <- quantile(Data$x.p, prob=.75)
IQR <- q3 - q1

l <- q1 - IQR
u <- q3 + IQR

upper_outliers <- filter(Data, Data$x.p > u)
lower_outliers <- filter(Data, Data$x.p < l)
```

Question 2

1.

```
library("ggplot2")
library("tidyr")
```

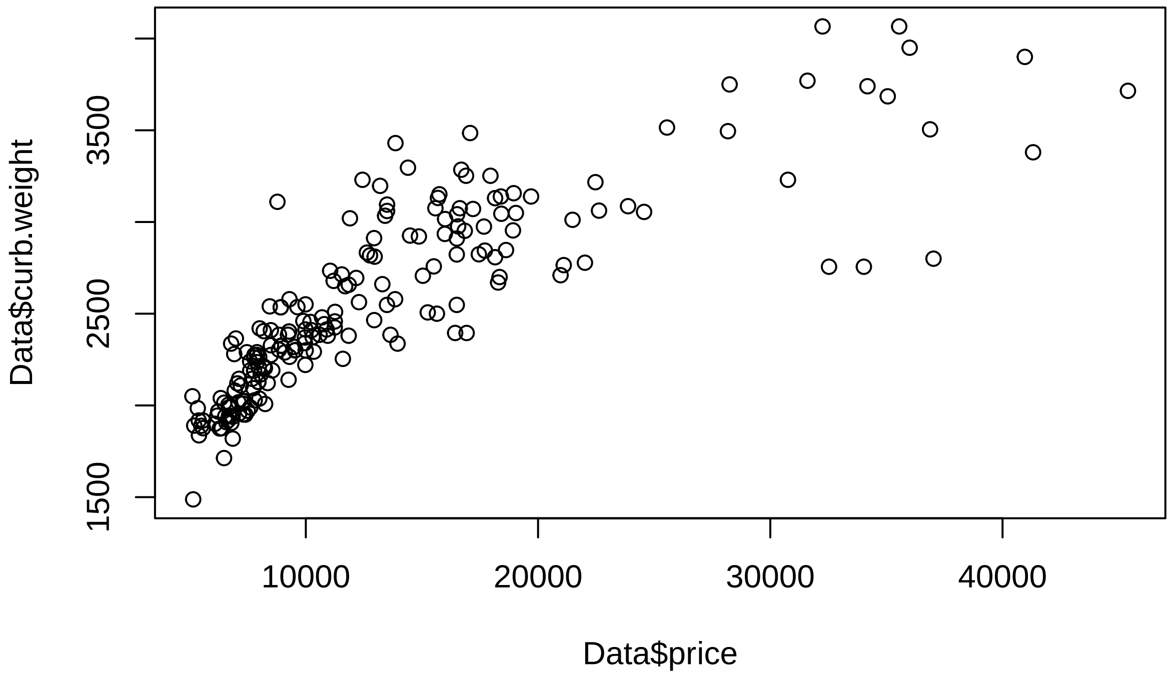
```
Data = read.csv("Car_data.csv", na.strings=c("?"))

Data = Data %>%
  drop_na(c("price"))
```

2.

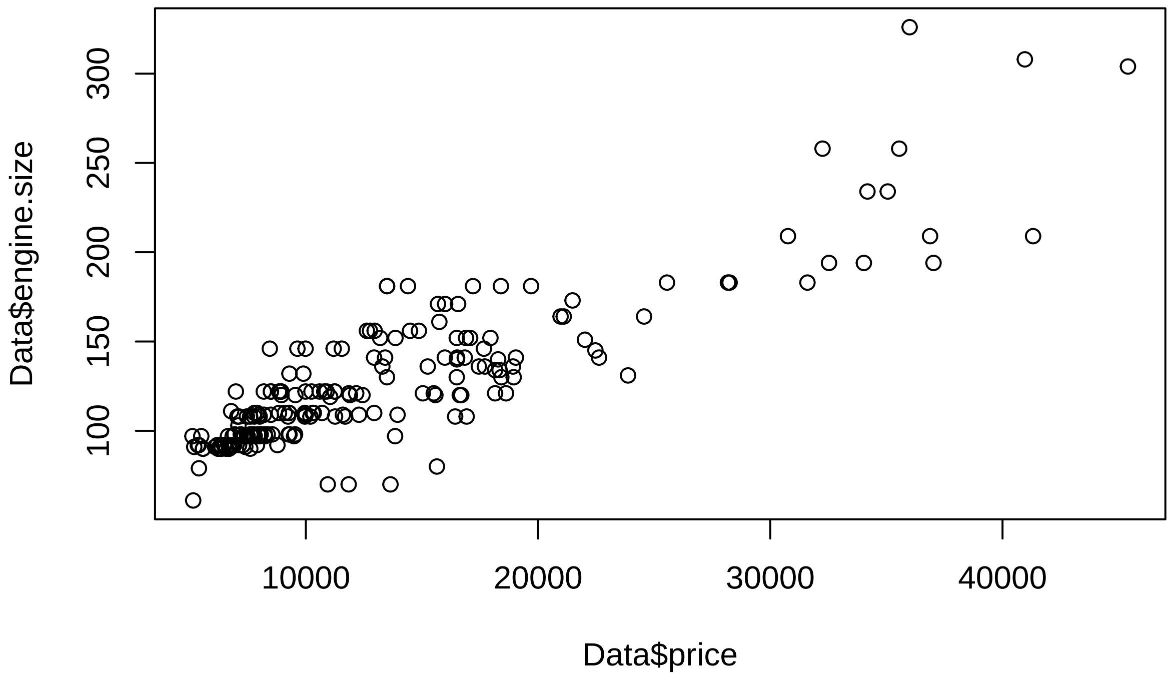
```
hist = ggplot(Data, aes(x= price)) + geom_histogram(bins = 20)
```

3.

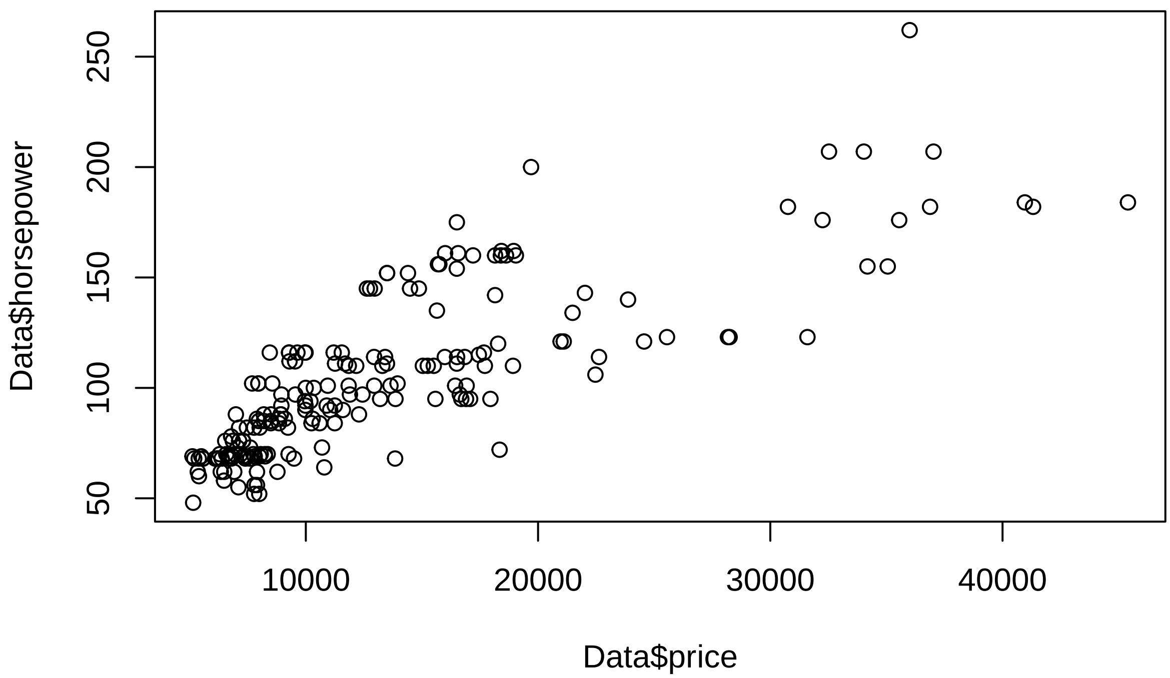
```
plot(Data$price, Data$curb.weight)
```



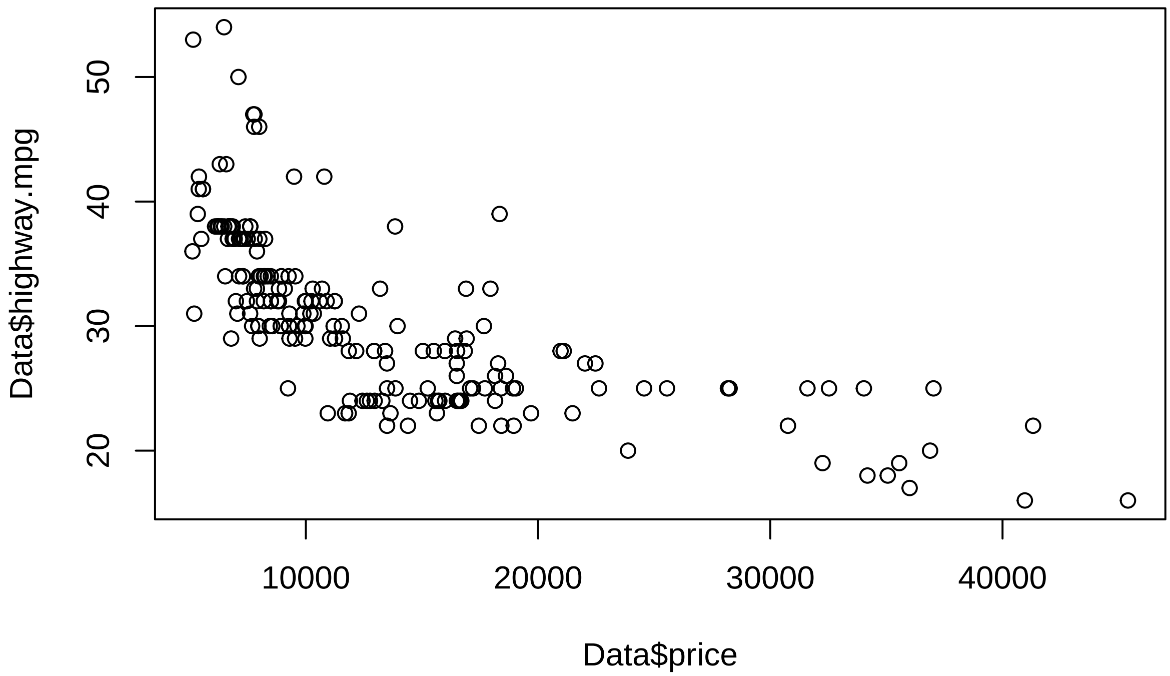

```
plot(Data$price, Data$engine.size)
```




```
plot(Data$price, Data$horsepower)
```




```
plot(Data$price, Data$highway.mpg)
```



We can see from these graphs that price is directly related to curb weight, engine size, and horsepower. However, it has an inverse relationship with highway mileage.

4.

```
#read data
Data = read.csv("Car_data.csv", na.strings=c("?"))

#take out rows of NA
Data = Data %>%
  drop_na(c("horsepower"))

#normalize the data
x1 = (Data$curb.weight - mean(Data$curb.weight)) / sd(Data$curb.weight)
x2 = (Data$engine.size - mean(Data$engine.size)) / sd(Data$engine.size)
x3 = (Data$horsepower - mean(Data$horsepower)) / sd(Data$horsepower)
x4 = (Data$highway.mpg - mean(Data$highway.mpg)) / sd(Data$highway.mpg)

#create the dataframe
X = data.frame(c_weight = x1, engine = x2, horsepower = x3, mileage = x4)

#perform the analysis
prcomp(X)
```

```
## Standard deviations (1, .., p=4):
## [1] 1.8249029 0.5716967 0.5024550 0.3007179
##
## Rotation (n x k) = (4 x 4):
##                   PC1         PC2         PC3        PC4
## c_weight    0.5109969 -0.09039592 -0.61687136 -0.5917605
## engine      0.5020525 -0.62485679 -0.06293488  0.5945893
## horsepower  0.5002726 -0.03380273  0.77951083 -0.3754298
## mileage    -0.4863669 -0.77475153  0.08872308 -0.3941262
```

Principle component 1 is influenced equally by all variables except mileage. More specifically, it carries a similar magnitude but with the opposite sign. It can be noted that cars that are heavier and cars with powerful engines (large engine size and high horsepower) often do not have as high of a mileage. Thus the first principle component corresponds with the efficiency of the car–with heavier, stronger cars on one end; and lighter, more fuel efficient cars on the other

The second principle component is mostly influenced by mileage and engine size. Diesel engines, compared to petrol, are often larger and are also more fuel efficient, which would result in a higher mileage. Thus, the second principle component could correspond with the engine type of the car, such as possibly separating diesel and petrol engines.

The third principle component is weakly influenced by engine size, mostly being affected by curb weight and horsepower. These variables have different signs however, so this principle component largely describes the difference between these two. Thus, the third principle component may describe the difference between large, heavier family cars and more compact, lighter sports cars which have high horsepower

5.

```
Data = read.csv("Car_data.csv", na.strings=c("?"))
Data = Data %>%
  drop_na(c("horsepower"))
Data = Data %>%
  drop_na(c("price"))


#normalize the data
x1 = (Data$curb.weight - mean(Data$curb.weight)) / sd(Data$curb.weight)
x2 = (Data$engine.size - mean(Data$engine.size)) / sd(Data$engine.size)
x3 = (Data$horsepower - mean(Data$horsepower)) / sd(Data$horsepower)
x4 = (Data$highway.mpg - mean(Data$highway.mpg)) / sd(Data$highway.mpg)

#create the dataframe
X = data.frame(c_weight = x1, engine = x2, horsepower = x3, mileage = x4)
X
```

```
##         c_weight      engine  horsepower     mileage
## 1   -0.015446889  0.07606363  0.20245638 -0.53777142
## 2   -0.015446889  0.07606363  0.20245638 -0.53777142
## 3    0.513546219  0.60297274  1.34747902 -0.68376939
## 4   -0.421328873 -0.42689507 -0.03719952 -0.09977751
## 5    0.515469830  0.21976611  0.30897011 -1.26776128
## 6   -0.094314952  0.21976611  0.17582795 -0.82976737
## 7    0.553942056  0.21976611  0.17582795 -0.82976737
## 8    0.765539300  0.21976611  0.17582795 -0.82976737
## 9    1.019455991  0.10001404  0.97468095 -1.55975722
## 10  -0.309759418 -0.45084548 -0.06382795 -0.24577548
## 11  -0.309759418 -0.45084548 -0.06382795 -0.24577548
## 12   0.296178142  0.89037771  0.46874071 -0.39177345
## 13   0.401976764  0.89037771  0.46874071 -0.39177345
## 14   0.959824041  0.89037771  0.46874071 -0.82976737
## 15   1.296456019  1.96814634  2.09307515 -1.26776128
## 16   1.584997714  1.96814634  2.09307515 -1.26776128
## 17   1.825449126  1.96814634  2.09307515 -1.55975722
## 18  -2.054474868 -1.57651494 -1.47513492  3.25817583
## 19  -1.311960906 -0.88195294 -0.88930939  1.79819612
## 20  -1.244634510 -0.88195294 -0.88930939  1.79819612
## 21  -1.308113683 -0.88195294 -0.94256626  1.50620018
```

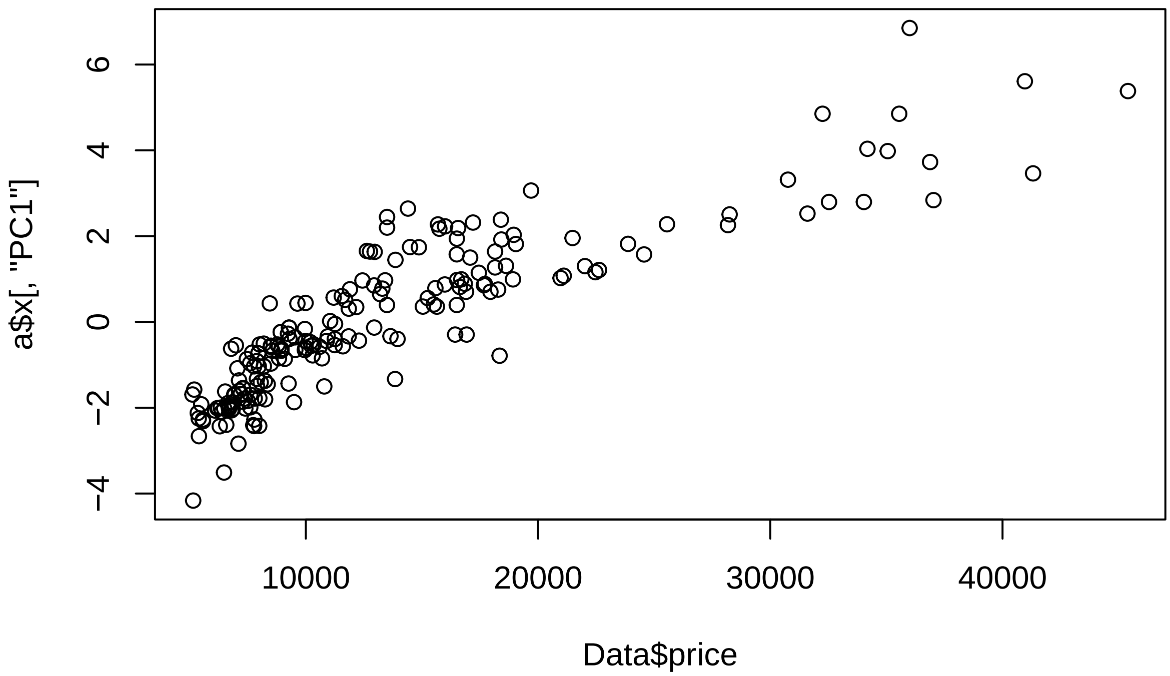
```
## 22  -1.308113683 -0.88195294 -0.94256626  1.06820626
## 23  -0.823363635 -0.69034962 -0.03719952 -0.09977751
## 24  -1.133065055 -0.88195294 -0.94256626  1.06820626
## 25  -1.090745606 -0.88195294 -0.94256626  1.06820626
## 26  -1.090745606 -0.88195294 -0.94256626  1.06820626
## 27  -0.702176123 -0.69034962 -0.03719952 -0.09977751
## 28  -0.040453836 -0.11553969 -0.40999759 -0.09977751
## 29   0.490462883  0.69877439  1.10782312 -0.97576534
## 30  -1.621662325 -0.83405211 -1.20885059  3.40417381
## 31  -1.417759527 -0.83405211 -0.72953879  1.06820626
## 32  -1.383134524 -1.14540749 -1.15559372  1.65219815
## 33  -1.185002560 -0.83405211 -0.72953879  0.48421438
## 34  -1.154224779 -0.83405211 -0.72953879  0.48421438
## 35  -1.050349769 -0.83405211 -0.72953879  0.48421438
## 36  -1.023419211 -0.83405211 -0.72953879  0.48421438
## 37  -0.615613615 -0.40294465 -0.46325445  0.33821641
## 38  -0.513662216 -0.40294465 -0.46325445  0.33821641
## 39  -0.484808046 -0.40294465 -0.46325445  0.33821641
## 40  -0.354002478 -0.40294465 -0.46325445  0.33821641
## 41  -0.175106627 -0.40294465 -0.06382795 -0.39177345
## 42  -0.505967771 -0.40294465 -0.09045639  0.04622046
## 43  -0.421328873 -0.37899424 -0.67628192 -0.24577548
## 44   0.342344813 -0.18739093 -0.35674072 -0.24577548
## 45   2.904595066  3.14171663  1.93330455 -1.70575519
## 46   2.904595066  3.14171663  1.93330455 -1.70575519
## 47   2.681456155  4.77034479  4.22334983 -1.99775114
## 48  -1.281183125 -0.85800252 -0.94256626  0.04622046
## 49  -1.261947012 -0.85800252 -0.94256626  1.06820626
## 50  -1.252328955 -0.85800252 -0.94256626  1.06820626
## 51  -1.175384503 -0.85800252 -0.94256626  1.06820626
## 52  -1.165766447 -0.85800252 -0.94256626  1.06820626
## 53  -0.338613587 -1.36096122 -0.06382795 -1.12176331
## 54  -0.338613587 -1.36096122 -0.06382795 -1.12176331
## 55  -0.328995531 -1.36096122 -0.06382795 -1.12176331
## 56  -0.107780231 -1.12145708  0.84153878 -1.12176331
## 57  -0.328995531 -0.11553969 -0.51651132  0.19221843
## 58  -0.280905248 -0.11553969 -0.51651132  0.19221843
## 59  -0.328995531 -0.11553969 -0.51651132  0.19221843
## 60  -0.280905248 -0.11553969 -0.51651132  0.19221843
## 61  -0.217426075 -0.11553969 -1.04907999  1.65219815
## 62  -0.252051079 -0.11553969 -0.51651132  0.19221843
## 63   0.219233690  0.31556777  0.44211228 -0.53777142
## 64   0.276942029  0.17186528 -0.83605252  1.21420423
## 65   1.844685239  1.34543557  0.52199758 -0.82976737
## 66   2.296733895  1.34543557  0.52199758 -0.82976737
## 67   1.806213013  1.34543557  0.52199758 -0.82976737
## 68   2.335206121  1.34543557  0.52199758 -0.82976737
## 69   2.277497782  2.56690669  1.37410745 -1.85175317
## 70   2.171699161  2.56690669  1.37410745 -1.85175317
## 71   2.585275590  4.33923733  2.14633202 -2.14374911
## 72   2.229407500  4.24343568  2.14633202 -2.14374911
## 73   0.680900402  0.31556777  1.90667612 -0.97576534
## 74  -1.227322008 -0.83405211 -0.94256626  1.50620018
## 75  -1.177308115 -0.83405211 -0.94256626  1.06820626
```

```
## 76  -1.061891437 -0.83405211 -0.94256626  1.06820626
## 77  -0.790662243 -0.69034962 -0.03719952 -0.09977751
## 78  -0.357849700 -0.40294465  0.33559855 -0.09977751
## 79  -0.438641375 -0.11553969 -0.40999759  0.19221843
## 80   0.532782332  0.69877439  1.10782312 -0.97576534
## 81   0.702060127  0.69877439  1.10782312 -0.97576534
## 82   0.711678183  0.69877439  1.10782312 -0.97576534
## 83  -0.367467757 -0.11553969 -0.40999759  0.19221843
## 84  -0.290523305 -0.11553969 -0.40999759  0.19221843
## 85  -0.294370527 -0.40294465  0.33559855 -0.09977751
## 86  -0.294370527 -0.40294465  0.33559855 -0.09977751
## 87  -1.283106736 -0.71430004 -0.91593782  0.92220829
## 88  -1.036884490 -0.57059755 -1.28873589  2.82018192
## 89  -1.227322008 -0.71430004 -0.91593782  0.92220829
## 90  -1.188849782 -0.71430004 -0.91593782  0.92220829
## 91  -1.023419211 -0.71430004 -0.91593782  0.92220829
## 92  -1.163842835 -0.71430004 -0.91593782  0.92220829
## 93  -1.015724765 -0.71430004 -0.91593782  0.92220829
## 94  -1.125370609 -0.71430004 -0.91593782  0.92220829
## 95  -0.998412264 -0.71430004 -0.91593782  0.92220829
## 96  -1.054196991 -0.71430004 -0.91593782  0.92220829
## 97  -0.446335820 -0.16344051 -0.17034169  0.48421438
## 98  -0.488655269 -0.16344051 -0.17034169  0.48421438
## 99   1.036768493  1.29753474  1.29422215 -1.26776128
## 100  1.423414364  1.29753474  1.29422215 -1.26776128
## 101  0.969442097  1.29753474  1.29422215 -0.82976737
## 102  0.990601822  1.29753474  1.50724962 -0.82976737
## 103  1.121407390  1.29753474  2.57238695 -1.12176331
## 104  1.121407390  1.29753474  1.50724962 -0.82976737
## 105  0.892497645 -0.16344051 -0.17034169 -0.97576534
## 106  1.232976846  0.60297274 -0.22359855  0.33821641
## 107  1.296456019 -0.16344051 -0.17034169 -0.97576534
## 108  1.681178279  0.60297274 -0.22359855 -0.82976737
## 109  0.998296267 -0.16344051 -0.22359855 -0.97576534
## 110  1.338775467  0.60297274 -0.22359855  0.33821641
## 111  1.402254640 -0.16344051 -0.22359855 -0.97576534
## 112  1.786976900  0.60297274 -0.22359855 -0.82976737
## 113  0.998296267 -0.16344051 -0.17034169 -0.97576534
## 114  1.338775467  0.60297274 -0.22359855  0.33821641
## 115  1.104094888  0.17186528  1.02793782 -0.97576534
## 116 -1.227322008 -0.88195294 -0.94256626  1.50620018
## 117 -0.823363635 -0.69034962 -0.03719952 -0.09977751
## 118 -1.133065055 -0.88195294 -0.94256626  1.06820626
## 119 -1.090745606 -0.88195294 -0.94256626  1.06820626
## 120 -0.702176123 -0.69034962 -0.94256626  1.06820626
## 121 -0.040453836 -0.11553969 -0.40999759 -0.09977751
## 122  0.503928163  0.69877439  1.10782312 -0.97576534
## 123  0.426983711  0.57902232  1.05456625 -0.53777142
## 124  0.384664262  1.60889013  2.75878599 -0.82976737
## 125  0.384664262  1.60889013  2.75878599 -0.82976737
## 126  0.469303159  1.60889013  2.75878599 -0.82976737
## 127  0.196150354 -0.13949010  0.17582795 -0.39177345
## 128  0.267323973 -0.13949010  0.17582795 -0.39177345
## 129  0.290407308 -0.13949010  0.17582795 -0.39177345
```

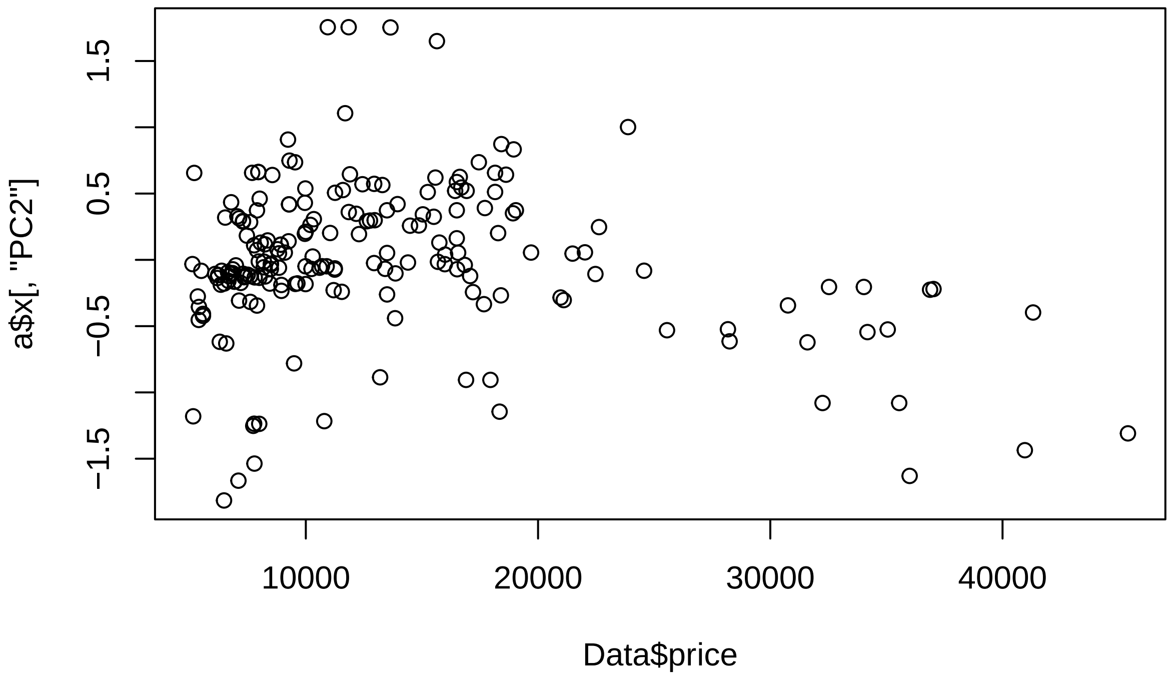
```
## 130  0.388511484 -0.13949010  0.17582795 -0.39177345
## 131  0.484692050 -0.13949010  1.50724962 -0.68376939
## 132  0.559712890 -0.13949010  1.50724962 -0.68376939
## 133 -0.973405317 -0.71430004 -0.91593782  0.77621032
## 134 -0.838752526 -0.45084548 -0.80942409  0.04622046
## 135 -0.607919169 -0.45084548 -0.80942409  0.04622046
## 136 -0.790662243 -0.45084548 -0.56976819  0.92220829
## 137 -0.704099735 -0.45084548 -0.56976819  0.33821641
## 138 -0.415558039 -0.45084548 -0.25022699  0.19221843
## 139 -0.328995531 -0.45084548 -0.56976819 -0.82976737
## 140 -0.088544118 -0.45084548  0.20245638 -0.24577548
## 141 -0.511738604 -0.45084548 -0.56976819  0.19221843
## 142 -0.194342740 -0.45084548 -0.25022699  0.04622046
## 143 -0.261669135 -0.45084548 -0.56976819 -0.24577548
## 144  0.180761464 -0.45084548  0.20245638 -1.12176331
## 145 -1.098440051 -0.83405211 -1.10233686  1.21420423
## 146 -0.992641430 -0.83405211 -1.10233686  1.06820626
## 147 -1.040731712 -0.83405211 -1.10233686  1.06820626
## 148 -0.530974717 -0.83405211 -1.10233686  0.92220829
## 149 -0.511738604 -0.83405211 -1.10233686  0.19221843
## 150  1.065622662 -0.83405211 -1.10233686  0.19221843
## 151 -0.913773366 -0.69034962 -0.88930939  0.92220829
## 152 -0.859912250 -0.69034962 -0.88930939  0.92220829
## 153 -0.540592774 -0.40294465 -1.26210746  0.77621032
## 154 -0.540592774 -0.40294465 -1.26210746  2.38218801
## 155 -0.888766419 -0.69034962 -0.88930939  2.38218801
## 156 -0.834905303 -0.69034962 -0.88930939  0.48421438
## 157 -0.800280300 -0.69034962 -0.88930939  0.48421438
## 158 -0.744495572 -0.69034962 -0.88930939  0.48421438
## 159 -0.677169176 -0.69034962 -0.88930939  0.48421438
## 160 -0.559828887 -0.69034962  0.22908481 -0.24577548
## 161 -0.492502491 -0.69034962  0.22908481 -0.24577548
## 162 -0.030835779  0.45927025  0.33559855 -0.09977751
## 163 -0.038530224  0.45927025  0.33559855 -0.09977751
## 164 -0.009676055  0.45927025  0.33559855 -0.09977751
## 165  0.236546192  0.45927025  0.33559855 -0.09977751
## 166  0.303872587  0.45927025  0.33559855 -0.09977751
## 167  0.805935137  0.45927025  0.33559855 -0.09977751
## 168 -0.442488598 -0.11553969 -0.30348385  0.48421438
## 169 -0.146252457 -0.40294465 -0.80942409  0.33821641
## 170 -0.273210803 -0.11553969 -0.30348385  0.19221843
## 171 -0.273210803 -0.11553969 -0.30348385  0.19221843
## 172 -0.188571906 -0.11553969 -0.30348385  0.19221843
## 173  0.807858748  1.05803060  1.53387805 -0.97576534
## 174  0.884803200  1.05803060  1.53387805 -0.97576534
## 175  1.106018500  1.05803060  1.40073588 -0.97576534
## 176  1.144490726  0.81852646  1.40073588 -0.97576534
## 177 -0.567523332 -0.71430004 -1.36862119  2.23619003
## 178 -0.667551120 -0.42689507 -0.48988289  0.48421438
## 179 -0.561752498 -0.71430004 -1.36862119  2.23619003
## 180 -0.661780286 -0.42689507 -0.48988289  0.48421438
## 181 -0.540592774 -0.42689507 -0.48988289  0.48421438
## 182 -0.455953877 -0.71430004 -0.94256626  1.65219815
## 183 -0.492502491 -0.42689507 -0.09045639  0.19221843
```

```
## 184 -0.580988611 -0.42689507 -0.35674072 -0.24577548
## 185 -0.644467784 -0.42689507 -0.35674072 -0.24577548
## 186  0.201921188  0.21976611  0.17582795 -0.97576534
## 187  0.044185062 -0.71430004 -0.94256626  1.06820626
## 188  0.013407281 -0.42689507 -0.40999759  0.04622046
## 189  0.684747625  0.33951818  0.28234168 -0.39177345
## 190  0.919428204  0.33951818  0.28234168 -0.39177345
## 191  0.728990685  0.33951818  0.28234168 -0.39177345
## 192  0.934817094  0.33951818  0.28234168 -0.39177345
## 193  0.940587928  0.07606363  1.56050648 -1.26776128
## 194  1.156032394  0.07606363  1.56050648 -1.26776128
## 195  0.761692077  0.33951818  0.28234168 -0.39177345
## 196  0.948282373  0.33951818  1.50724962 -0.82976737
## 197  0.877108755  1.10593143  0.81491035 -1.12176331
## 198  1.271449072  0.43531984  0.06931421 -0.53777142
## 199  0.973289320  0.33951818  0.28234168 -0.82976737
```

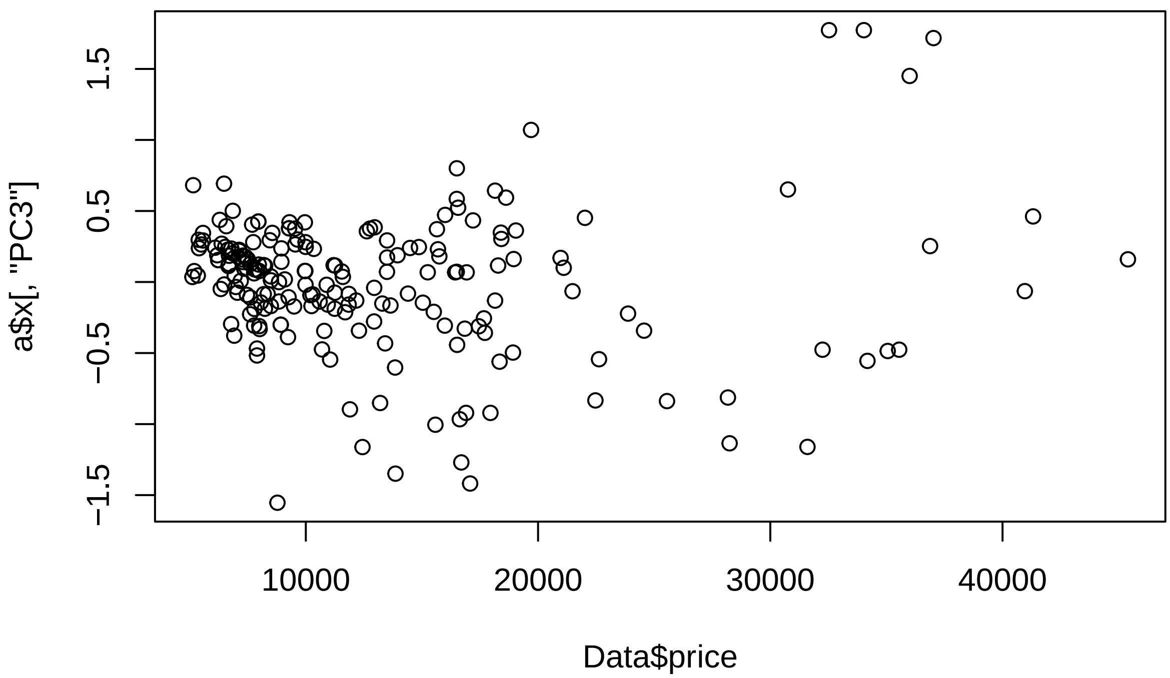
```r
a = prcomp(X)

plot(Data$price, a$x[, "PC1"])
```



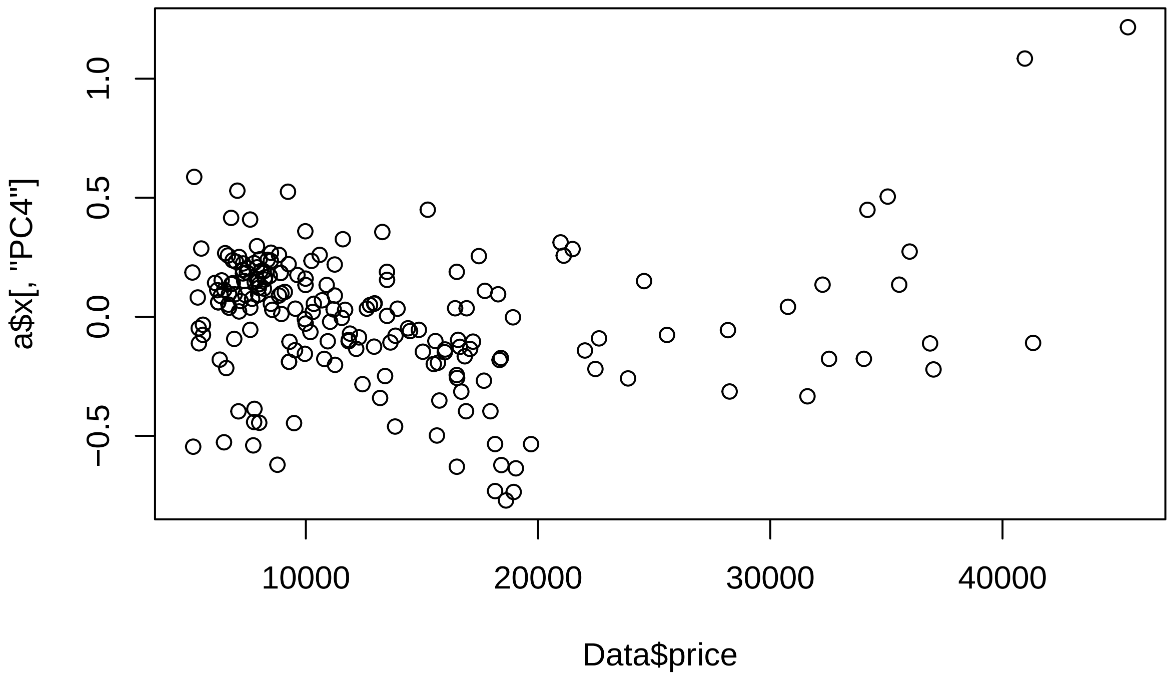

```r
plot(Data$price, a$x[, "PC2"])
```




```r
plot(Data$price, a$x[, "PC3"])
```




```r
plot(Data$price, a$x[, "PC4"])
```



as seen in the first graph, the variable price has a strong direct relationship with the first principle component. This is consistent with our findings in part (2.3) as we found price to have a directly positive relationship with curb weight, engine size, and horsepower; but an inverse relationship with highway mileage. This makes sense as we found our first principle component to have an equally significant influence from all variables, and only highway mileage with a negative value. Thus this graph further describes the variable price's relationship with these other variables.

The following graphs also are consistent with this information