DSCC/CSC/TCS 462 Statistics Assignment 1

Aradhya Mathur

2022-09-19

```
library(readr)
library(ggplot2)
library(moments)
```

Question 1) For the first part of this assignment, we will explore the relationships between variables using the same "car_sales.csv" dataset as HW0. In particular, we will explore the relationships between multiple variables.

```
#reading data
data1 <- read_csv("car_sales.csv")
```

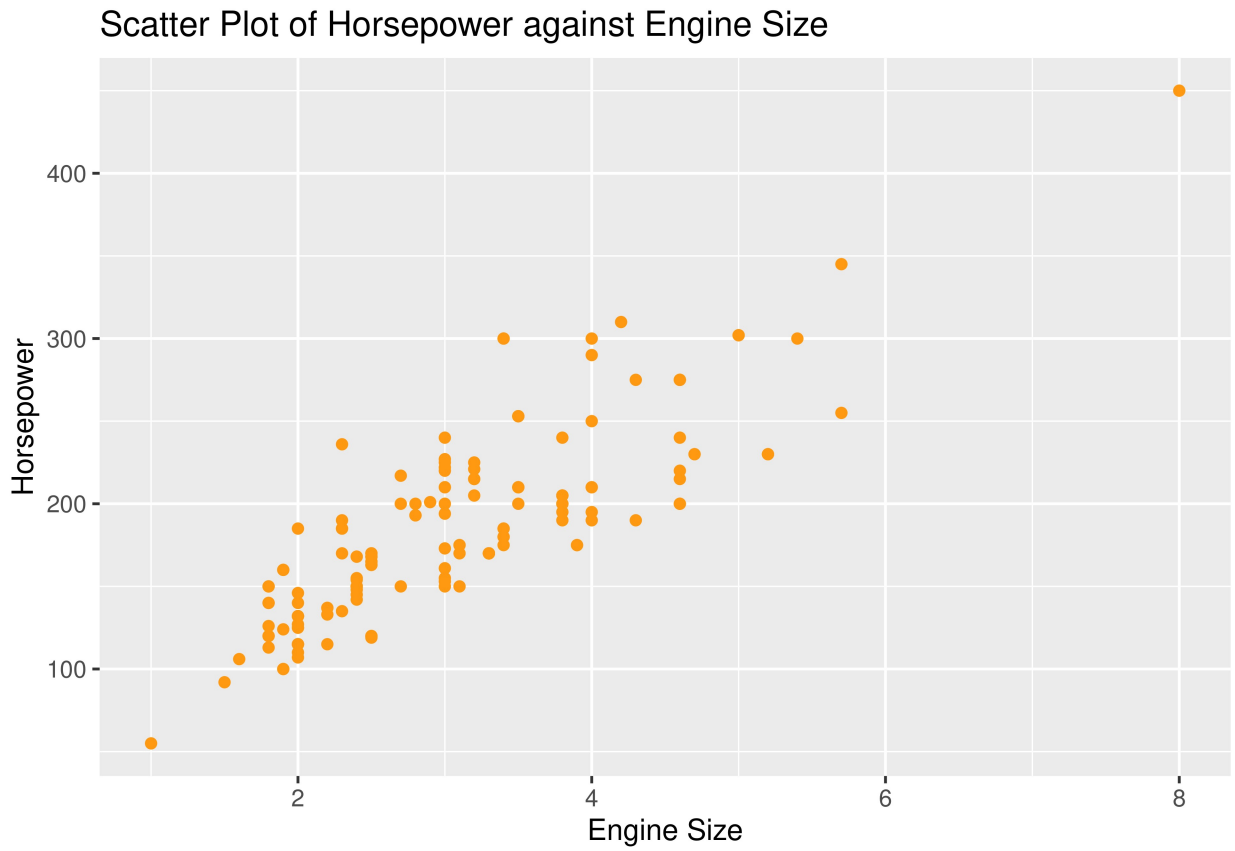
```
## Rows: 152 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): Manufacturer, Model
## dbl (9): price, Engine_size, Horsepower, Wheelbase, Width, Length, Curb_weig...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
data2 <- data1[c('Horsepower','Engine_size')]
data2
```

```
## # A tibble: 152 x 2
##    Horsepower Engine_size
##         <dbl>       <dbl>
##  1         55         1
##  2         92         1.5
##  3        113         1.8
##  4        100         1.9
##  5        106         1.6
##  6        100         1.9
##  7        120         1.8
##  8        120         1.8
##  9        140         1.8
## 10        124         1.9
## # ... with 142 more rows
```

a. Plot horsepower (y axis) against engine size (x axis). Make sure to label your axes. Comment on the form, strength, and direction of the plot. Note if there are any potential outliers.

```
#Scatter Plot of Horsepower against Engine Size
ggplot(data=data2, mapping = aes(x = Engine_size, y = Horsepower)) +
  geom_point(color='#ff9911') +
  ggtitle("Scatter Plot of Horsepower against Engine Size") +
  xlab("Engine Size") + ylab("Horsepower")
```



Strongly correlated, positive and linear association between horsepower and engine size. Yes, there are few outliers for ex - (8,450)

b. Calculate the correlation between horsepower and engine size.
Comment on this value in relation to your scatterplot

```
#Pearson Correlation
pearson <- cor.test(data2$Engine_size, data2$Horsepower, method = "pearson")
pearson
```

```
##
##  Pearson's product-moment correlation
##
## data:  data2$Engine_size and data2$Horsepower
## t = 18.707, df = 150, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7815469 0.8787984
## sample estimates:
```

```
##       cor
## 0.8366494
```

Pearson Correlation = 0.8366494. Value suggests that there is strong correlation which was also seen graphically.

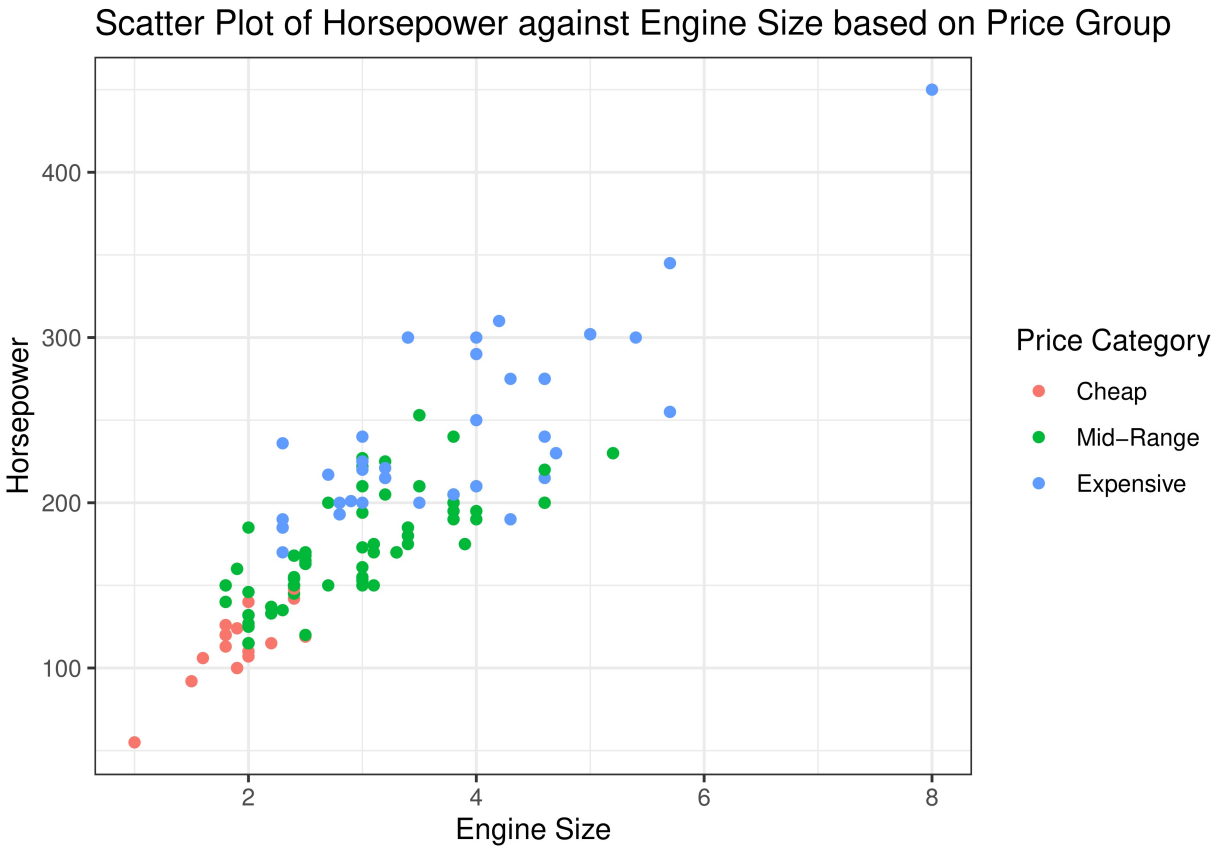
c. Let's break down prices into three groups: the cheapest cars being between 0 and $15000, and mid-range cars being between $15000 and $30000, and the expensive cars costing over $30000. You can use sample code such as this to break price into these three categories.

```
#Breaking price into 3 groups
data1$price_category<- cut(data1$price,breaks=c(0,15000,30000,200000),labels=c("Cheap","Mid-Range","Exp
data1$price_category
```

```
##   [1] Cheap     Cheap     Cheap     Cheap     Cheap     Cheap     Cheap
##   [8] Cheap     Mid-Range Cheap     Cheap     Cheap     Cheap     Cheap
##  [15] Cheap     Cheap     Cheap     Mid-Range Mid-Range Cheap     Mid-Range
##  [22] Cheap     Mid-Range Mid-Range Mid-Range Expensive Expensive Mid-Range
##  [29] Mid-Range Mid-Range Mid-Range Mid-Range Mid-Range Mid-Range Mid-Range
##  [36] Mid-Range Mid-Range Mid-Range Mid-Range Expensive Mid-Range Mid-Range
##  [43] Mid-Range Mid-Range Mid-Range Expensive Mid-Range Mid-Range Cheap
##  [50] Mid-Range Expensive Mid-Range Cheap     Mid-Range Expensive Mid-Range
##  [57] Cheap     Mid-Range Mid-Range Mid-Range Mid-Range Mid-Range Mid-Range
##  [64] Expensive Mid-Range Mid-Range Expensive Expensive Mid-Range Mid-Range
##  [71] Expensive Mid-Range Expensive Mid-Range Mid-Range Mid-Range Mid-Range
##  [78] Mid-Range Mid-Range Mid-Range Expensive Expensive Mid-Range Mid-Range
##  [85] Mid-Range Mid-Range Mid-Range Mid-Range Mid-Range Mid-Range Expensive
##  [92] Mid-Range Mid-Range Mid-Range Mid-Range Mid-Range Mid-Range Mid-Range
##  [99] Mid-Range Expensive Mid-Range Mid-Range Mid-Range Mid-Range Expensive
## [106] Expensive Expensive Expensive Expensive Mid-Range Mid-Range Expensive
## [113] Expensive Mid-Range Expensive Expensive Expensive Mid-Range Expensive
## [120] Mid-Range Expensive Mid-Range Expensive Expensive Expensive Mid-Range
## [127] Mid-Range Mid-Range Mid-Range Mid-Range Mid-Range Expensive Expensive
## [134] Mid-Range Expensive Expensive Expensive Expensive Expensive Mid-Range
## [141] Mid-Range Mid-Range Mid-Range Expensive Mid-Range Mid-Range Expensive
## [148] Expensive Expensive Expensive Expensive Expensive
## Levels: Cheap Mid-Range Expensive
```

d. Plot total horsepower (y axis) against engine size (x axis), but now color points based on which price group they fall into. You can do this by specifying the `col=new_var` option in the `plot()` function. Comment on the results

```
# Scatter Plot of Horsepower against Engine Size based on Price group
ggplot(data=data1, mapping = aes(x = Engine_size, y = Horsepower)) +
  geom_point(aes(color = price_category)) +  theme_bw() +
  ggtitle("Scatter Plot of Horsepower against Engine Size based on Price Group") +
  xlab("Engine Size") + ylab("Horsepower") +labs(colour="Price Category")
```



Comment: It is evident that cheap cars have smaller engine size and horsepower, followed by mid range cars, while expensive cars have most superior engine size and horsepower. It can be observed that around (3,200) mid-range and expensive cars both are present which essentially means that for that engine size and horsepower, mid range cars are available instead of expensive cars.

e. Create a new categorical variable that indicates whether the fuel efficiency is greater than 30. Use the following example code as a template:

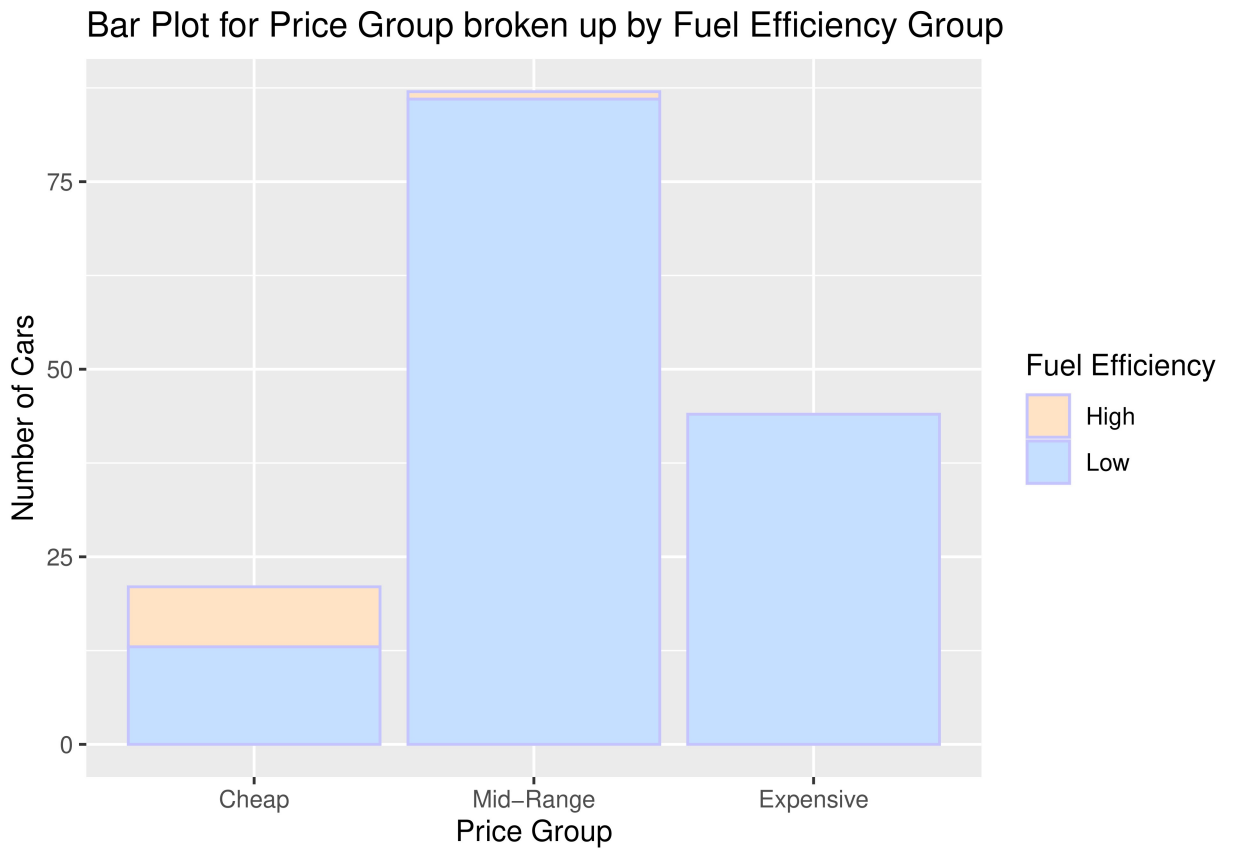
```
#Created new variable fuel_eff_category
data1$fuel_eff_category <- ifelse(data1$Fuel_efficiency > 30, "high_efficient",
                                  "low_efficient")
data1$fuel_eff_category
```

```
##   [1] "high_efficient" "high_efficient" "low_efficient"  "high_efficient"
##   [5] "high_efficient" "high_efficient" "high_efficient" "high_efficient"
##   [9] "high_efficient" "high_efficient" "low_efficient"  "low_efficient"
##  [13] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [17] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [21] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [25] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [29] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [33] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [37] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [41] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [45] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [49] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
```

```
##  [53] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [57] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [61] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [65] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [69] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [73] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [77] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [81] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [85] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [89] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [93] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
##  [97] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
## [101] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
## [105] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
## [109] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
## [113] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
## [117] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
## [121] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
## [125] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
## [129] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
## [133] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
## [137] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
## [141] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
## [145] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
## [149] "low_efficient"  "low_efficient"  "low_efficient"  "low_efficient"
```

f. Create a stacked barplot with a bar for each price group (i.e. use `new_var` from above). Each bar should be broken up into two pieces: one for high fuel efficiency and one for low fuel efficiency. Make sure to label your axes and add a legend. Comment on the results.

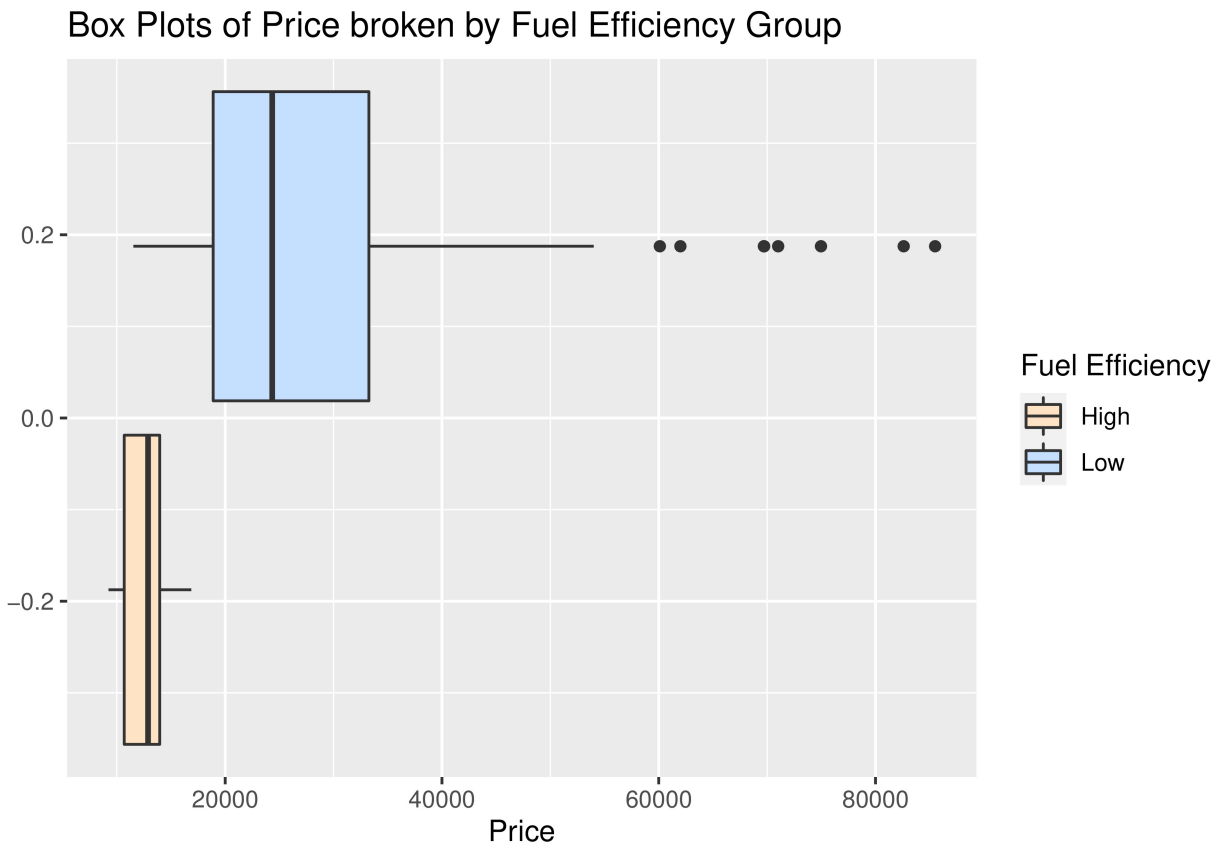
```
# Bar Plot for Price Group broken up by Fuel Efficiency Group
library(ggplot2)

ggplot(data1, aes(x = price_category, fill = fuel_eff_category)) +  geom_bar() +
  scale_fill_manual(values=c("#ffe4c4", "#c4dfff"), labels = c("High", "Low")) +
  geom_bar(color = "#c7c4ff") +
  guides(fill = guide_legend(title = "Fuel Efficiency")) +
  ggtitle("Bar Plot for Price Group broken up by Fuel Efficiency Group ") +
  xlab("Price Group") + ylab("Number of Cars")
```



Comment: Using the plot we can observe that for cheap cars, there are almost as many as high fuel efficient cars as low fuel efficient cars. In the mid-range section, there are very few high fuel efficient cars while, in expensive price section there are no high fuel efficient cars. It can be inferred that as the price of car increases, fuel efficiency decreases. In real life this trend is followed as expensive cars have bigger engines which require more fuel and hence less efficiency.

g. Make side-by-side boxplots of `price` (not price groups), broken down by fuel efficiency group (low vs. high). Comment on the result:

```
# Box Plots of Price broken by Fuel Efficiency Group
ggplot(data1, aes(x=data1$price, fill= fuel_eff_category)) +
  geom_boxplot() + scale_fill_manual(values=c("#ffe4c4", "#c4dfff"),
                                     labels = c("High", "Low"))  +
  ggtitle("Box Plots of Price broken by Fuel Efficiency Group") + xlab("Price")+
  guides(fill = guide_legend(title = "Fuel Efficiency"))
```



Comment: In this plot it can be observed that there are very few cars in the high efficiency category with respect to low efficiency. Median price of the highly efficient cars lies around 12000 while the median price of low efficient cars lies around 25000. Outliers can be observed.Box-plot of High fuel efficient cars in comparitively smaller than low fuel efficient cars.

Question 2) Probability: PPV and NPV. A test is created to help detect a disease The test is administered to a group of 84 subjects known to have the disease. Of this group, 59 test positive. The test is also administered to a group of 428 subjects known to not have the disease. Of this group, 12 test positive.

a. Present this data in a tabular form similar to the following:

```
#Creating matrix and converting into table
tab <- matrix(c(59, 12, 71, 25, 416, 441, 84, 428, 512), ncol=3, byrow=TRUE)
rownames(tab) <- c('Positive','Negative','Total')
colnames(tab) <- c('Disease','Not Disease','Total')
tab <- as.table(tab)
library(knitr)
    kable(x=tab, digits=2,row.names=T, format="markdown")
```

	Disease	Not Disease	Total
Positive	59	12	71
Negative	25	416	441
Total	84	428	512

b. Calculate the sensitivity and specificity of this test directly from the data.

```r
#Calculating Sensitivity
sensitivity = tab[1,1]/(tab[1,1]+tab[2,1])
sensitivity
```

```
## [1] 0.702381
```

0.702381 is the sensitivity

```r
#Calculating Specificity
specificity = tab[2,2]/(tab[2,2]+tab[1,2])
specificity
```

```
## [1] 0.9719626
```

0.9719626 is the Specificity

c. Assume that the prevalence of the disease is 2.7%. Calculate the NPV and PPV with this prevalence.

```r
#Finding PPV
prevalence = 0.027
PPV = (sensitivity * prevalence)/((sensitivity * prevalence)+
                                    ((1 - specificity) * (1 -prevalence)))
PPV
```

```
## [1] 0.410086
```

PPV is 0.410086

```r
#Finding NPV
NPV= (specificity*(1-prevalence))/(((1-sensitivity)*prevalence)+
                                     ((specificity)*(1-prevalence)))
NPV
```

```
## [1] 0.9915747
```

NPV is 0.9915747

d. What conclusions can be drawn regarding the effectiveness of this test?

The chance that a person with a positive test result actually has the disease is 41.0086% which is pretty bad considering almost 60% people that were given positive result, don't actually have disease. And the chance that a person with a negative test result actually does not have the disease is 99.15747%, this is very accurate as only less than 1% people have disease if the report comes negative. The reason for such bad PPV is prevalence value. As prevalence is so small, PPV will be bad. In this case we can't trust on single examination result.