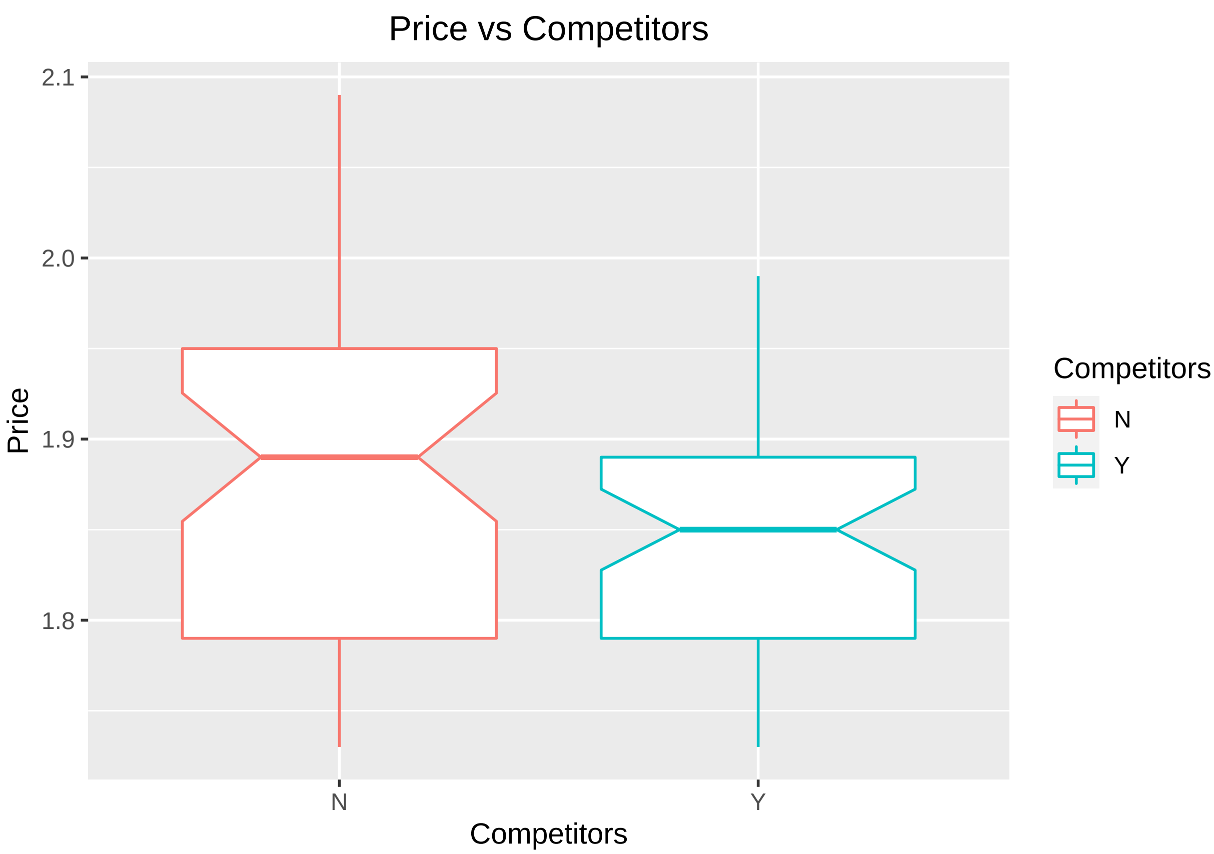
Data Mining Assignment 1

Patrick Chase

2/5/2021

1A)

```
price.comp <- ggplot(GasPrices, aes(x=Competitors, y=Price, color=Competitors)) +
  geom_boxplot(notch = TRUE) + ggtitle("Price vs Competitors")+
  theme(plot.title = element_text(hjust = 0.5))
price.comp
```



Given traditional economic theory, if a station has competitors we would expect a lower average price. "Price vs Competitors" provides evidence that this is true. Gas stations with competitors have both an average lower price, as well as a distribution that is lower than those without competitors.

1B)

```
price.inc <- ggplot(data = GasPrices) +
  geom_point(mapping = aes(x=Income, y=Price, color = Competitors)) +
  ggtitle("Price vs Income") +
  theme(plot.title = element_text(hjust = 0.5))
```

```
price.inc
```

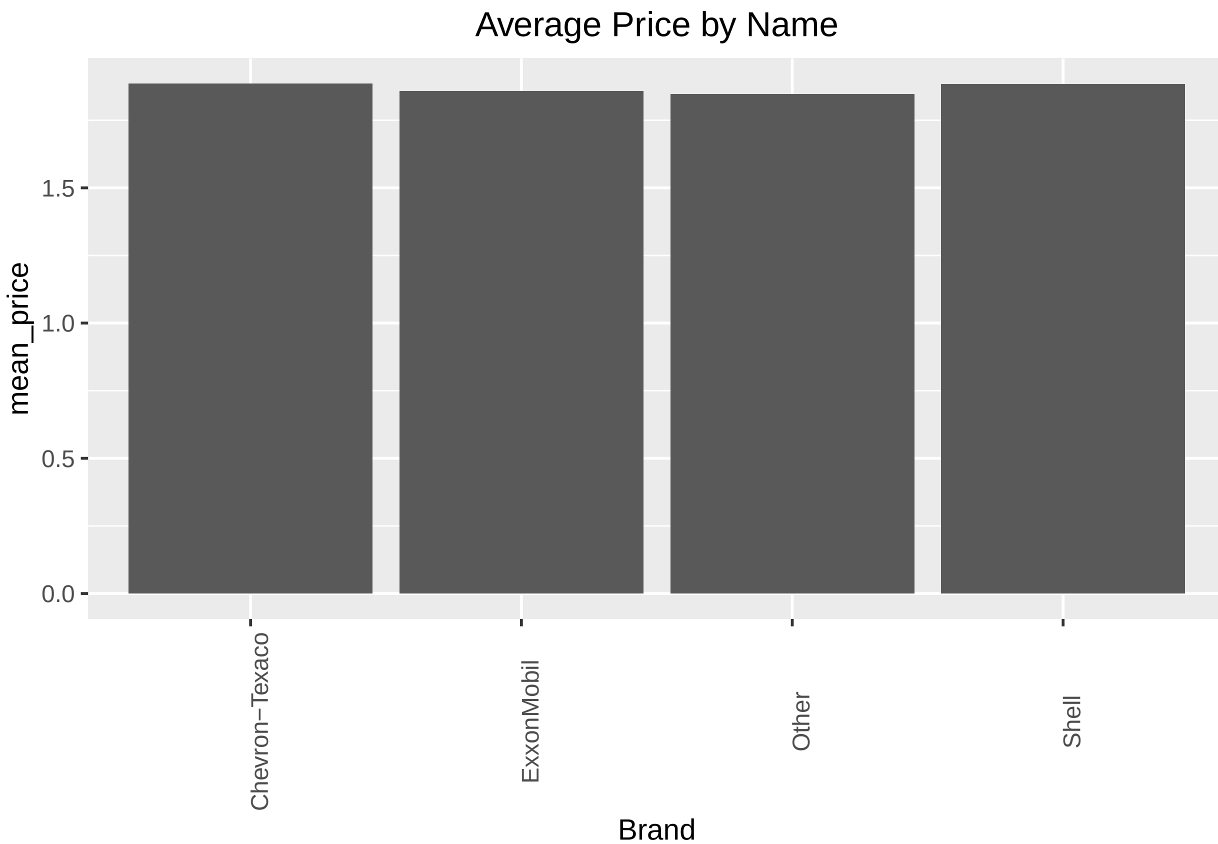
The claim that richer areas tend to have higher gas prices seems to be mildly supported by the available data. On my own, I chose to color each observation based on whether or not there were competitors near by, which shows an interesting relationship. There seems to be less competition at the extremes of income.

1C)

```
brand_price <- GasPrices %>%
  group_by(Brand) %>%
  summarize(mean_price = mean(Price))
brand_price
```

```
## # A tibble: 4 x 2
##   Brand          mean_price
## * <chr>               <dbl>
## 1 Chevron-Texaco       1.88
## 2 ExxonMobil           1.86
## 3 Other                1.85
## 4 Shell                1.88
```

```
ggplot(data = brand_price) +
  geom_col(mapping = aes(x=Brand, y=mean_price),
           position = 'dodge') +
  theme(axis.text.x = element_text(angle = 90)) +
  ggtitle("Average Price by Name")+
  theme(plot.title = element_text(hjust = 0.5))
```



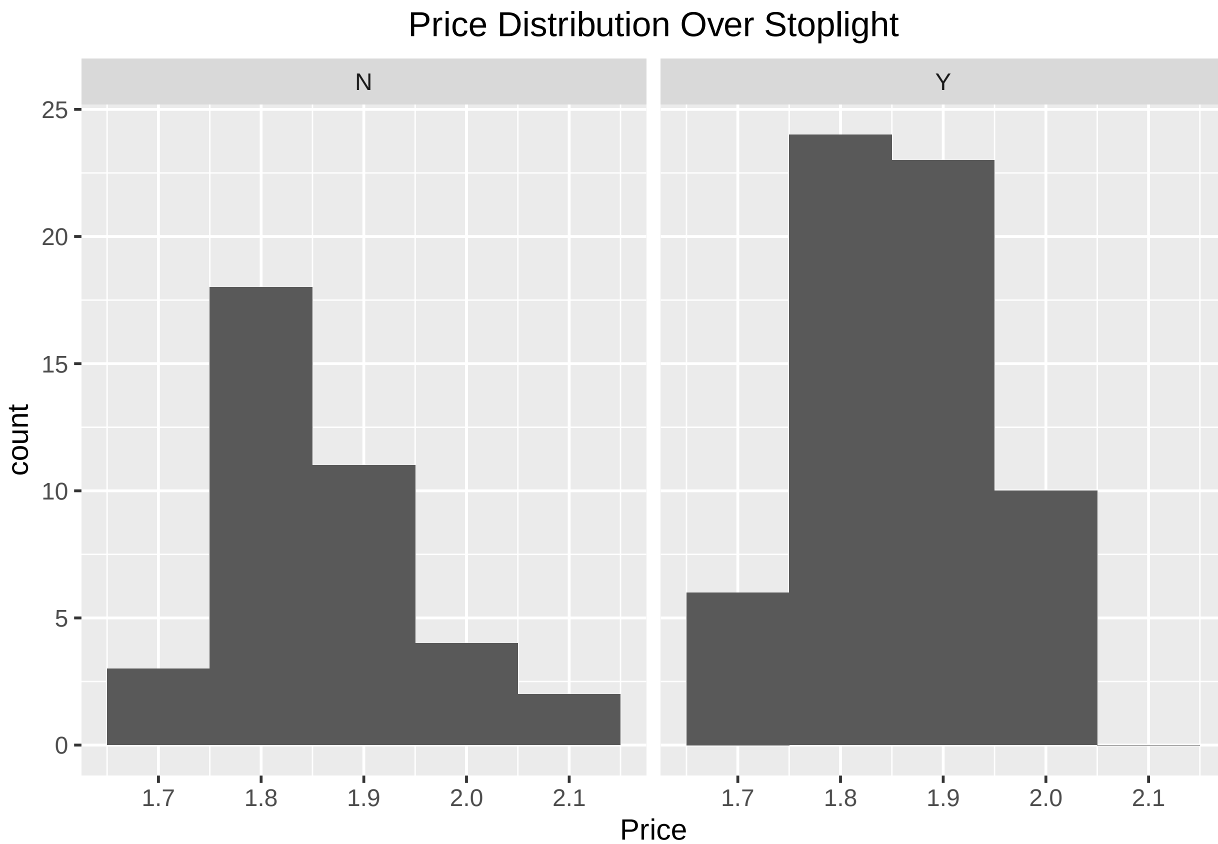
The claim that Shell gas stations charge more than others does not seem to be supported by this data. Visually, it appears that shell is charging about the same average price as all the other stations.

1D)

```
stoplight <- GasPrices %>%
  group_by(Stoplight) %>%
  summarize(mean_price = mean(Price))
stoplight
```

```
## # A tibble: 2 x 2
##   Stoplight mean_price
## * <chr>          <dbl>
## 1 N               1.87
## 2 Y               1.86
```

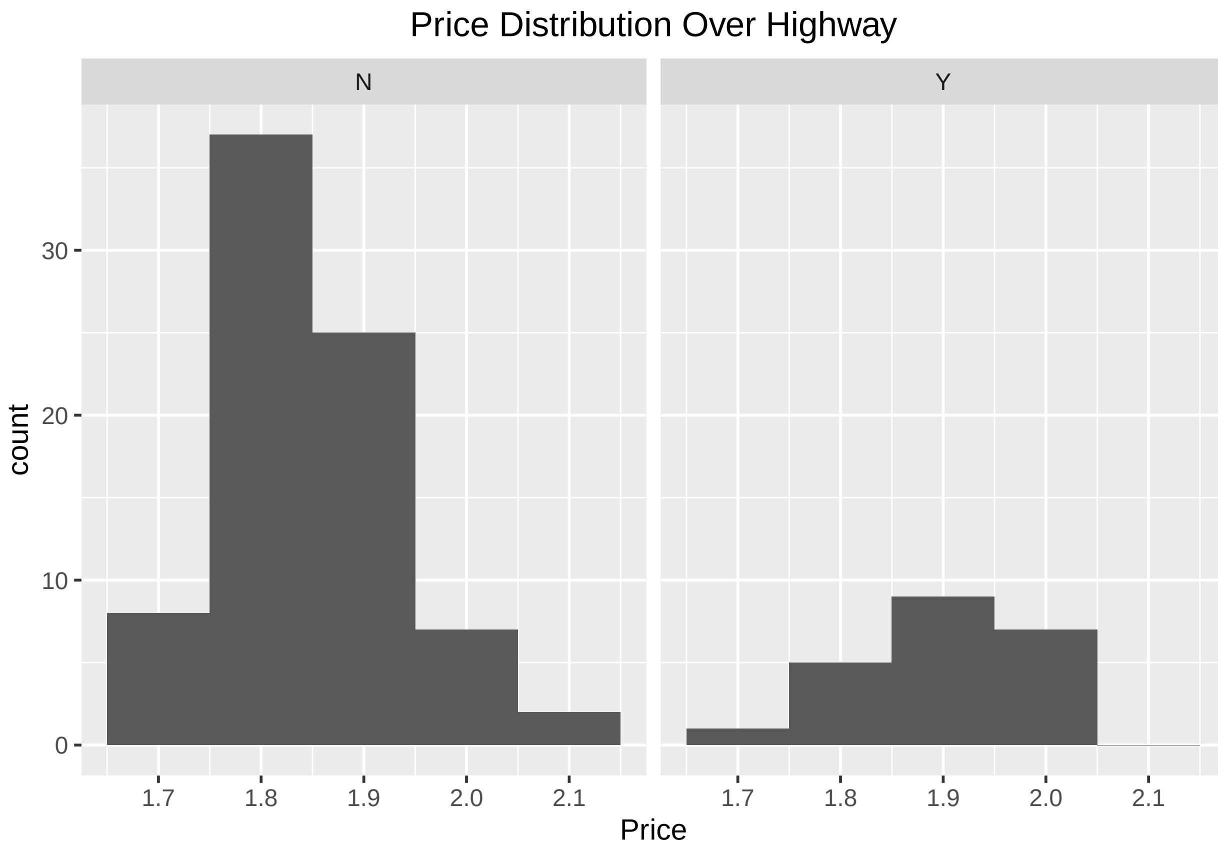
```
ggplot(data = GasPrices) +
  geom_histogram(aes(x=Price), binwidth = .1) +
  facet_wrap(~Stoplight) +
  ggtitle("Price Distribution Over Stoplight")+
  theme(plot.title = element_text(hjust = 0.5))
```



I don't think this visualization supports the claim that gas stations near stoplights charge more for gas. The average price near stoplights is probably higher, however prices that aren't near a stop light have a wider range and a higher max price.

1E)

```
ggplot(data = GasPrices) +
  geom_histogram(aes(x=Price), binwidth = .1) +
  facet_wrap(~Highway) +
  ggtitle("Price Distribution Over Highway")+
  theme(plot.title = element_text(hjust = 0.5))
```



I chose to generate a faceted histogram in order to show the difference in between prices given distance from a highway. Preliminarily, I'd say that there is some evidence that suggests that prices are higher when one is close to a highway. That said, the differing counts between the two indicate that we may have some selection bias. Our sample of stations near the highway may not be representative and as such should be taken with a grain of salt.

2)

```
bikeshare <- read.csv("https://raw.githubusercontent.com/jgscott/ECO395M/master/data/bikeshare.csv")
head(bikeshare)
```

```
##   instant     dteday season yr mnth hr holiday weekday workingday weathersit
## 1       1 2011-01-01      1  0    1  0       0       6          0          1
## 2       2 2011-01-01      1  0    1  1       0       6          0          1
## 3       3 2011-01-01      1  0    1  2       0       6          0          1
## 4       4 2011-01-01      1  0    1  3       0       6          0          1
## 5       5 2011-01-01      1  0    1  4       0       6          0          1
## 6       6 2011-01-01      1  0    1  5       0       6          0          2
##   temp total
## 1 0.24    16
## 2 0.22    40
## 3 0.22    32
## 4 0.24    13
## 5 0.24     1
## 6 0.24     1
```

Plot A

```
df1 <- bikeshare %>%
  group_by(hr, workingday) %>%
```

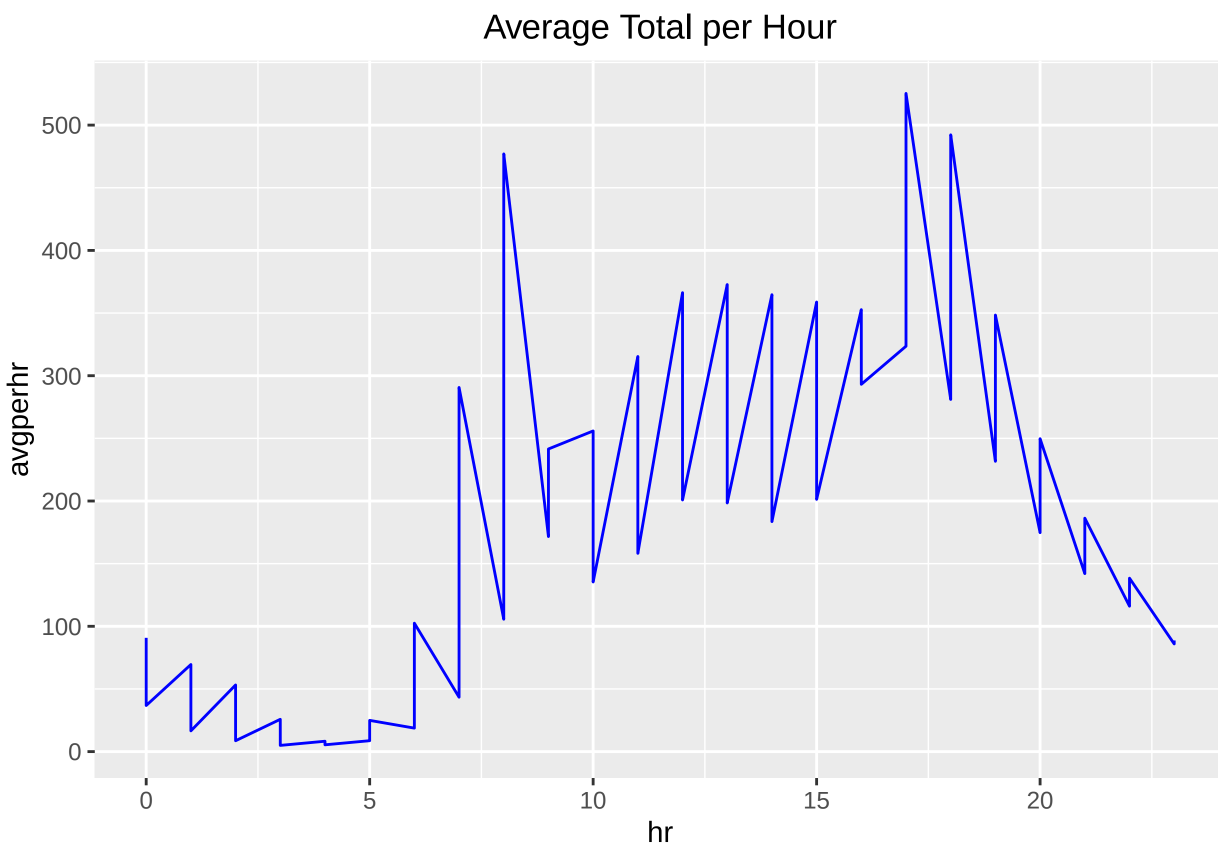
```r
  summarize(totalhr = sum(total),
            number_hr = n(),
            avgperhr = totalhr/number_hr)
```

```
## `summarise()` has grouped output by 'hr'. You can override using the `.groups` argument.
```

```r
df1
```

```
## # A tibble: 48 x 5
## # Groups:   hr [24]
##       hr workingday totalhr number_hr avgperhr
##    <int>      <int>   <int>     <int>    <dbl>
##  1     0          0   20884       230    90.8
##  2     0          1   18246       496    36.8
##  3     1          0   15987       230    69.5
##  4     1          1    8177       494    16.6
##  5     2          0   12123       228    53.2
##  6     2          1    4229       487     8.68
##  7     3          0    5851       227    25.8
##  8     3          1    2323       470     4.94
##  9     4          0    1876       227     8.26
## 10     4          1    2552       470     5.43
## # ... with 38 more rows
```

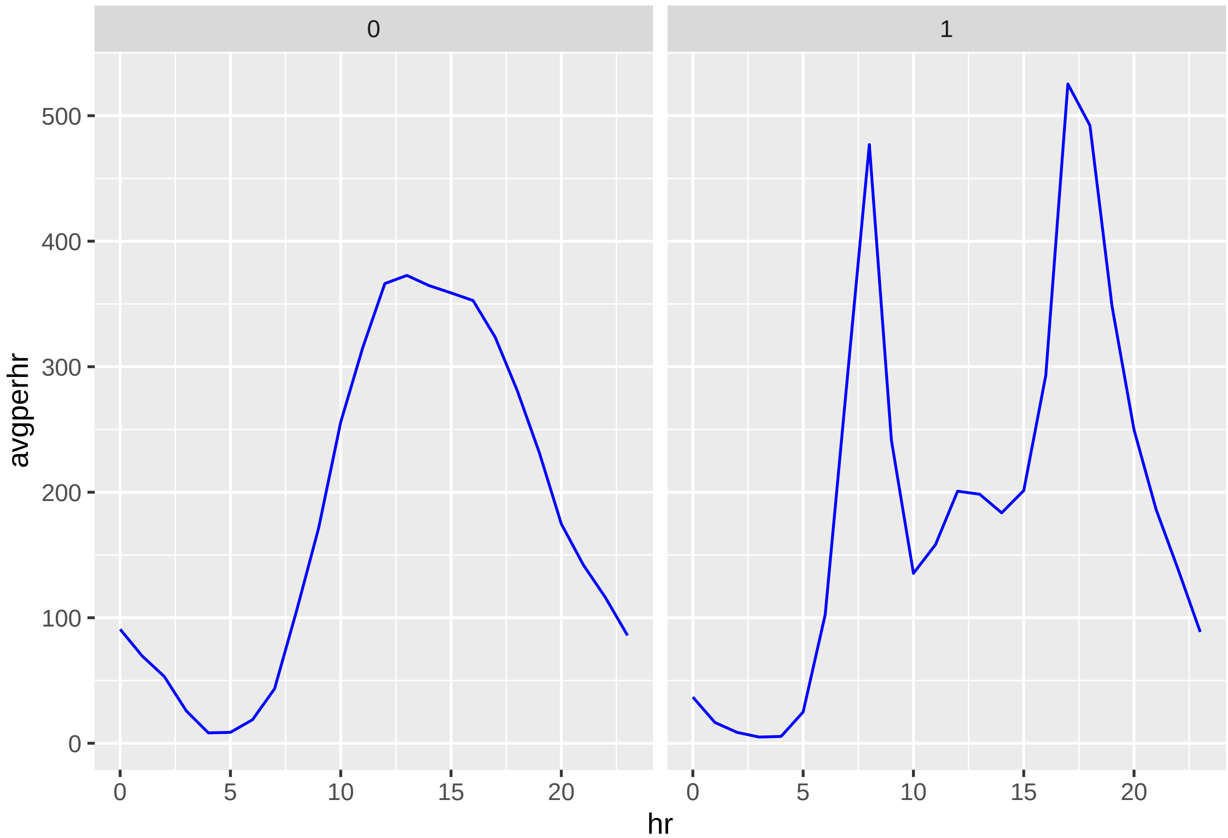
```r
ggplot(data = df1, aes(x = hr, y = avgperhr)) +
  geom_line(color="blue") +
  ggtitle("Average Total per Hour")+
  theme(plot.title = element_text(hjust = 0.5))
```



Plot A is showing the average bike rentals per hour on the y-axis and a 24 hour time scale on the x axis. We see peak demand between the times of 0700 to 1000 and 1600 to 1900. In the United States these are the traditional commuting hours. As workers move to and from work we see the highest volume of rentals, on average.

Plot B

```
ggplot(data = df1, aes(x = hr, y = avgperhr)) +
  geom_line(color="blue") +
  facet_wrap(~workingday)
```




```
  ggtitle("Average Total per Hour")+
  theme(plot.title = element_text(hjust = 0.5))
```

```
## NULL
```

Plot B shows us similar information as Plot A but broken down by holidays (0) vs working days (1). The y-axis represents the total amount rented in a given hour shown on the x-axis utilizing a 24 hour time scale. These graphs demonstrate that peak demand is largely being driven by cycles related to commuting to and from work in the population.

Plot C

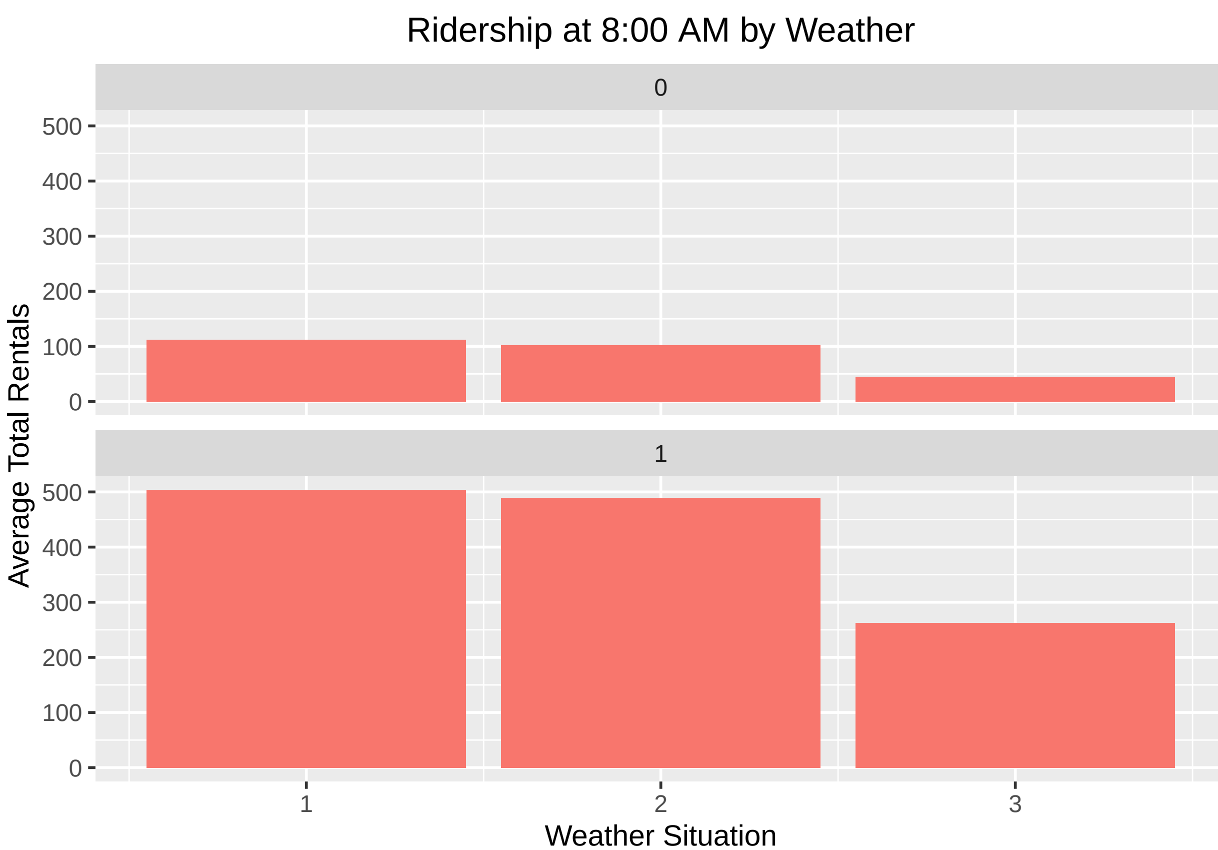
```
df2 <- bikeshare%>%
  filter(hr==8) %>%
  group_by(workingday, weathersit)%>%
  summarize(totalhr = sum(total),
            number_hr = n(),
            avgperhr = totalhr/number_hr)
```

```
## `summarise()` has grouped output by 'workingday'. You can override using the `.groups` argument.
```

```r
df2
```

```
## # A tibble: 6 x 5
## # Groups:   workingday [2]
##   workingday weathersit totalhr number_hr avgperhr
##        <int>      <int>   <int>     <int>    <dbl>
## 1          0          1   17916       160    112.
## 2          0          2    5904        58    102.
## 3          0          3     586        13     45.1
## 4          1          1  141082       280    504.
## 5          1          2   83700       171    489.
## 6          1          3   11813        45    263.
```

```r
ggplot(data = df2, mapping = aes(x = weathersit, y =avgperhr, fill = "red")) +
  geom_col()+
  facet_wrap(~workingday, nrow = 2) +
    labs(
    title = "Ridership at 8:00 AM by Weather",
    x="Weather Situation",
    y="Average Total Rentals"
  ) +
  theme(plot.title = element_text(hjust = 0.5), legend.position = "none")
```



The y-axis represents average total rentals at 8:00 AM and the y-axis is a measure of the weather, with higher numbers representing more adverse weather. Plot C shows that regardless of whether or not it's a holiday, as the weather get's progressively worse ridership falls. While the scale of the decline is larger on workdays, both holidays and workdays show a similar relationship.

3)

```r
abia <- read.csv("https://raw.githubusercontent.com/jgscott/ECO395M/master/data/ABIA.csv")
head(abia)
```

```
##   Year Month DayofMonth DayOfWeek DepTime CRSDepTime ArrTime CRSArrTime
## 1 2008     1          1         2     120       1935     309       2130
## 2 2008     1          1         2     555        600     826        835
## 3 2008     1          1         2     600        600     728        729
## 4 2008     1          1         2     601        605     727        750
## 5 2008     1          1         2     601        600     654        700
## 6 2008     1          1         2     636        645     934        932
##   UniqueCarrier FlightNum TailNum ActualElapsedTime CRSElapsedTime AirTime
## 1            9E      5746  84129E               109            115      88
## 2            AA      1614  N438AA               151            155     133
## 3            YV      2883  N922FJ               148            149     125
## 4            9E      5743  89189E                86            105      70
## 5            AA      1157  N4XAAA                53             60      38
## 6            NW      1674   N967N               178            167     145
##   ArrDelay DepDelay Origin Dest Distance TaxiIn TaxiOut Cancelled
## 1      339      345    MEM  AUS      559      3      18         0
## 2       -9       -5    AUS  ORD      978      7      11         0
## 3       -1        0    AUS  PHX      872      7      16         0
## 4      -23       -4    AUS  MEM      559      4      12         0
## 5       -6        1    AUS  DFW      190      5      10         0
## 6        2       -9    AUS  MSP     1042     11      22         0
##   CancellationCode Diverted CarrierDelay WeatherDelay NASDelay SecurityDelay
## 1                         0          339            0        0             0
## 2                         0           NA           NA       NA            NA
## 3                         0           NA           NA       NA            NA
## 4                         0           NA           NA       NA            NA
## 5                         0           NA           NA       NA            NA
## 6                         0           NA           NA       NA            NA
##   LateAircraftDelay
## 1                 0
## 2                NA
## 3                NA
## 4                NA
## 5                NA
## 6                NA
```

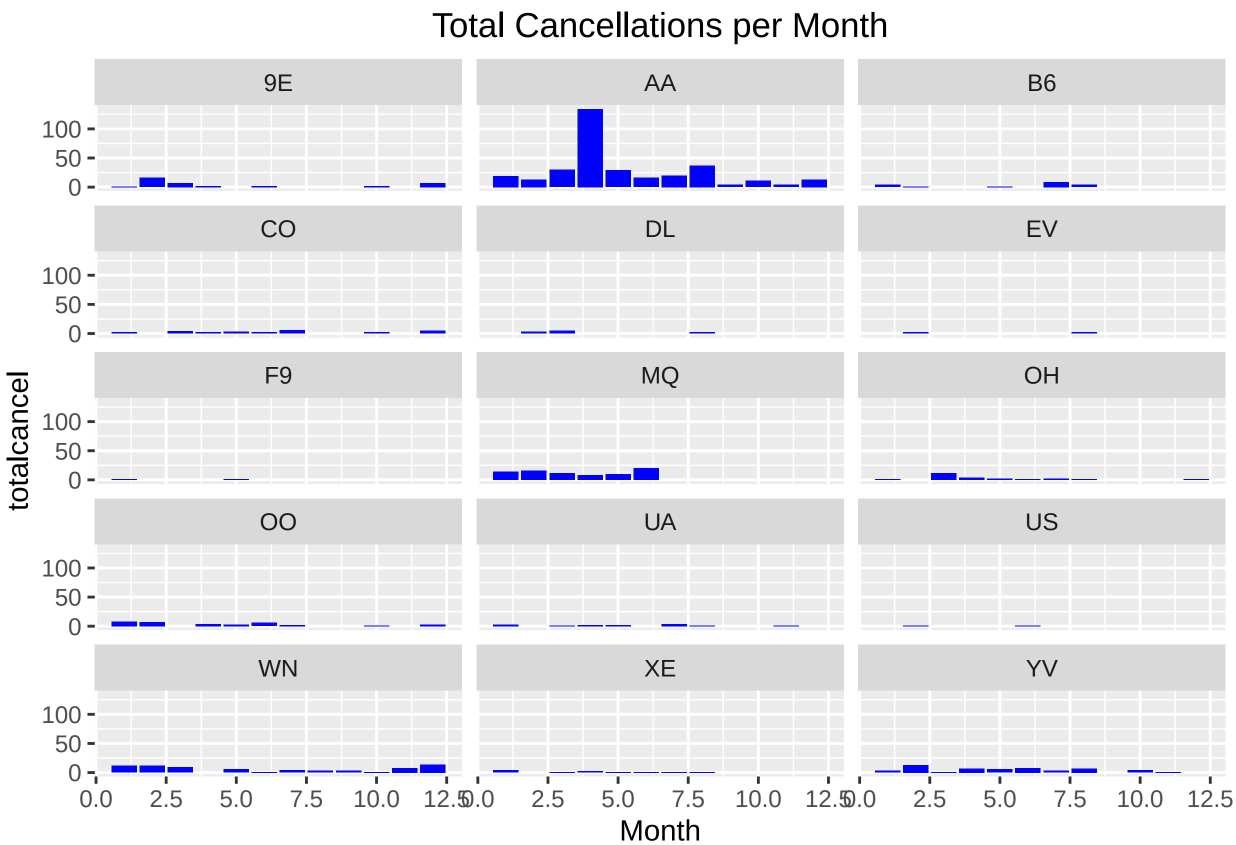
I'd like to categorize the carriers, as defined by the UniqueCarrier variable, by the amount of canceled flights per month. For our purposes we will focus only on cancellations that are explicitly identified as being related to the carrier. This will allow us to mitigate the impact of random chance related to weather, security, and NAS cancellations.

```r
df3 <- abia%>%
  filter(CancellationCode=="A") %>%
  group_by(UniqueCarrier, Month) %>%
  summarize(totalcancel = sum(Cancelled))
```

```
## `summarise()` has grouped output by 'UniqueCarrier'. You can override using the `.groups` argument.
```

```r
ggplot(data = df3, aes(x = Month, y = totalcancel)) +
  geom_col(position = "dodge", fill = "blue") +
  facet_wrap(~UniqueCarrier, nrow = 5)+
```

```r
  ggtitle("Total Cancellations per Month")+
  theme(plot.title = element_text(hjust = 0.5))
```

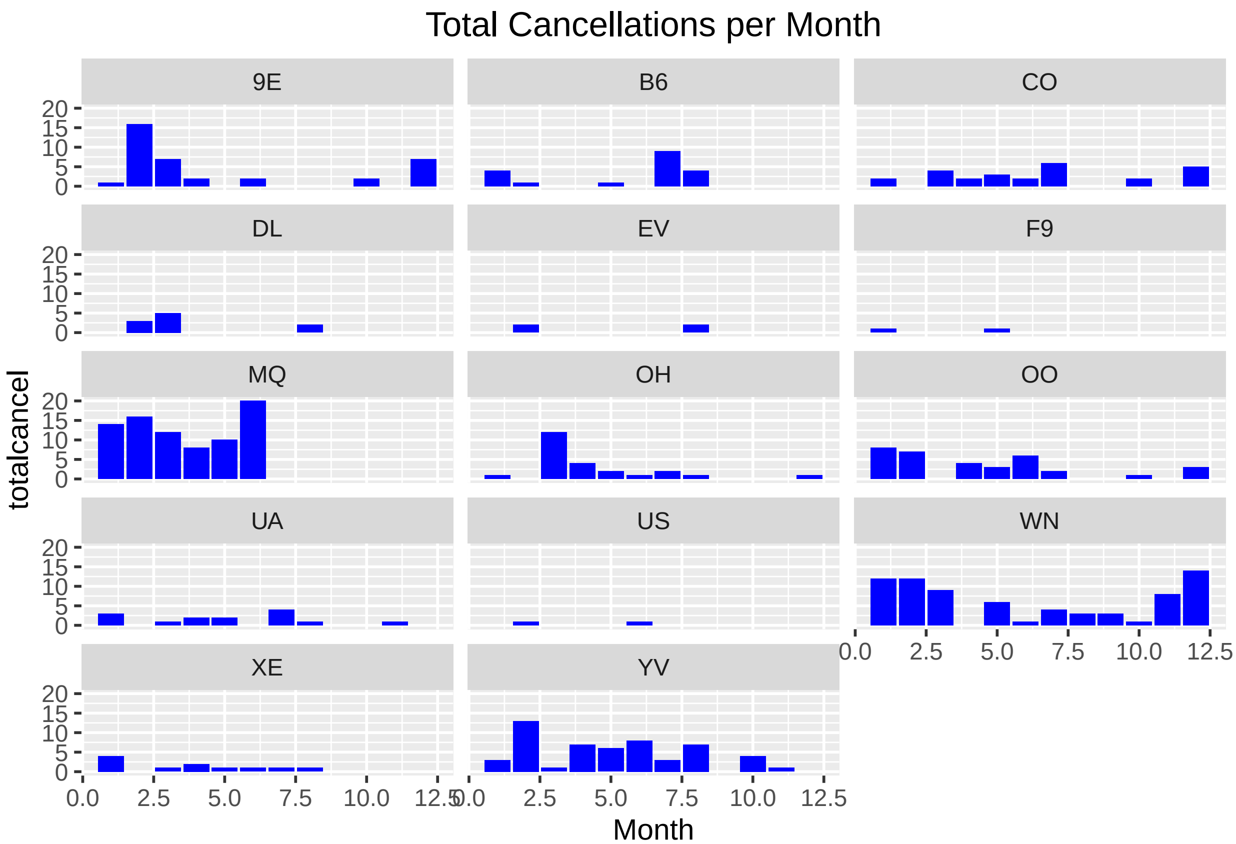


From the view of cancellations we can see that AA seems to have really struggled during 2008. To be more specific, in the month of April AA had 134 carrier cancellations while all other carriers had less than 20! This points to a rough year for AA. I feel comfortable stating that in 2008 they were the worst carrier out of Austin purely from the perspective of total canceled flights. However, after that it becomes more difficult. Next, I'll filter AA out and see if there are any other clear relationships to be seen.

```r
df4 <- abia%>%
  filter(CancellationCode=="A" , UniqueCarrier != "AA") %>%
  group_by(UniqueCarrier, Month) %>%
  summarize(totalcancel = sum(Cancelled))
```

```
## `summarise()` has grouped output by 'UniqueCarrier'. You can override using the `.groups` argument.
```

```r
ggplot(data = df4, aes(x = Month, y = totalcancel)) +
  geom_col(position = "dodge", fill = "blue") +
  facet_wrap(~UniqueCarrier, nrow = 5)+
  ggtitle("Total Cancellations per Month")+
  theme(plot.title = element_text(hjust = 0.5))
```



Now that our scale has adjusted, we can see two categories beginning to emerge. Generally speaking, there are airlines with more than 10 cancellations a month and those with less than 10 cancellations a month. For my ranking, I'll identify two distinct categories. The better category will be compromised of those carriers with less than 10 cancellations in all months. They are XE, UA, US, DL, EV, OO, and F9. Second, the less than ideal category is made up of carriers with more than 10 cancellations in any month. They are AA, 9E, MQ, OH, WN, YV.

A caveat to this analysis that should be considered is that it is not accounting for the size of airlines or volume of flights out of Austin. It's possible that AA has more cancellations because they have 5-10 times more flights in total then any of the other airlines. If I were to continue this analysis, the next step I would take would be to control for total size of airline and flight volume out of Austin specifically.

4)

```
sclass <- read.csv("https://raw.githubusercontent.com/jgscott/ECO395M/master/data/sclass.csv")
head(sclass)
```

```
##   id trim subTrim condition isOneOwner mileage year  color displacement
## 1  2  320    unsp      Used          f  129948 1995   Gold        3.2 L
## 2  4  320    unsp      Used          f  140428 1997  White        3.2 L
## 3  7  420    unsp      Used          f  113622 1999 Silver        4.2 L
## 4  8  420    unsp      Used          f  167673 1999 Silver        4.2 L
## 5 11  500    unsp      Used          f   63457 1997 Silver        5.0 L
## 6 13  430    unsp      Used          f   82419 2002  White        4.3 L
##       fuel state region soundSystem wheelType wheelSize featureCount price
## 1 Gasoline    PA    Mid     Premium     Alloy      unsp           26  6595
## 2 Gasoline    NY    Mid        Bose     Alloy      unsp           22  7993
## 3 Gasoline    NJ    Mid        unsp     Alloy      unsp           24  5995
## 4 Gasoline    GA    SoA        unsp     Alloy      unsp           24  3000
```

```
## 5 Gasoline    CO    Mtn      Alpine     Alloy        20            23 14975
## 6 Gasoline    NJ    Mid         Bose     Alloy        16            35  7400
```

```
library(parallel)
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(foreach)
```

```
##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
library(FNN)
library(rsample)
library(modelr)

df350 <- sclass %>%
  filter(trim == 350)%>%
  summarize(mileage = mileage,
            price = price)
head(df350)
```

```
##   mileage price
## 1   21929 55994
## 2   17770 60900
## 3   29108 54995
## 4   35004 59988
## 5   66689 37995
## 6   19567 59977
```

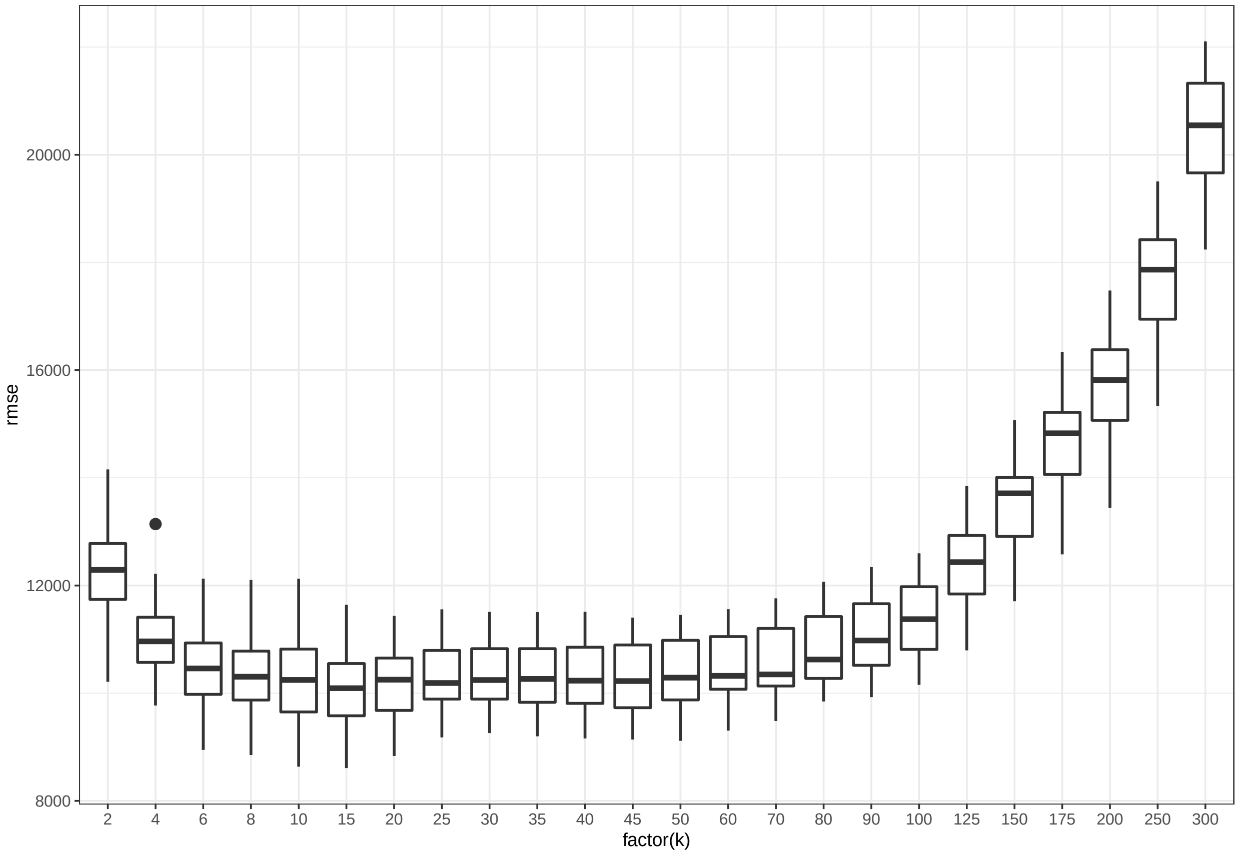
trim = 350 k validation

```
k_val = c(2, 4, 6, 8, 10, 15, 20, 25, 30, 35, 40, 45,
           50, 60, 70, 80, 90, 100, 125, 150, 175, 200, 250, 300)
df350_out = foreach(i=1:20, .combine='rbind') %dopar% {
  df350_split = initial_split(df350, prop = .8)
  df350_train = training(df350_split)
  df350_test = testing(df350_split)
  rmse350 = foreach(k = k_val, .combine = 'c') %do% {
  model350 = knnreg(price ~ mileage, data = df350_train, k = k, use.all = TRUE)
  modelr:: rmse(model350, df350_test)
  }
  data.frame(k=k_val, rmse=rmse350)
}
```

```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

```r
df350_out = arrange(df350_out, k)
```

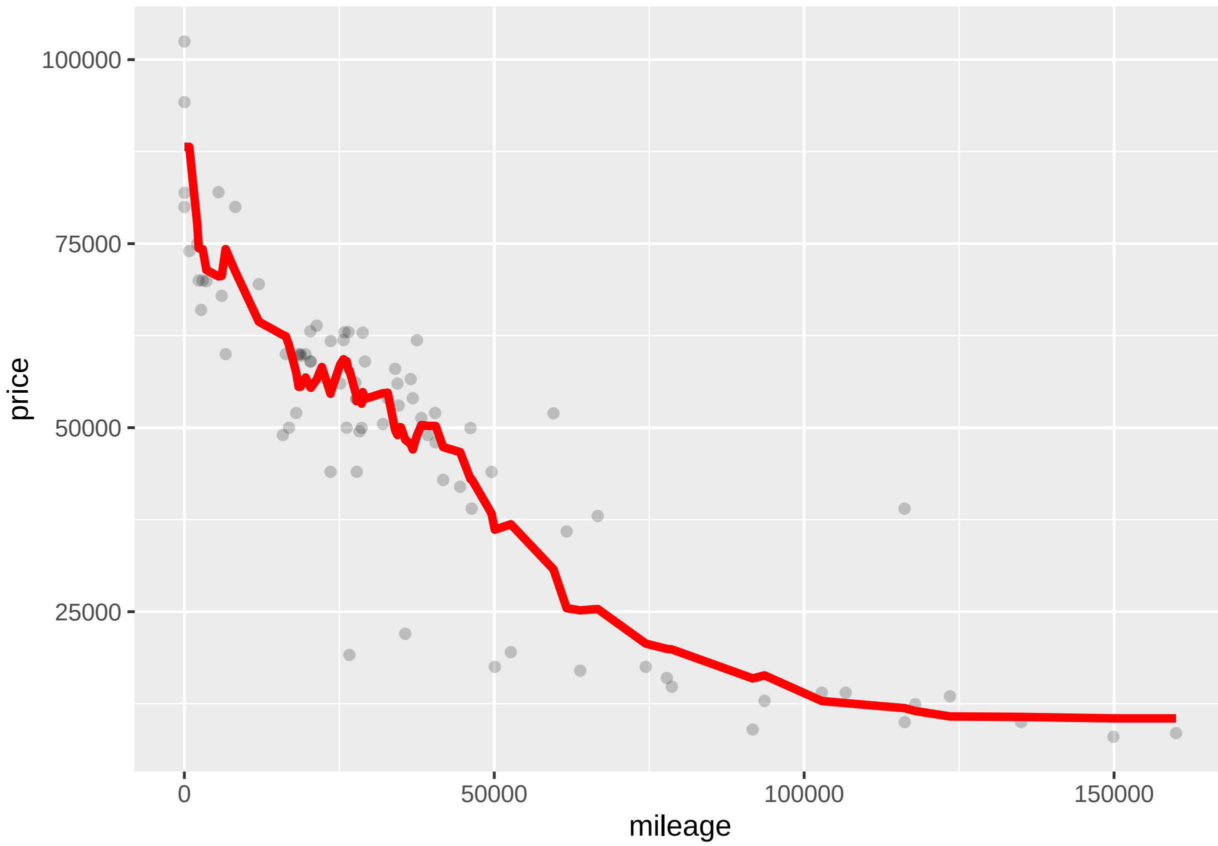
```r
ggplot(df350_out) + geom_boxplot(aes(x=factor(k), y=rmse)) + theme_bw(base_size=7)
```




```r
df350_split = initial_split(df350, prop = .8)
  df350_train = training(df350_split)
  df350_test = testing(df350_split)

knn15 = knnreg(price ~ mileage, data = df350_train, k=15)
rmse(knn15, df350_test)
```

```
## [1] 9355.68
```

```r
df350_test = df350_test%>%
  mutate(price_pred = predict(knn15, df350_test))

p_test = ggplot(data = df350_test) +
  geom_point(mapping = aes(x=mileage, y = price), alpha = .2)

p_test + geom_line(aes(x=mileage, y=price_pred), color='red', size = 1.5)
```



trim = 65 AMG analysis

```
df65 <- sclass %>%
  filter(trim == "65 AMG")%>%
  summarize(mileage = mileage,
            price = price)
head(df65)
```

```
##   mileage  price
## 1     106 235375
## 2      11 226465
## 3   74461  24995
## 4   73415  54981
## 5   17335 102500
## 6       7 230860
```

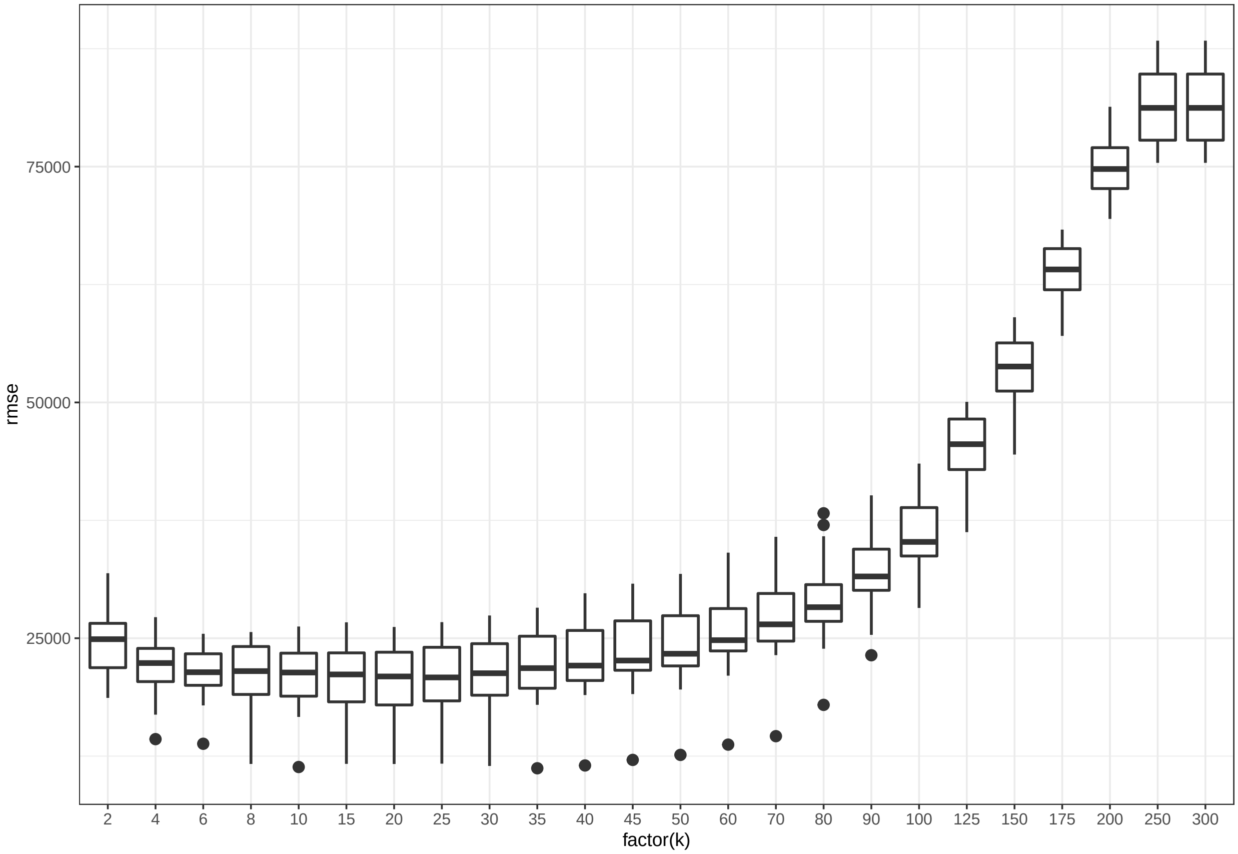
```
df65_out = foreach(i=1:20, .combine='rbind') %dopar% {
  df65_split = initial_split(df65, prop = .8)
  df65_train = training(df65_split)
  df65_test = testing(df65_split)
  rmse65 = foreach(k = k_val, .combine = 'c') %do% {
  model65 = knnreg(price ~ mileage, data = df65_train, k = k, use.all = TRUE)
  modelr:: rmse(model65, df65_test)
  }
  data.frame(k=k_val, rmse=rmse65)
}
```

```
## Warning in knnregTrain(train = structure(c(106, 74461, 73415, 7, 48398, : k =
## 250 exceeds number 234 of patterns
```

```
## Warning in knnregTrain(train = structure(c(106, 74461, 73415, 7, 48398, : k =
## 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(11, 74461, 73415, 17335, 7, 48398, :
## k = 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(11, 74461, 73415, 17335, 7, 48398, :
## k = 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 17335, : k =
## 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 17335, : k =
## 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 17335, : k =
## 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 17335, : k =
## 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 17335, : k =
## 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 17335, : k =
## 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(11, 74461, 17335, 7, 48398, 61500, :
## k = 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(11, 74461, 17335, 7, 48398, 61500, :
## k = 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 73415, 17335, 7, 48398, : k
## = 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 73415, 17335, 7, 48398, : k
## = 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 48398, : k =
## 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 48398, : k =
## 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 73415, 17335, 7, 48398, : k
## = 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 73415, 17335, 7, 48398, : k
## = 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(11, 74461, 73415, 17335, 7, 61500, :
## k = 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(11, 74461, 73415, 17335, 7, 61500, :
## k = 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 48398, : k =
## 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 48398, : k =
## 300 exceeds number 234 of patterns
```

```
## Warning in knnregTrain(train = structure(c(74461, 73415, 17335, 7, 70692, : k =
## 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(74461, 73415, 17335, 7, 70692, : k =
## 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 17335, : k =
## 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 17335, : k =
## 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 7, 48398, 70692, 5, : k =
## 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 7, 48398, 70692, 5, : k =
## 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 17335, : k =
## 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 17335, : k =
## 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 73415, 17335, 7, 48398, : k
## = 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 73415, 17335, 7, 48398, : k
## = 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 17335, 7, 61500, : k
## = 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 17335, 7, 61500, : k
## = 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 17335, 48398, 61500, : k =
## 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 17335, 48398, 61500, : k =
## 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 17335, : k =
## 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 73415, 17335, : k =
## 300 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 17335, 7, 48398, : k
## = 250 exceeds number 234 of patterns

## Warning in knnregTrain(train = structure(c(106, 11, 74461, 17335, 7, 48398, : k
## = 300 exceeds number 234 of patterns
```

```r
df65_out = arrange(df65_out, k)

ggplot(df65_out) + geom_boxplot(aes(x=factor(k), y=rmse)) + theme_bw(base_size=7)
```



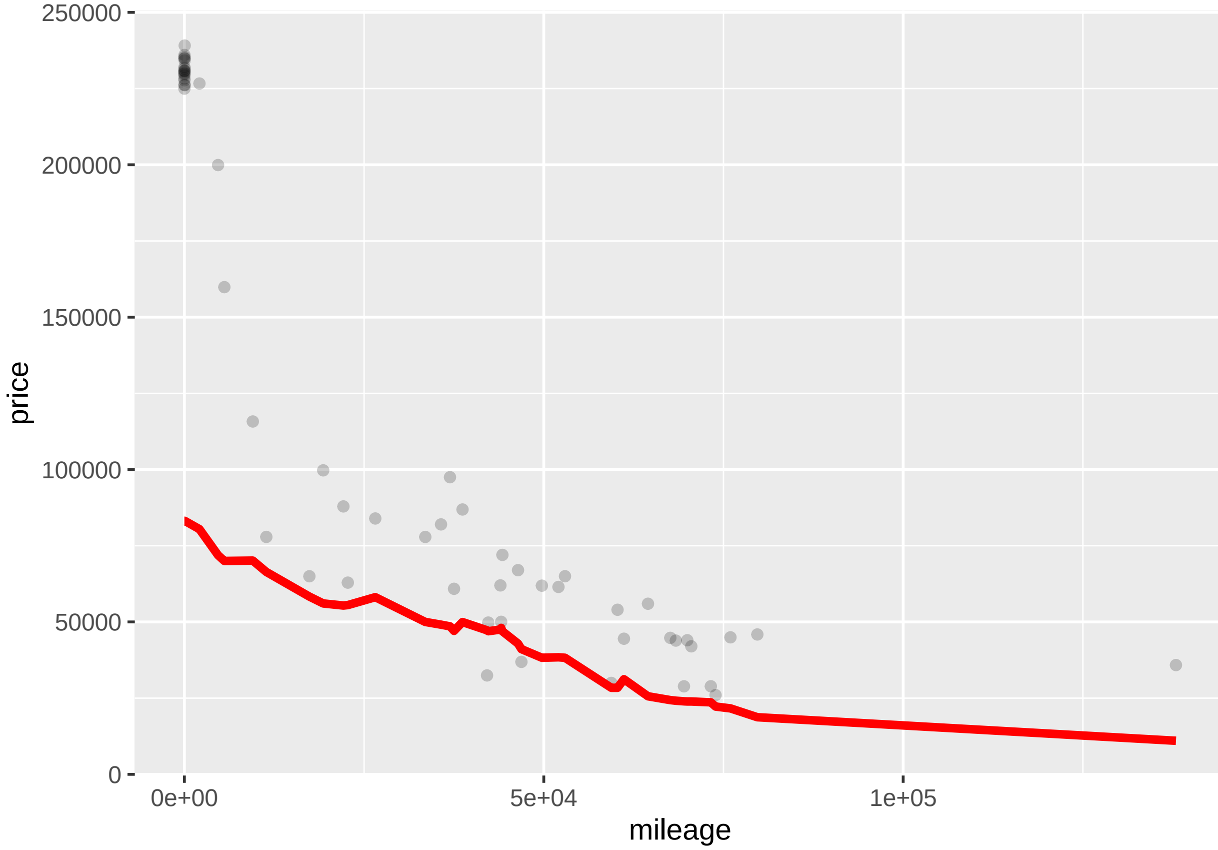

```
df65_split = initial_split(df65, prop = .8)
  df65_train = training(df65_split)
  df65_test = testing(df65_split)

predicted65 <- knnreg(price ~ mileage, data = df350_train, k=20)
rmse(predicted65, df65_test)
```

```
## [1] 89274.69
```

```
df65_split = initial_split(df65, prop = .8)
  df65_train = training(df65_split)
  df65_test = testing(df65_split)

knn20 = knnreg(price ~ mileage, data = df350_train, k=20)
rmse(knn20, df65_test)
```

```
## [1] 91254.64
```

```
df65_test = df65_test%>%
  mutate(price_pred65 = predict(knn20, df65_test))

t_test = ggplot(data = df65_test) +
  geom_point(mapping = aes(x=mileage, y = price), alpha = .2)

t_test + geom_line(aes(x=mileage, y=price_pred65), color='red', size = 1.5)
```



So for trim == 350 optimal k was equal to 15. For trim == 65 AMG, optimal k was equal to 20. I'm pretty sure it has something to do with the sample size and the fact that the 350 data set has more observations. Because we want to minimize the RMSE and that's dependent on the value of M on the slides.