Assignment 1

Maria Gilbert

2/8/2021

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(FNN)
library(class)
```

```
##
## Attaching package: 'class'

## The following objects are masked from 'package:FNN':
##
##     knn, knn.cv
```

```
library(rsample)
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

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

```
library(modelr)
library(parallel)
library(foreach)
```

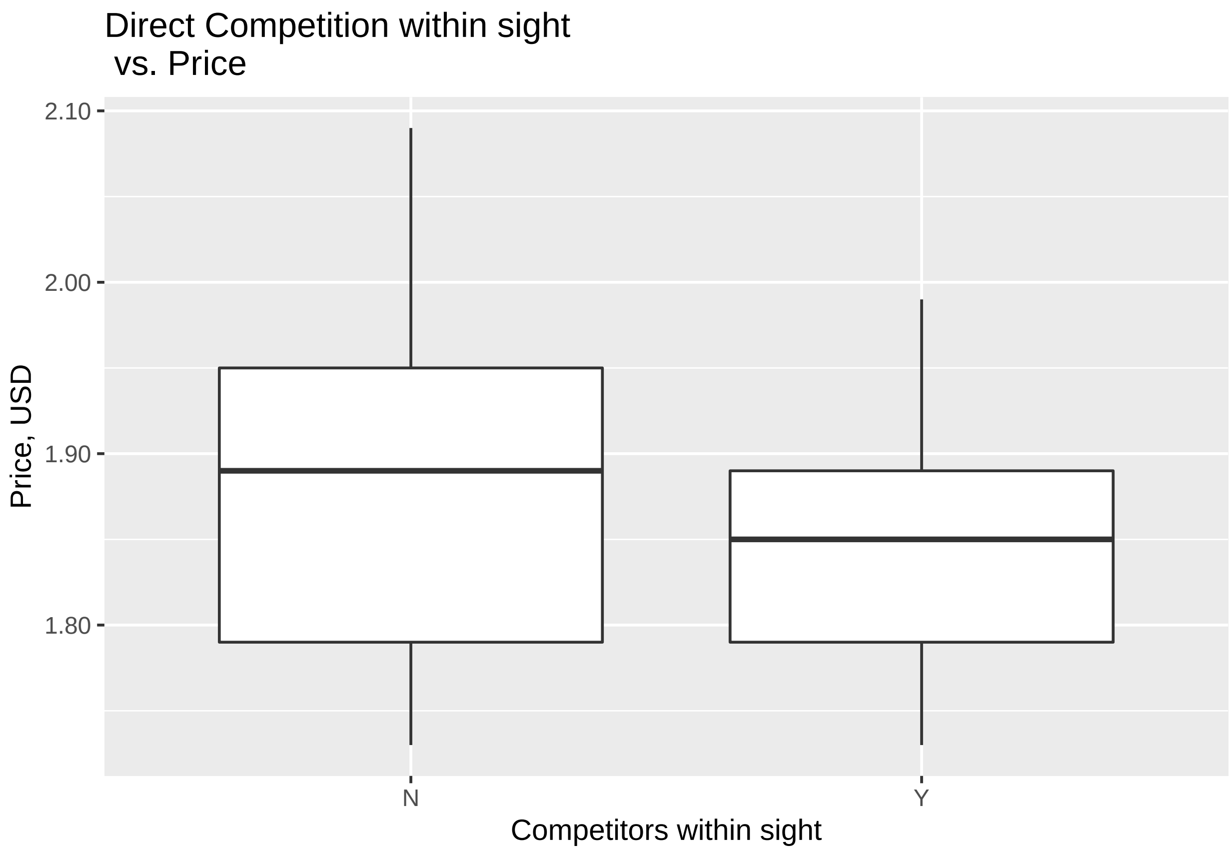
```
##
## Attaching package: 'foreach'

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

```
GasPrices <- read.csv("/Volumes/G-DRIVE mobile USB-C/GasPrices.csv")
bikeshare <- read.csv("/Volumes/G-DRIVE mobile USB-C/bikeshare.csv")
ABIA <- read.csv("/Volumes/G-DRIVE mobile USB-C/ABIA.csv")
sclass <- read.csv("/Volumes/G-DRIVE mobile USB-C/sclass.csv")
```

Data Visualization: gas prices

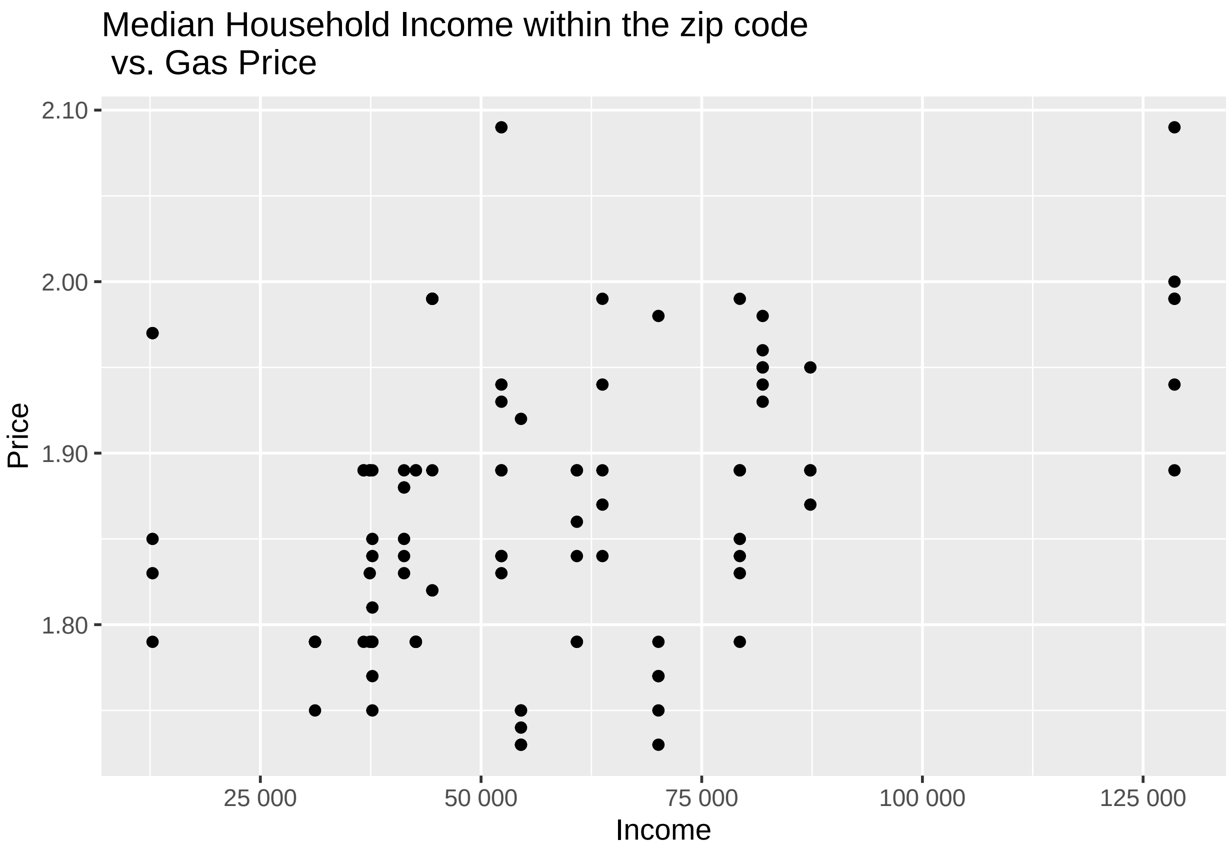
A) Gas stations charge more if they lack direct competition in sight (boxplot).

```
ggplot(data=GasPrices)+geom_boxplot(aes(x=factor(Competitors),
    y=Price))+ggtitle("Direct Competition within sight \n vs. Price")+scale_y_continuous(labels=scales:
```



B) The richer the area, the higher the gas price (scatter plot).

```
ggplot(data=GasPrices)+geom_point(mapping=aes(x=Income,y=Price))+scale_y_continuous(labels=scales::numbe
```

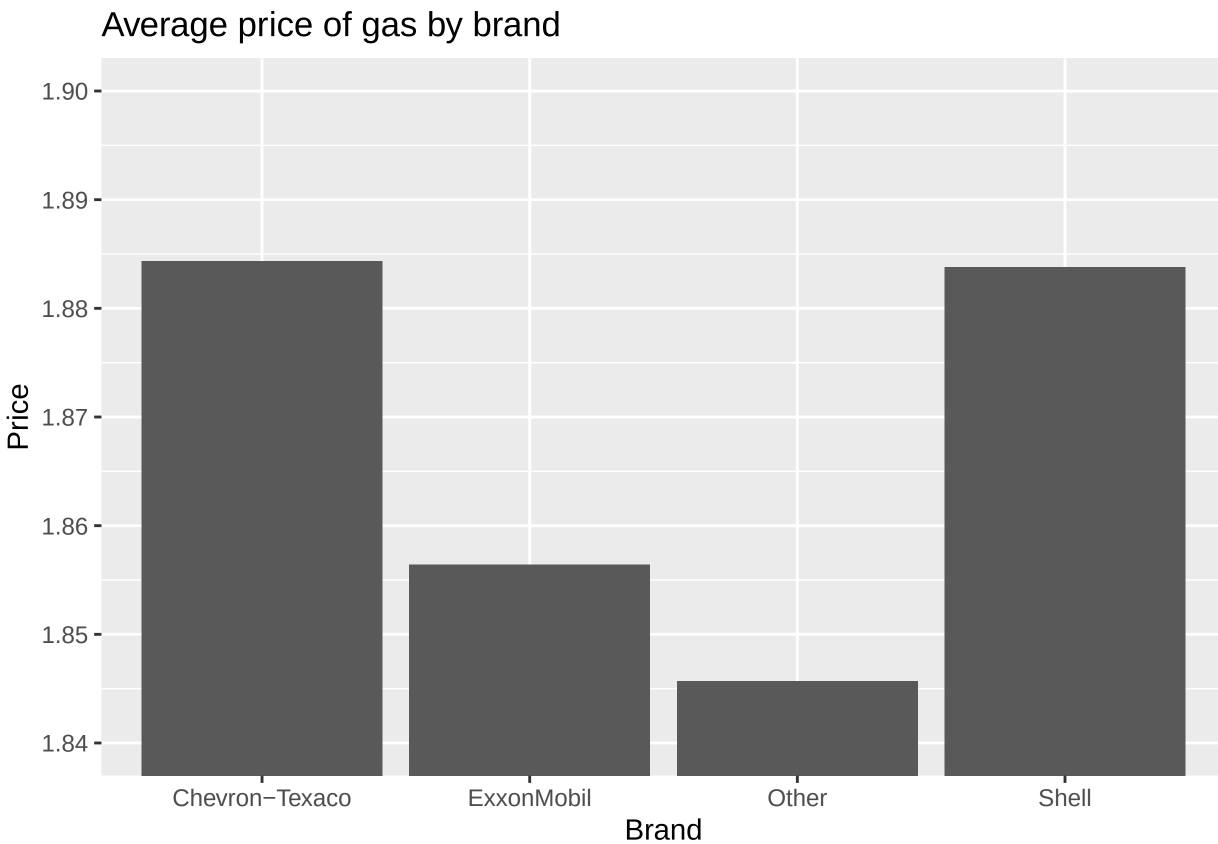


C) Shell charges more than other brands (bar plot).

```
ggplot(data=GasPrices,aes(x=Brand,y=Price))+geom_bar(stat="summary",fun.y="mean")+scale_y_continuous(la
```

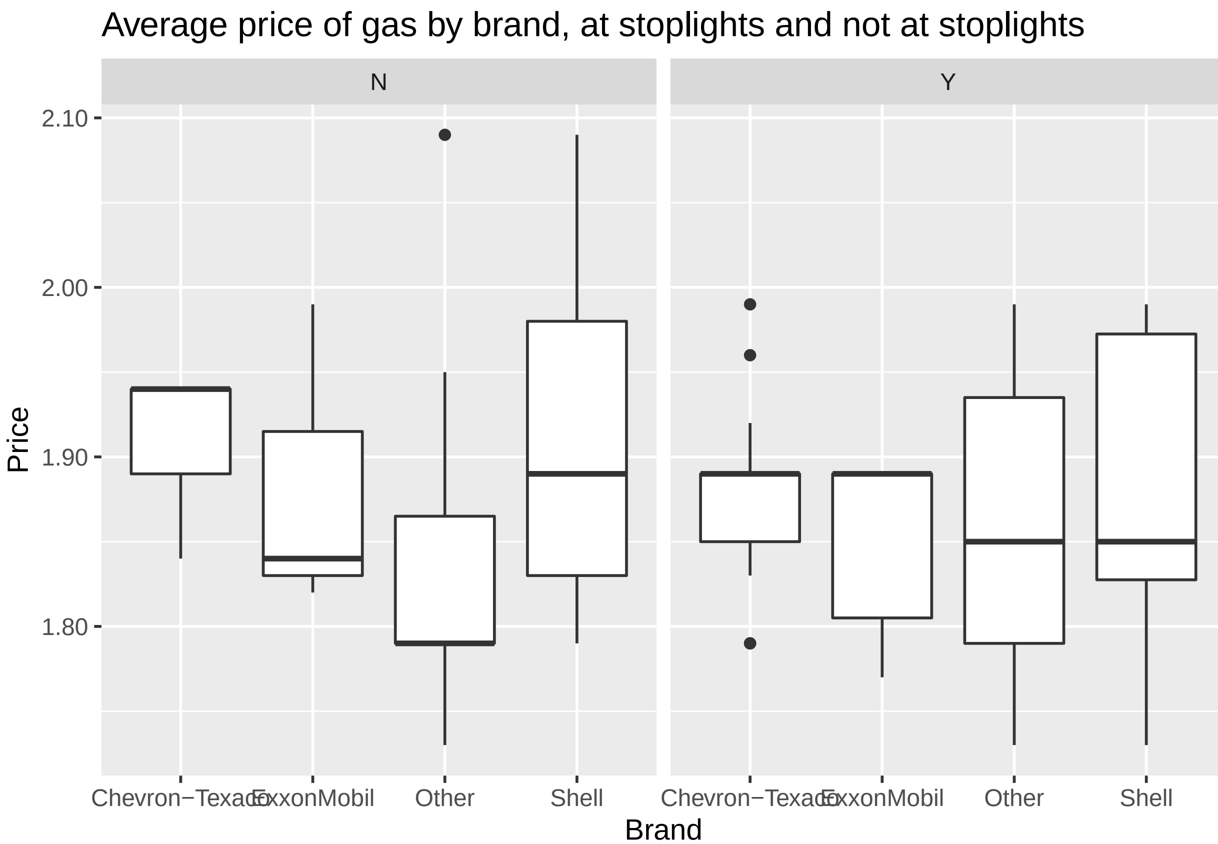
```
## Warning: Ignoring unknown parameters: fun.y
```

```
## No summary function supplied, defaulting to `mean_se()`
```



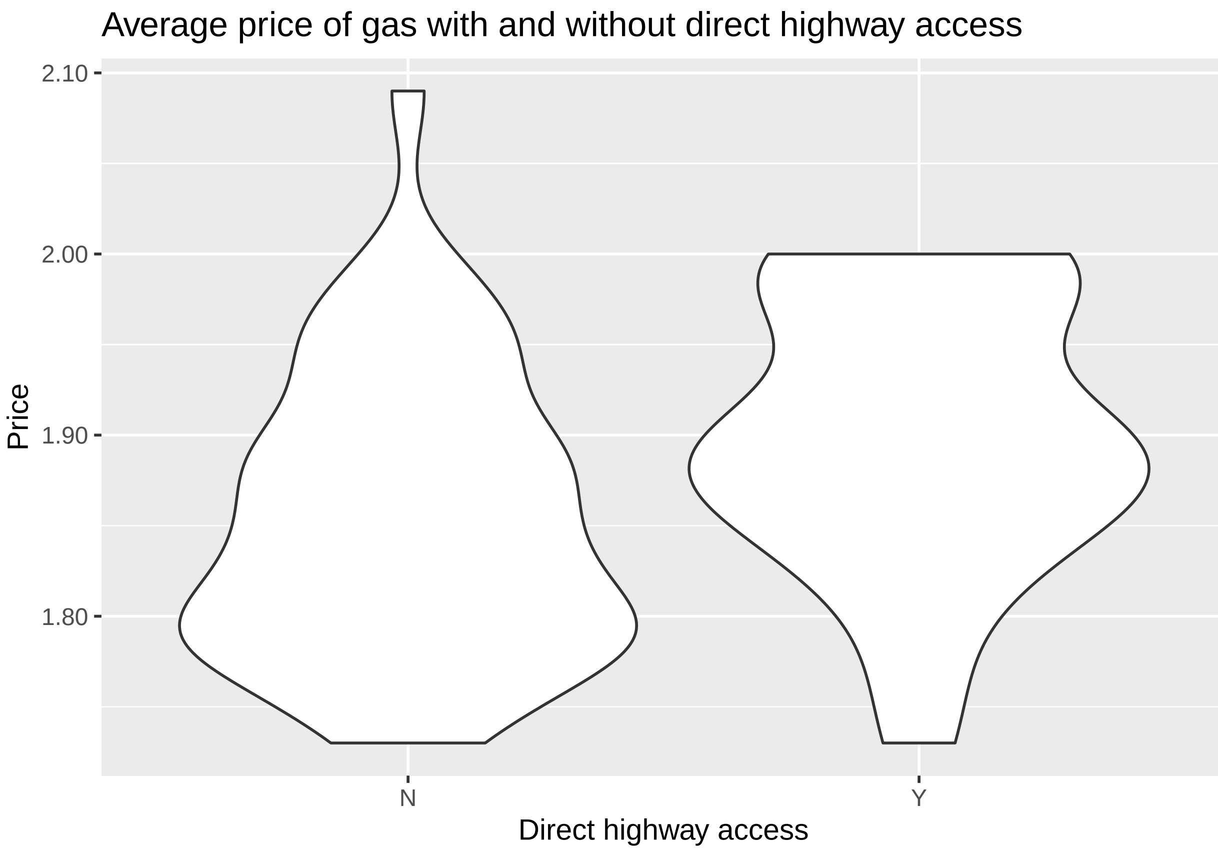
D) Gas stations at stoplights charge more (faceted histogram).

```
ggplot(data=GasPrices)+geom_boxplot(aes(x=Brand,y=Price))+facet_wrap(~Stoplight)+ggtitle("Average price
```



E) Gas stations with direct highway access charge more (your choice of plot).

```
ggplot(data=GasPrices)+geom_violin(aes(x=factor(Highway),y=Price))+scale_y_continuous(labels=scales::nu
```

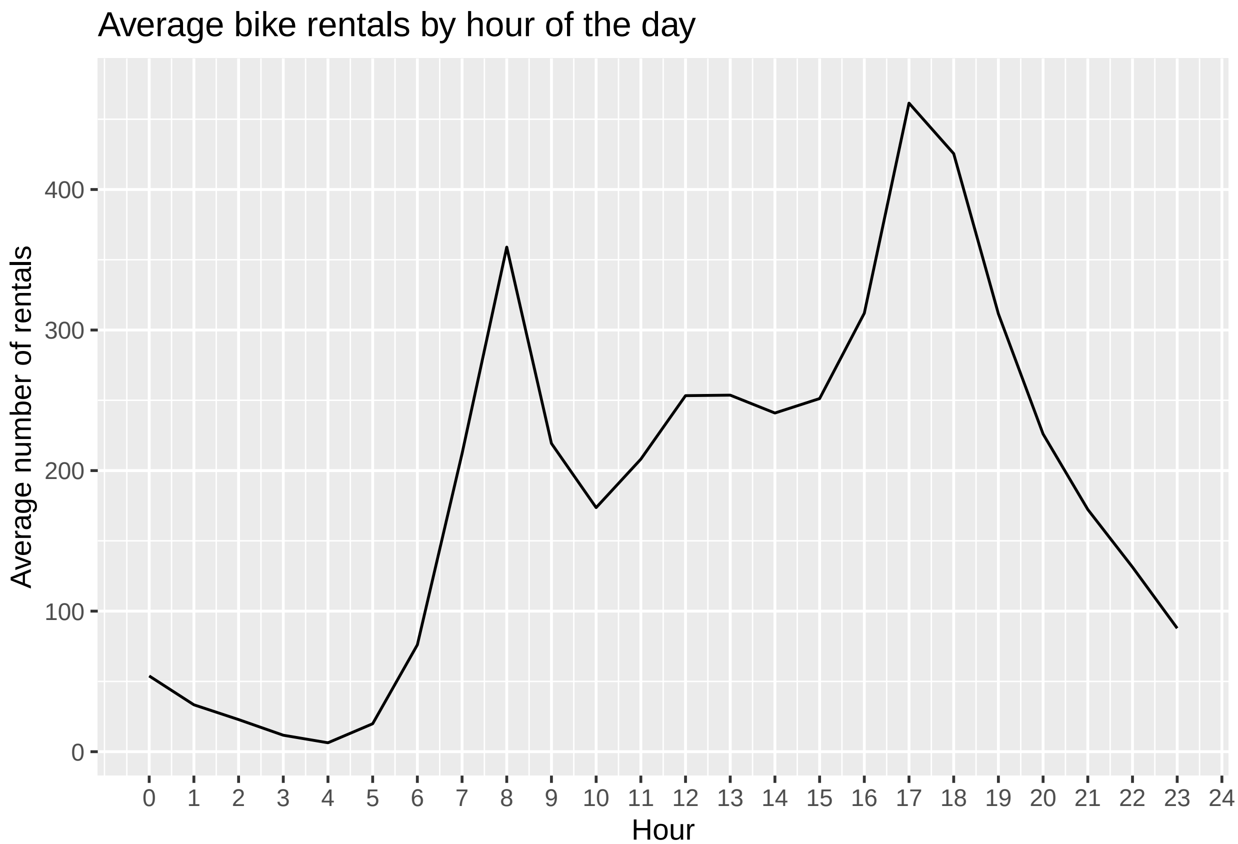


Data visualization: a bike share network

Plot A: a line graph showing average bike rentals versus hour of the day.

```
ggplot(data=bikeshare)+geom_line(aes(hr,total),stat="summary",fun.y="mean")+scale_x_continuous(breaks=0
```

```
## Warning: Ignoring unknown parameters: fun.y

## No summary function supplied, defaulting to `mean_se()`
```

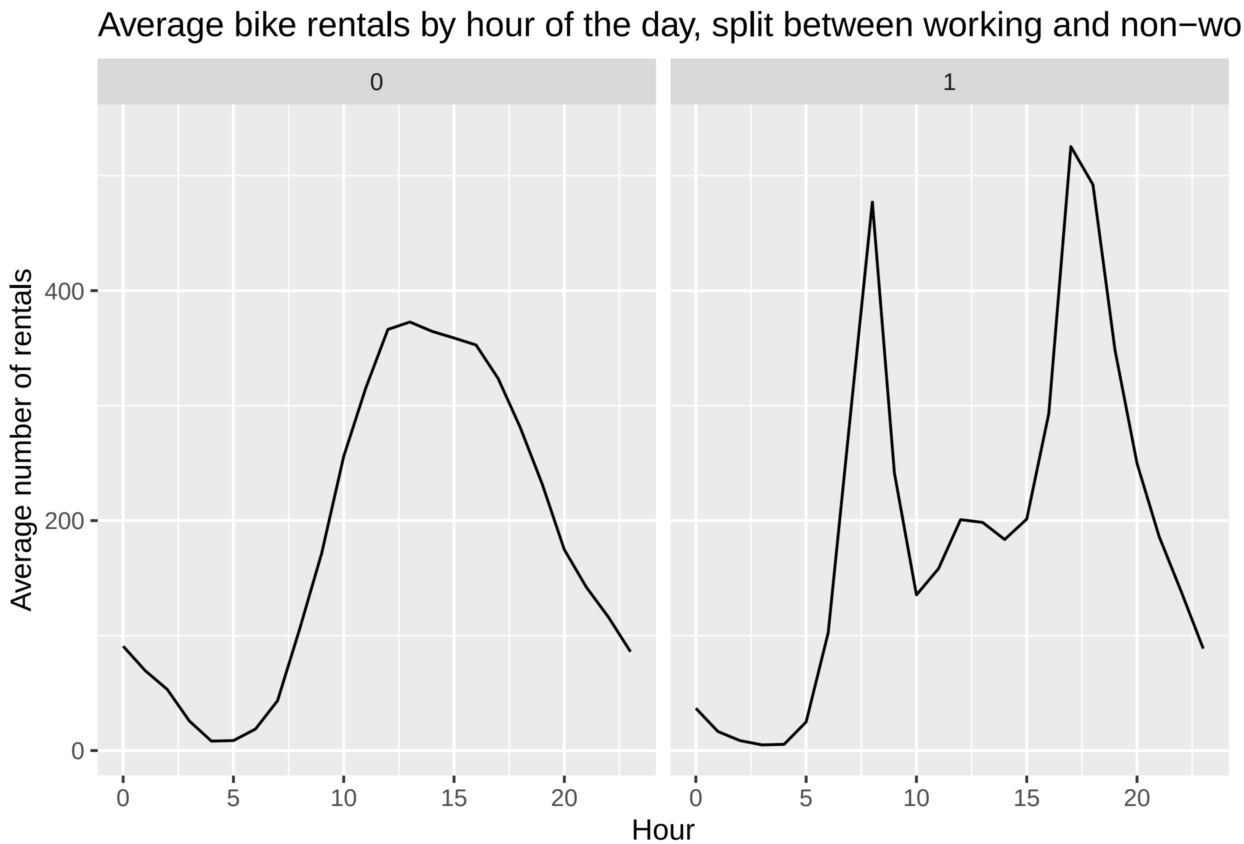


Plot B: a faceted line graph showing average bike rentals versus hour of the day, faceted according to whether it is a working day.

```
ggplot(data=bikeshare)+geom_line(aes(hr,total),stat="summary",fun.y="mean")+facet_wrap(~workingday)+ggt
```

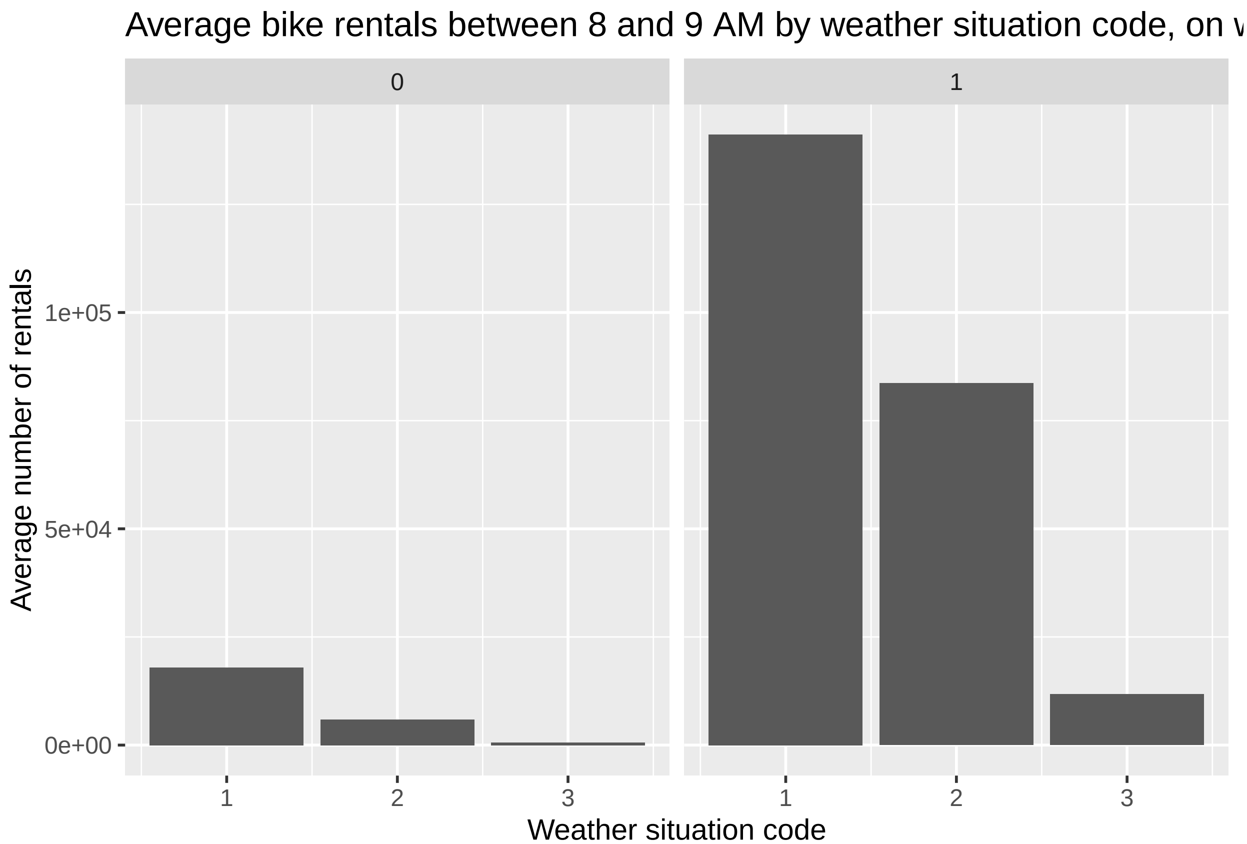
```
## Warning: Ignoring unknown parameters: fun.y
```

```
## No summary function supplied, defaulting to `mean_se()`
## No summary function supplied, defaulting to `mean_se()`
```



Plot C: a faceted bar plot showing average ridership during the 8 AM hour by weather situation code, faceted according to whether it is a working day or not.

```
bikeshare %>%
  filter(hr=="8") %>% ggplot()+geom_bar(aes(x=weathersit,y=total),stat="identity")+facet_wrap(~workingda
```

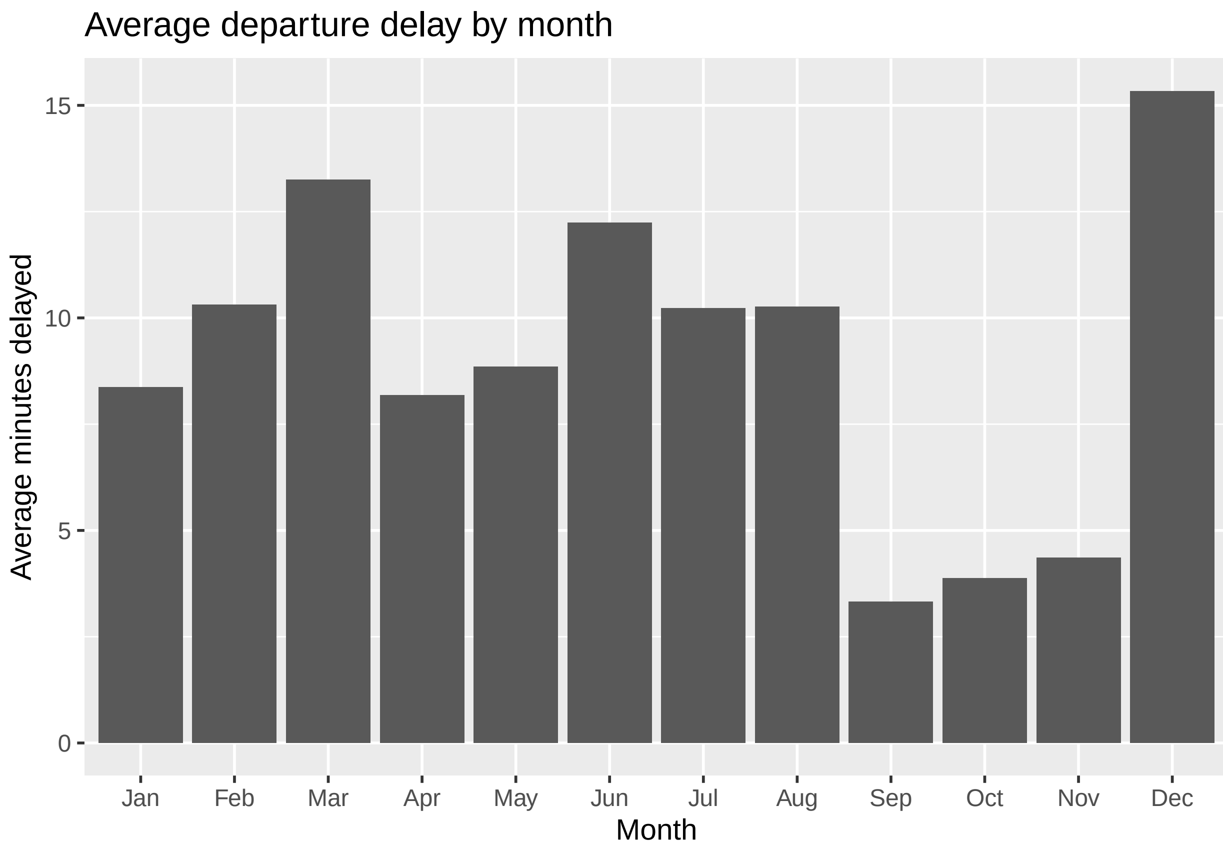


Data visualization: flights at ABIA

I would like to examine the difference in flight delays between different months, times of day, and carriers. With this information I could choose which month to travel, what time I should try to book my flight for, and which carrier to book my flight with, if my goal is to avoid all kinds of delays.

```
ggplot(data=ABIA,aes(x=factor(Month),y=DepDelay))+geom_bar(stat="summary",fun.y="mean")+scale_x_discrete
```

```
## Warning: Ignoring unknown parameters: fun.y

## Warning: Removed 1413 rows containing non-finite values (stat_summary).

## No summary function supplied, defaulting to `mean_se()`
```



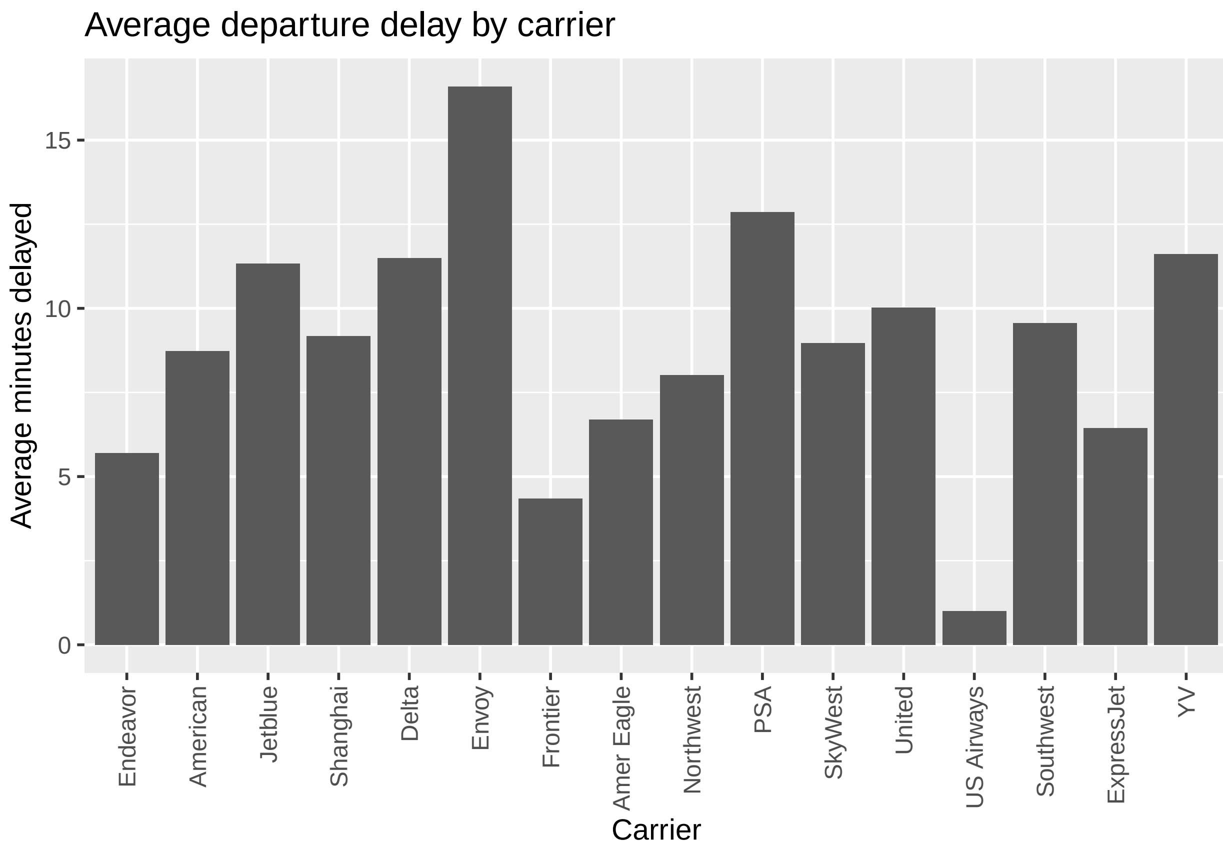
Next, I will see which carriers have the shortest and longest average departure delays.

```
ggplot(data=ABIA,aes(x=factor(UniqueCarrier),y=DepDelay))+geom_bar(stat="summary",fun.y="mean")+ggtitle
```

```
## Warning: Ignoring unknown parameters: fun.y
```

```
## Warning: Removed 1413 rows containing non-finite values (stat_summary).
```

```
## No summary function supplied, defaulting to `mean_se()`
```



K-nearest neighbors

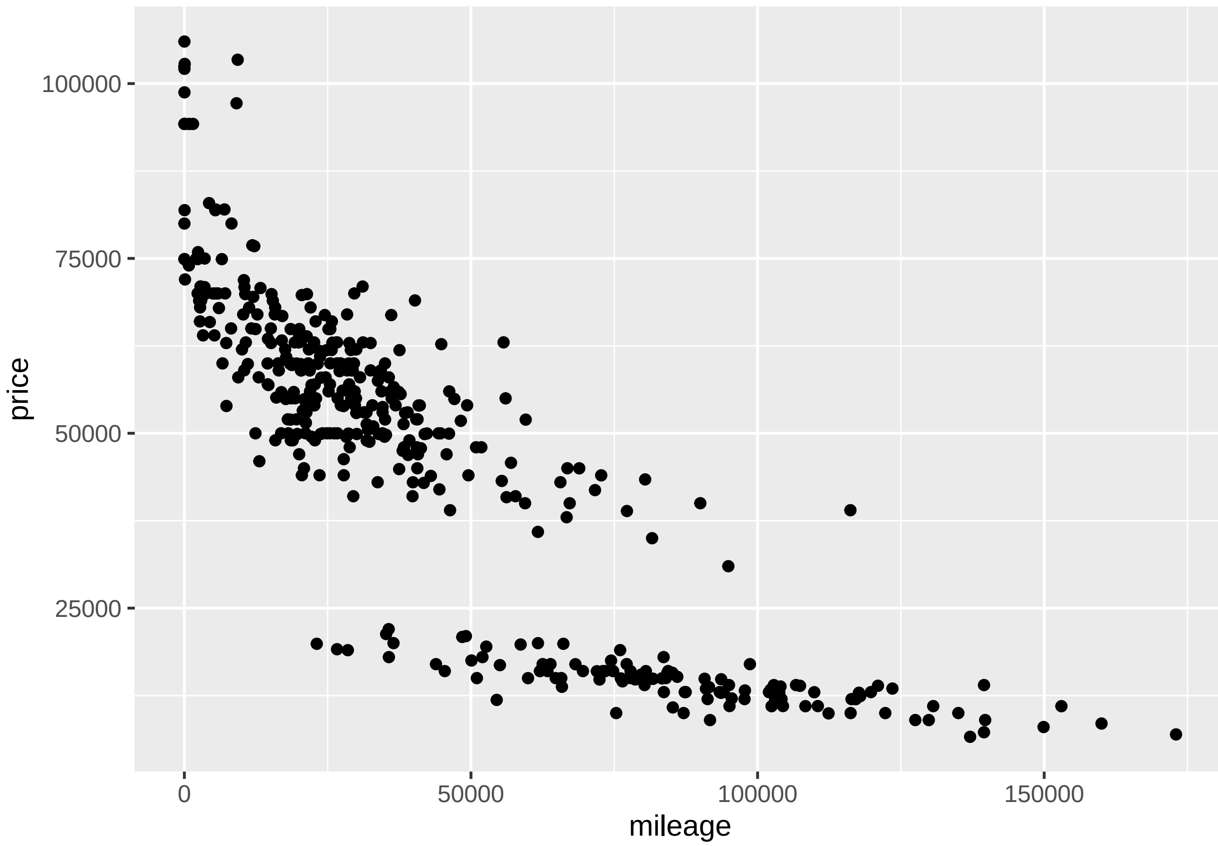
Splitting the 350 AMG data and the 63 AMG data.

```
threefifty <- sclass[sclass$trim=="350",]
sixtyfive <- sclass[sclass$trim=="65 AMG",]
```

350 AMG

Splitting the data into a training and a testing set

```
ggplot(data=threefifty)+geom_point(mapping=aes(x=mileage,y=price))
```




```
threefifty_split = initial_split(threefifty,prop=0.9)
threefifty_train = training(threefifty_split)
threefifty_test = testing(threefifty_split)
```

Running K-nearest-neighbors, from K=2 to K=25

```
knn2 = knnreg(price~mileage,data=threefifty_train,k=2)
knn3 = knnreg(price~mileage,data=threefifty_train,k=3)
knn4 = knnreg(price~mileage,data=threefifty_train,k=4)
knn5 = knnreg(price~mileage,data=threefifty_train,k=5)
knn6 = knnreg(price~mileage,data=threefifty_train,k=6)
knn7 = knnreg(price~mileage,data=threefifty_train,k=7)
knn8 = knnreg(price~mileage,data=threefifty_train,k=8)
knn9 = knnreg(price~mileage,data=threefifty_train,k=9)
knn10 = knnreg(price~mileage,data=threefifty_train,k=10)
knn11 = knnreg(price~mileage,data=threefifty_train,k=11)
knn12 = knnreg(price~mileage,data=threefifty_train,k=12)
knn13 = knnreg(price~mileage,data=threefifty_train,k=13)
knn14 = knnreg(price~mileage,data=threefifty_train,k=14)
knn15 = knnreg(price~mileage,data=threefifty_train,k=15)
knn16 = knnreg(price~mileage,data=threefifty_train,k=16)
knn17 = knnreg(price~mileage,data=threefifty_train,k=17)
knn18 = knnreg(price~mileage,data=threefifty_train,k=18)
knn19 = knnreg(price~mileage,data=threefifty_train,k=19)
knn20 = knnreg(price~mileage,data=threefifty_train,k=20)
```

```r
knn21 = knnreg(price~mileage,data=threefifty_train,k=21)
knn22 = knnreg(price~mileage,data=threefifty_train,k=22)
knn23 = knnreg(price~mileage,data=threefifty_train,k=23)
knn24 = knnreg(price~mileage,data=threefifty_train,k=24)
knn25 = knnreg(price~mileage,data=threefifty_train,k=25)
```

Calculating the out-of-sample root mean-squared error (RMSE) for each value of k

```r
rmse2 = rmse(knn2,threefifty_test)
rmse3 = rmse(knn3,threefifty_test)
rmse4 = rmse(knn4,threefifty_test)
rmse5 = rmse(knn5,threefifty_test)
rmse6 = rmse(knn6,threefifty_test)
rmse7 = rmse(knn7,threefifty_test)
rmse8 = rmse(knn8,threefifty_test)
rmse9 = rmse(knn9,threefifty_test)
rmse10 = rmse(knn10,threefifty_test)
rmse11 = rmse(knn11,threefifty_test)
rmse12 = rmse(knn12,threefifty_test)
rmse13 = rmse(knn13,threefifty_test)
rmse14 = rmse(knn14,threefifty_test)
rmse15 = rmse(knn15,threefifty_test)
rmse16 = rmse(knn16,threefifty_test)
rmse17 = rmse(knn17,threefifty_test)
rmse18 = rmse(knn18,threefifty_test)
rmse19 = rmse(knn19,threefifty_test)
rmse20 = rmse(knn20,threefifty_test)
rmse21 = rmse(knn21,threefifty_test)
rmse22 = rmse(knn22,threefifty_test)
rmse23 = rmse(knn23,threefifty_test)
rmse24 = rmse(knn24,threefifty_test)
rmse25 = rmse(knn25,threefifty_test)
```

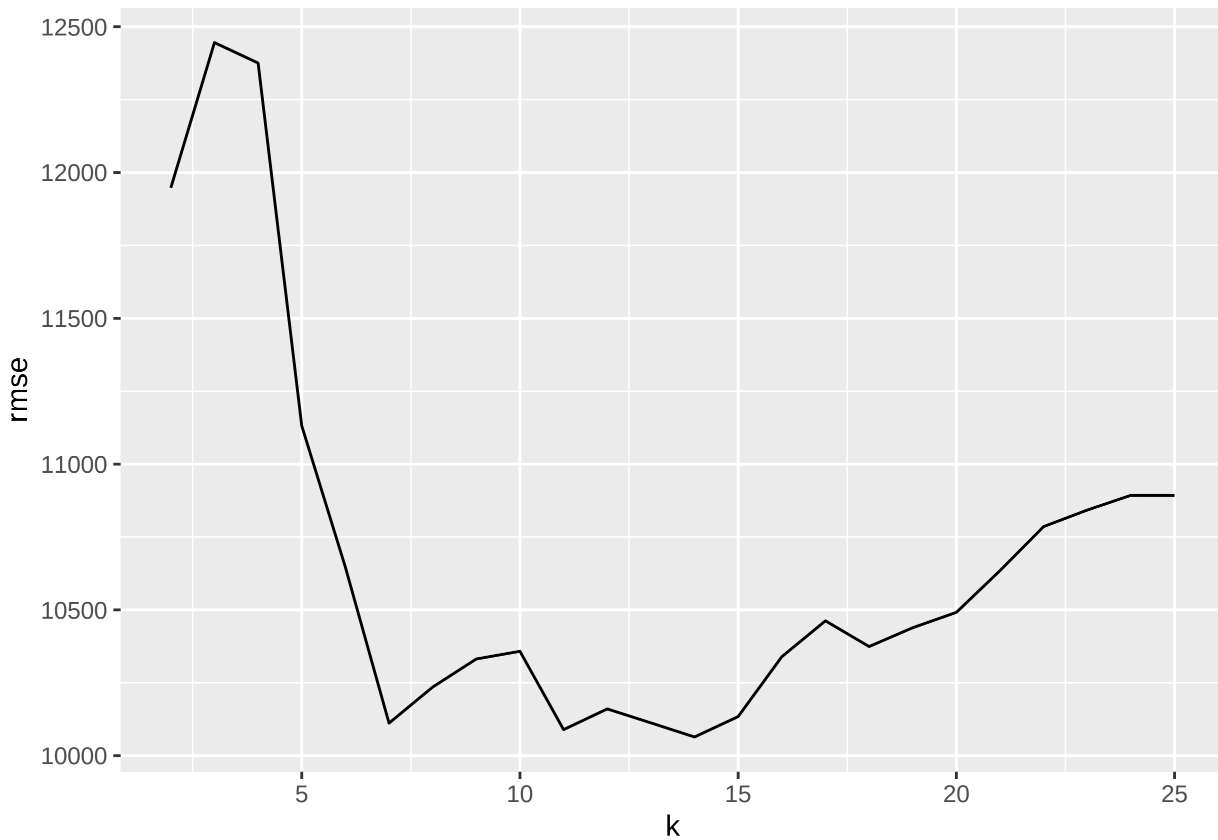
Plotting RMSE versus K

```r
k <- c(2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25)
rmse <- c(rmse2,rmse3,rmse4,rmse5,rmse6,rmse7,rmse8,rmse9,rmse10,rmse11,rmse12,rmse13,rmse14,rmse15,rmse
errors <- data.frame(k,rmse)
errors
```

```
##     k     rmse
## 1   2 11947.43
## 2   3 12445.49
## 3   4 12375.27
## 4   5 11131.40
## 5   6 10646.19
## 6   7 10111.60
## 7   8 10235.23
## 8   9 10331.56
## 9  10 10357.92
## 10 11 10089.25
## 11 12 10160.32
## 12 13 10112.53
```

```
## 13 14 10064.03
## 14 15 10134.03
## 15 16 10339.22
## 16 17 10462.43
## 17 18 10374.28
## 18 19 10438.77
## 19 20 10491.47
## 20 21 10634.54
## 21 22 10785.66
## 22 23 10842.32
## 23 24 10892.94
## 24 25 10892.73
```

```
ggplot(data=errors)+geom_line(aes(k,rmse))
```

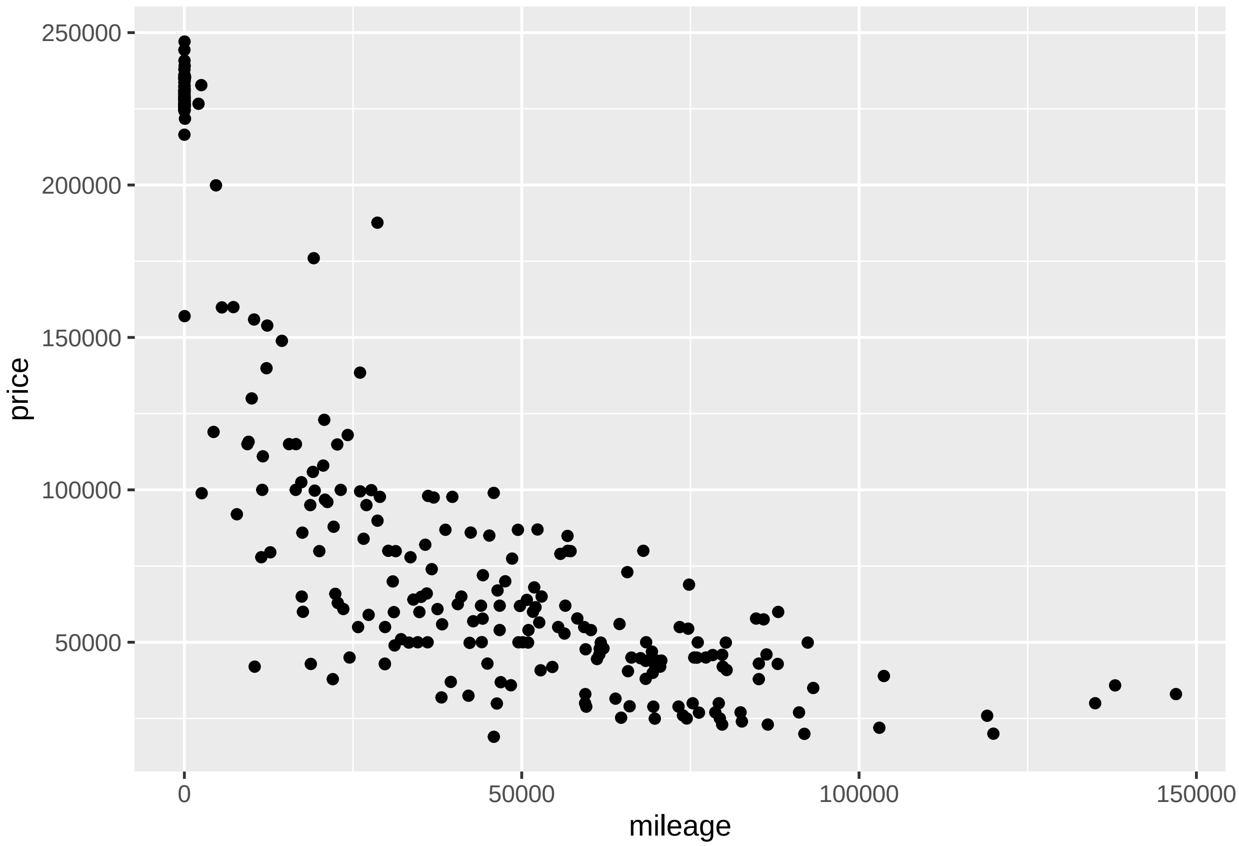


It looks like the optimal value of K is 19.

65 AMG

Splitting the data into a training and a testing set

```
ggplot(data=sixtyfive)+geom_point(mapping=aes(x=mileage,y=price))
```




```
sixtyfive_split = initial_split(sixtyfive,prop=0.9)
sixtyfive_train = training(sixtyfive_split)
sixtyfive_test = testing(sixtyfive_split)
```

Running K-nearest-neighbors, from K=2 to K=100

```
knn2 = knnreg(price~mileage,data=sixtyfive_train,k=2)
knn3 = knnreg(price~mileage,data=sixtyfive_train,k=3)
knn4 = knnreg(price~mileage,data=sixtyfive_train,k=4)
knn5 = knnreg(price~mileage,data=sixtyfive_train,k=5)
knn6 = knnreg(price~mileage,data=sixtyfive_train,k=6)
knn7 = knnreg(price~mileage,data=sixtyfive_train,k=7)
knn8 = knnreg(price~mileage,data=sixtyfive_train,k=8)
knn9 = knnreg(price~mileage,data=sixtyfive_train,k=9)
knn10 = knnreg(price~mileage,data=sixtyfive_train,k=10)
knn11 = knnreg(price~mileage,data=sixtyfive_train,k=11)
knn12 = knnreg(price~mileage,data=sixtyfive_train,k=12)
knn13 = knnreg(price~mileage,data=sixtyfive_train,k=13)
knn14 = knnreg(price~mileage,data=sixtyfive_train,k=14)
knn15 = knnreg(price~mileage,data=sixtyfive_train,k=15)
knn16 = knnreg(price~mileage,data=sixtyfive_train,k=16)
knn17 = knnreg(price~mileage,data=sixtyfive_train,k=17)
knn18 = knnreg(price~mileage,data=sixtyfive_train,k=18)
knn19 = knnreg(price~mileage,data=sixtyfive_train,k=19)
knn20 = knnreg(price~mileage,data=sixtyfive_train,k=20)
knn21 = knnreg(price~mileage,data=sixtyfive_train,k=21)
knn22 = knnreg(price~mileage,data=sixtyfive_train,k=22)
```

```
knn23 = knnreg(price~mileage,data=sixtyfive_train,k=23)
knn24 = knnreg(price~mileage,data=sixtyfive_train,k=24)
knn25 = knnreg(price~mileage,data=sixtyfive_train,k=25)
```

Calculating the out-of-sample root mean-squared error (RMSE) for each value of k

```
rmse2 = rmse(knn2,sixtyfive_test)
rmse3 = rmse(knn3,sixtyfive_test)
rmse4 = rmse(knn4,sixtyfive_test)
rmse5 = rmse(knn5,sixtyfive_test)
rmse6 = rmse(knn6,sixtyfive_test)
rmse7 = rmse(knn7,sixtyfive_test)
rmse8 = rmse(knn8,sixtyfive_test)
rmse9 = rmse(knn9,sixtyfive_test)
rmse10 = rmse(knn10,sixtyfive_test)
rmse11 = rmse(knn11,sixtyfive_test)
rmse12 = rmse(knn12,sixtyfive_test)
rmse13 = rmse(knn13,sixtyfive_test)
rmse14 = rmse(knn14,sixtyfive_test)
rmse15 = rmse(knn15,sixtyfive_test)
rmse16 = rmse(knn16,sixtyfive_test)
rmse17 = rmse(knn17,sixtyfive_test)
rmse18 = rmse(knn18,sixtyfive_test)
rmse19 = rmse(knn19,sixtyfive_test)
rmse20 = rmse(knn20,sixtyfive_test)
rmse21 = rmse(knn21,sixtyfive_test)
rmse22 = rmse(knn22,sixtyfive_test)
rmse23 = rmse(knn23,sixtyfive_test)
rmse24 = rmse(knn24,sixtyfive_test)
rmse25 = rmse(knn25,sixtyfive_test)
```

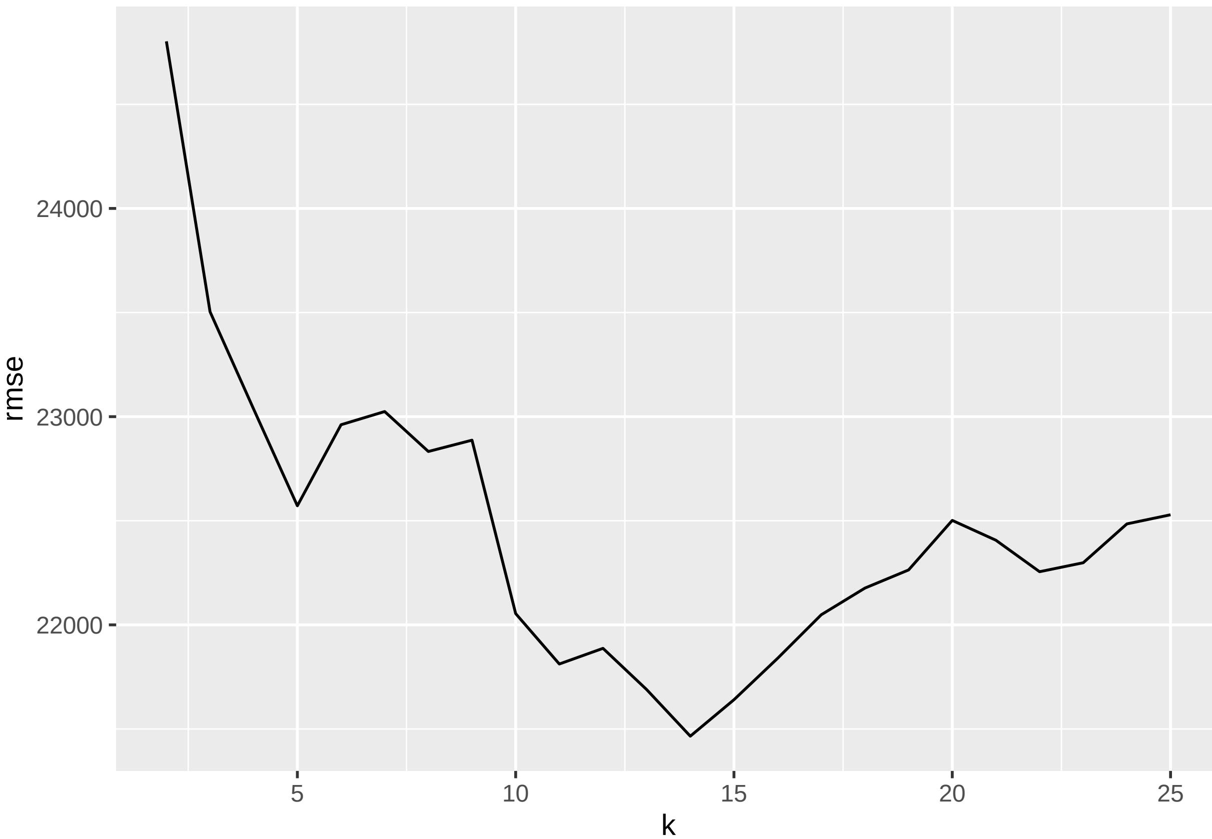
Plotting RMSE versus K

```
k <- c(2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25)
rmse <- c(rmse2,rmse3,rmse4,rmse5,rmse6,rmse7,rmse8,rmse9,rmse10,rmse11,rmse12,rmse13,rmse14,rmse15,rms
errors <- data.frame(k,rmse)
errors
```

```
##     k     rmse
## 1   2 24802.65
## 2   3 23503.90
## 3   4 23035.34
## 4   5 22571.96
## 5   6 22961.14
## 6   7 23024.52
## 7   8 22832.63
## 8   9 22886.96
## 9  10 22054.04
## 10 11 21811.97
## 11 12 21887.11
## 12 13 21689.23
## 13 14 21465.36
## 14 15 21640.36
```

```
## 15 16 21839.45
## 16 17 22048.61
## 17 18 22176.57
## 18 19 22263.75
## 19 20 22501.73
## 20 21 22406.36
## 21 22 22255.23
## 22 23 22298.31
## 23 24 22484.97
## 24 25 22528.73
```

```
ggplot(data=errors)+geom_line(aes(k,rmse))
```



It looks like the optimal value of K is 12.