Excercise1

Xiaohan Sun / Liyuan Zhang / Evelyn Cheng

2021/2/8

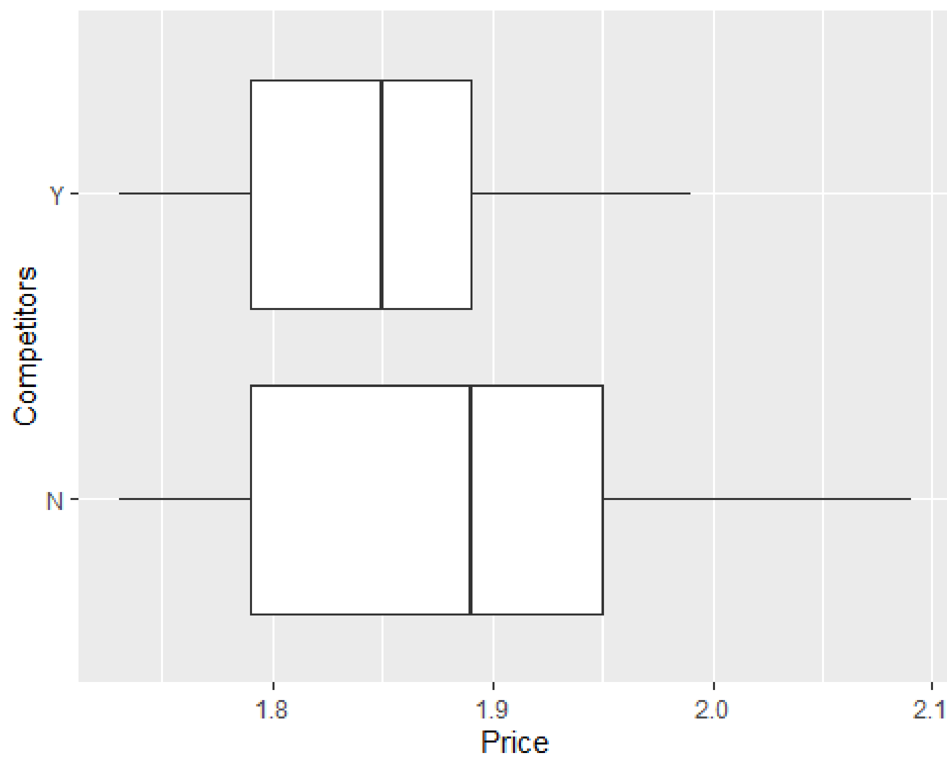
ECO 395M Homework 1: Xiaohan Sun / Liyuan Zhang / Evelyn Cheng

1) Data visualization: gas prices

```
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyve
rse 1.3.0 --

## √ ggplot2 3.3.3     √ purrr   0.3.4
## √ tibble  3.0.6     √ dplyr   1.0.3
## √ tidyr   1.1.2     √ stringr 1.4.0
## √ readr   1.4.0     √ forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_co
nflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(ggplot2)
GasPrices = read.csv('../data/GasPrices.csv')
head(GasPrices)

##   X ID     Name Price Pumps Interior Restaurant CarWash Highway Inte
rsection
## 1 1  1    Shell  1.79     4        Y          N       N       N
       Y
## 2 2  2   Valero  1.83     4        Y          N       N       N
       Y
## 3 3  3 7-Eleven  1.88     4        Y          N       N       N
       Y
## 4 4  4   Texaco  1.88     4        Y          N       Y       N
       Y
## 5 5  5    Shell  1.84     6        Y          N       N       N
       Y
## 6 6  6    Shell  1.83     8        Y          N       N       N
       Y
##   Stoplight IntersectionStoplight Gasolines Competitors Zipcode
## 1         N          Intersection         3           N   78705
## 2         N          Intersection         3           N   78705
```

```
## 3          Y                  Both         3            Y    78751
## 4          Y                  Both         4            Y    78751
## 5          Y                  Both         3            N    78751
## 6          N          Intersection         3            Y    78752
##                Address Income          Brand
## 1 3201 N Lamar Blvd  12786          Shell
## 2 3515 N Lamar Blvd  12786          Other
## 3 5101 N Lamar Blvd  41279          Other
## 4 5301 N Lamar Blvd  41279 Chevron-Texaco
## 5 5630 N Lamar Blvd  41279          Shell
## 6 6301 N Lamar Blvd  37396          Shell
```

```
ggplot(data=GasPrices) +
  geom_boxplot(aes(x=Price, y=Competitors))
```



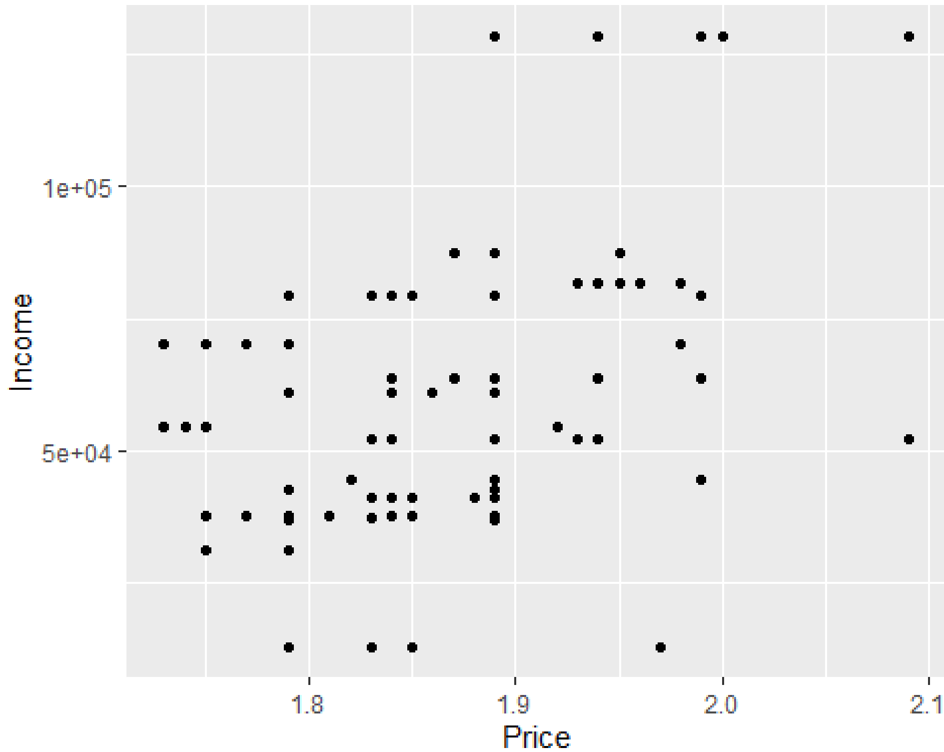
Claim: Gas stations charge more if they lack direct competition in sight.

Conclusion: As the graph shows, the median price for gas stations that lacking competitors is higher than the one having competitors. Also, upper edge and upper quartile price are higher than the one having competitors. So, this claim is correct.

```
summary(GasPrices$Income)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12786   37690   52306   56727   70095  128556
```

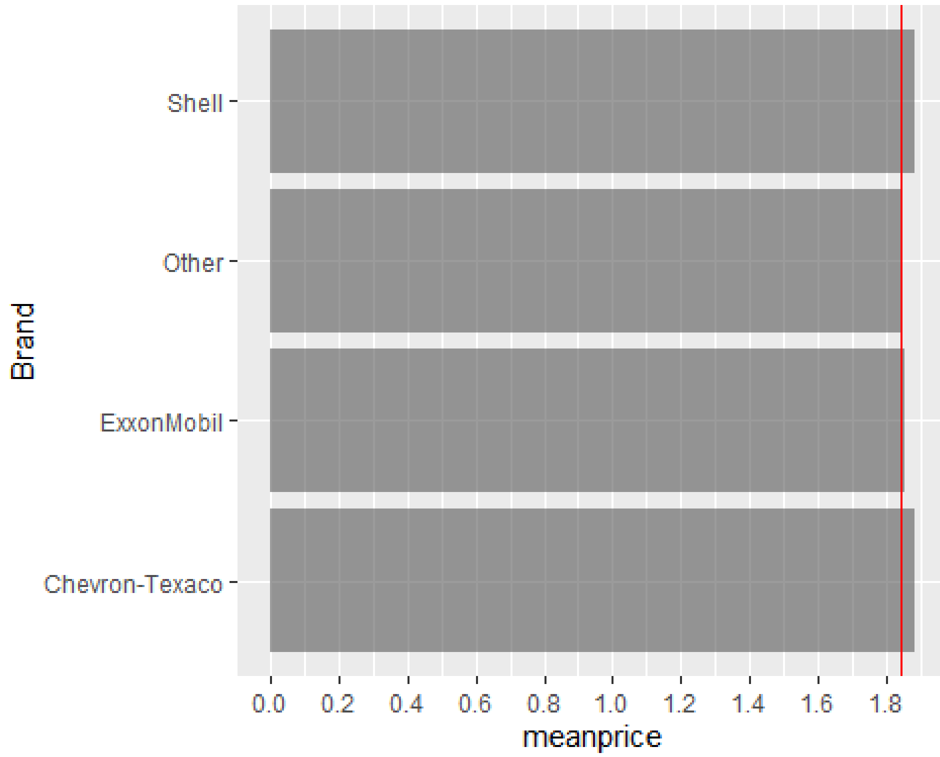
```
ggplot(data=GasPrices) +
  geom_point(aes(x=Price, y=Income))+
  ylim(12780,128560)
```



Claim: The richer the area, the higher the gas price.

Conclusion: As shown in the figure, when the income is higher, the dots will fall more on the right. The trend for this scatter plot is increasing. So, this claim is correct.

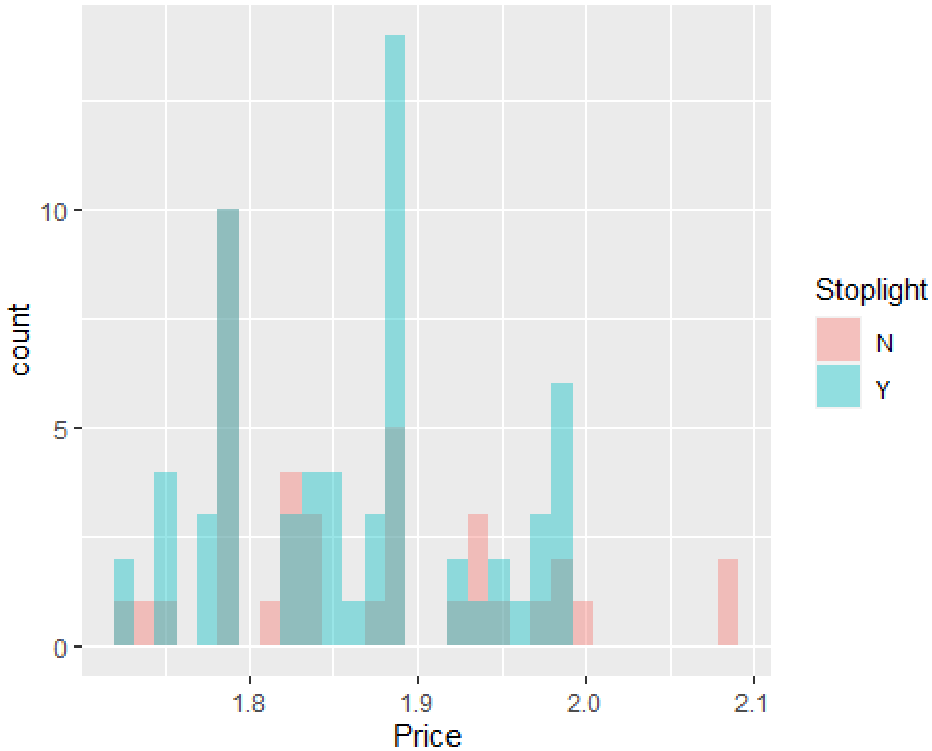
```
GasPricesC = GasPrices %>%
  group_by(Brand) %>%
  summarise(meanprice=mean(Price))
ggplot(data=GasPricesC) +
  geom_col(aes(x=meanprice, y=Brand), alpha=0.6) +
  scale_x_continuous(breaks=seq(0,2,0.2)) +
  geom_vline(xintercept=1.84,col="red")
```



Claim: Shell charges more than other brands.

Conclusion: In the bar chart, the bar of shell has higher price compared with Other brands. So, this claim is correct.

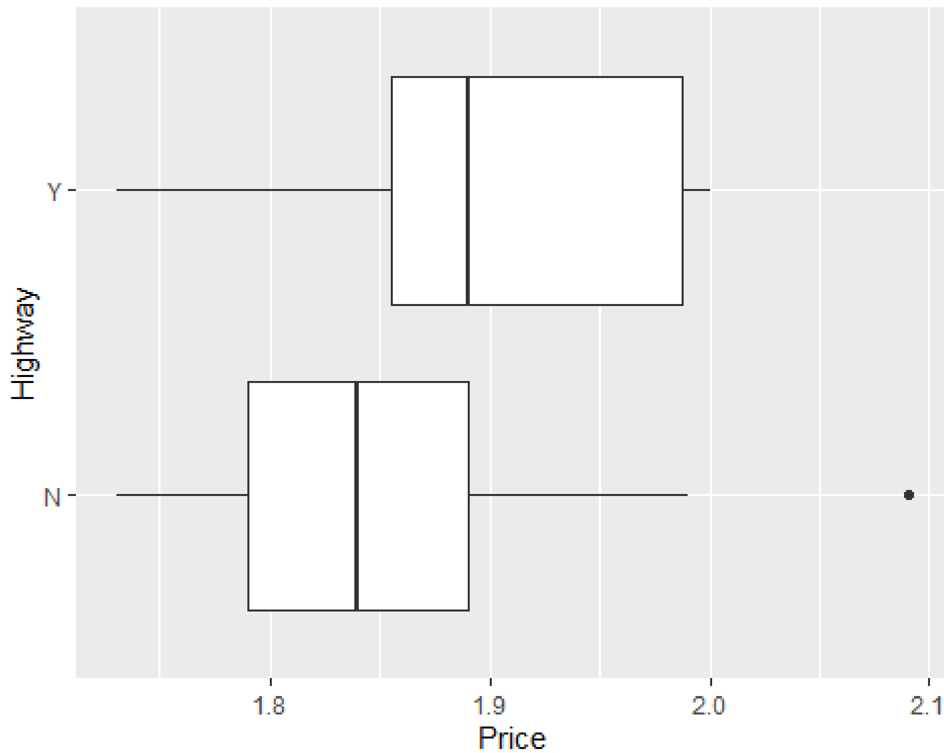
```
ggplot(data=GasPrices,aes(x=Price,fill=Stoplight))+geom_histogram(bins
= 30, alpha=0.4, position = "identity")
```



Claim: Gas stations at stoplights charge more.

Conclusion: The histogram present that the pick price for gas stations at stoplights (which is the blue one) is higher than the one doesn't (pink bars). So, this claim is correct.

```
ggplot(data=GasPrices) +
  geom_boxplot(aes(x=Price, y=Highway))
```

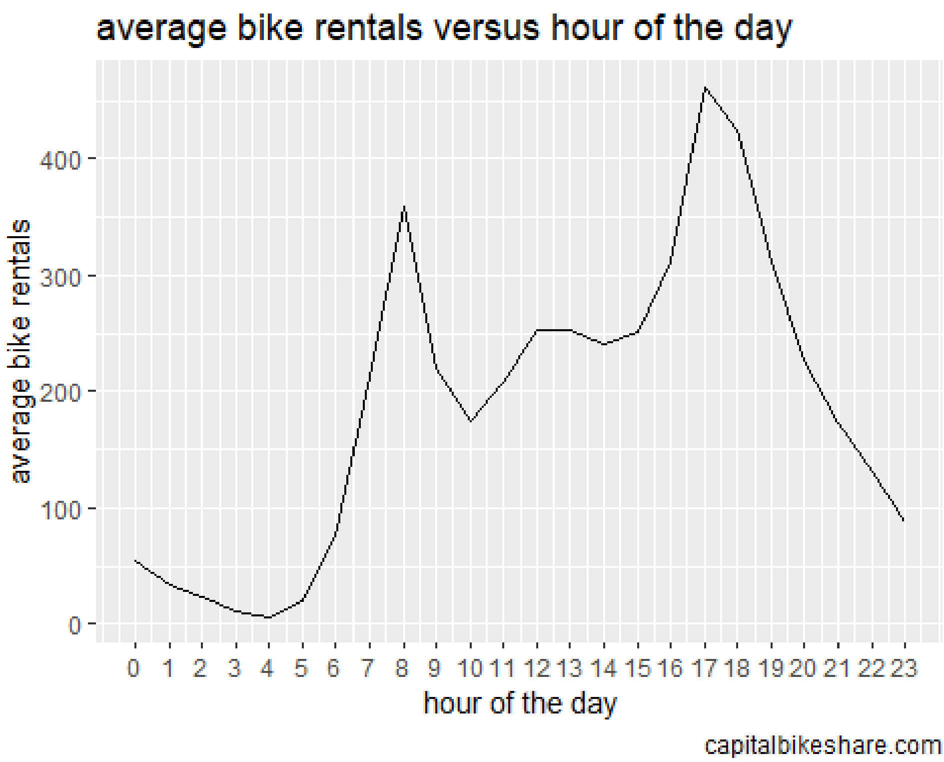


Claim: Gas stations with direct highway access charge more.

Conclusion: As the graph shows, the median price for gas stations with direct highway access is higher than the one without. Also, upper quartile price are higher than the one without direct highway access. So, this claim is correct.

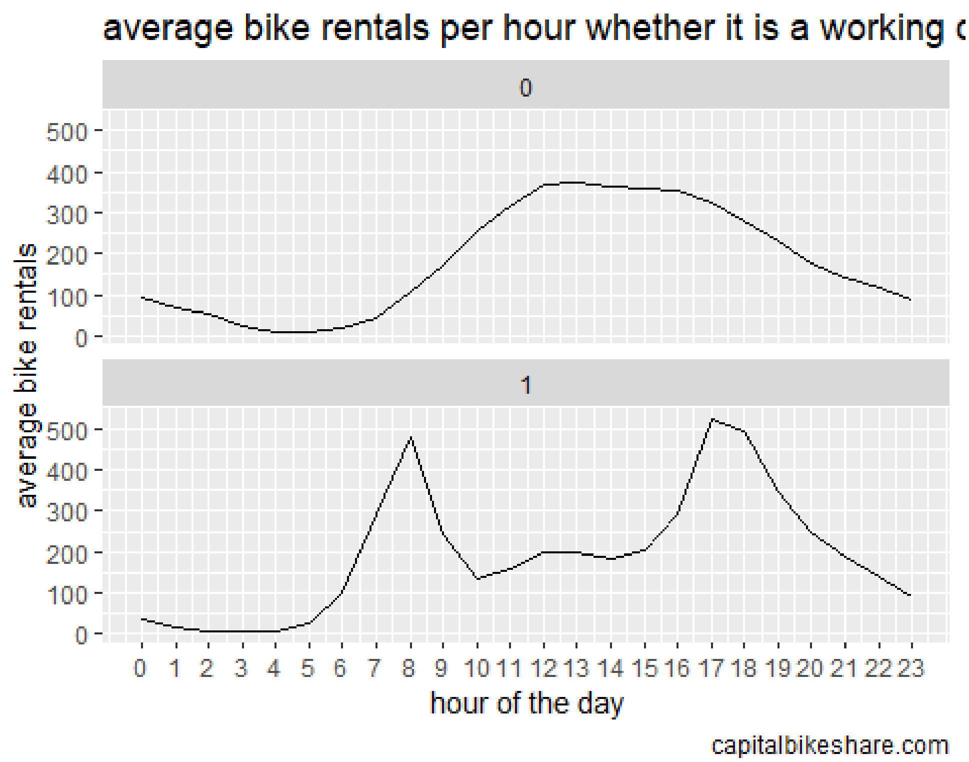
2) Data visualization: a bike share network

```
library(tidyverse)
library(ggplot2)
bikeshare=read.csv('../Data/bikeshare.csv')

#plot A
d1=bikeshare %>%
  group_by(hr) %>%
  summarize(bikeshare_mean=mean(total))

ggplot(data=d1)+
  geom_line(aes(x=hr,y=bikeshare_mean))+
              labs(title="average bike rentals versus hour of the day",
                   caption = "capitalbikeshare.com",
                   x="hour of the day",
                   y="average bike rentals")+
  scale_x_continuous(breaks = 0:23)
```



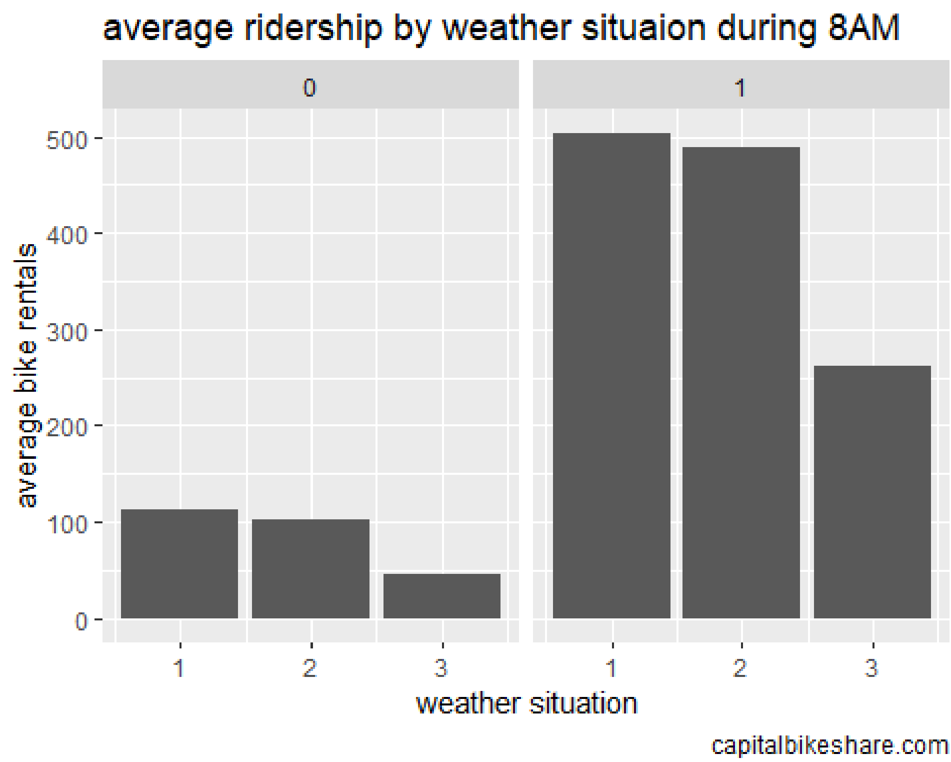
- According to the graph, we can see that between 5:00pm-6:00pm there is the most popular time for renting bikes, with 8:00am being the second most popular.At 4 a.m., there is the least number of people rent bikes.

```
#plot B
d2=bikeshare %>%
  group_by(workingday,hr) %>%
  summarize(bikeshare_mean=mean(total))

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

ggplot(data=d2)+
  geom_line(aes(x=hr,y=bikeshare_mean))+
  facet_wrap(~workingday,nrow = 2)+
  labs(title="average bike rentals per hour whether it is a working day
",
       caption = "capitalbikeshare.com",
       x="hour of the day",
       y="average bike rentals")+
  scale_x_continuous(breaks = 0:23)
```



- For non-working day, people will rent bike during 11:00am-4:00pm. However, there are two busy time plots for renting bike on the working day, they are 8:00am and 5:00pm.

```
#plot C
d3=bikeshare %>%
  filter(hr==8) %>%
  group_by(workingday,weathersit) %>%
  summarize(bikeshare_mean=mean(total))

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

ggplot(d3)+
  geom_col(mapping=aes(x=weathersit,y=bikeshare_mean),
           position = "dodge")+
  facet_wrap(~workingday)+
  labs(title="average ridership by weather situaion during 8AM",
       caption = "capitalbikeshare.com",
       x="weather situation",
       y="average bike rentals")+
  scale_x_continuous(breaks = 0:23)
```



*on the non-working day, most people rent the bike when the weather is clear,few clouds or partly cloudy during the 8:00am, and there are the least bike rent when the weather is light snow or light rain. However, on the working day, when the weather is clear, there still are the most rental bikes. comparing working day and non-working day, we recognize that the number of people renting bikes on working days is 5 times the number of people renting bikes on non-working days in the all weather situations.

3) Data visualization: flights at ABIA

```
library(tidyverse)
library(ggplot2)
ABIA = read.csv('../data/ABIA.csv')
```

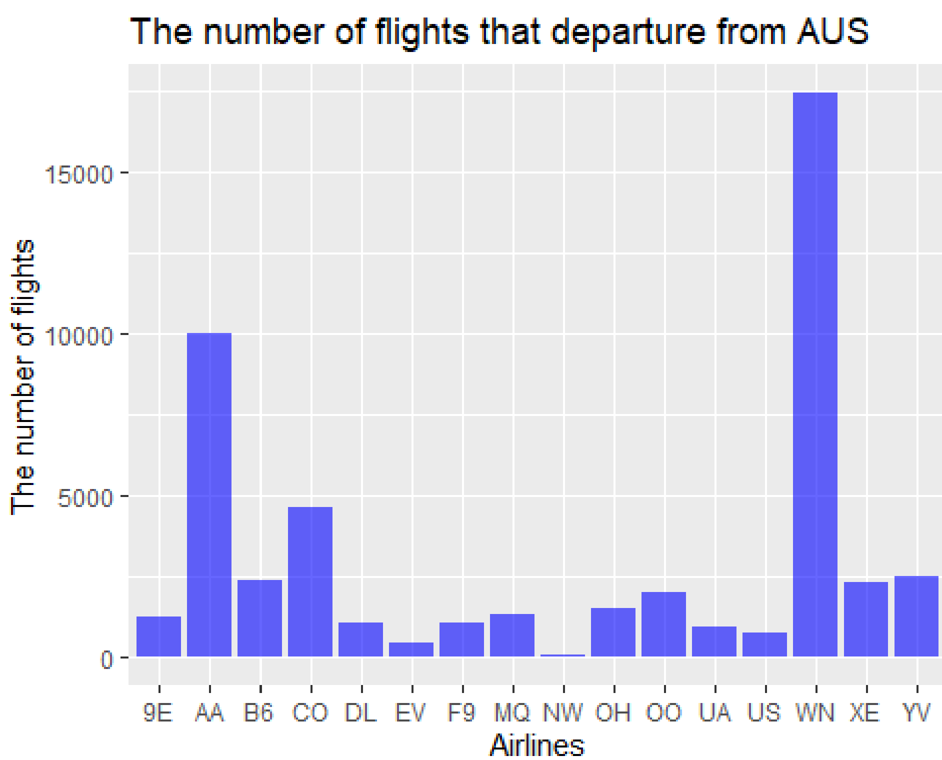
Overall:

In 2008, a total of 49,623 flights are scheduled to depart from Austin, for which American Airlines and Southwest Airlines have the largest number of flights.

```
library(tidyverse)
library(ggplot2)
Total_origin = ABIA %>%
  filter(Origin == 'AUS') %>%
  group_by(UniqueCarrier)%>%
  summarize(total_count = n())

summary(Total_origin)
```

```
##   UniqueCarrier        total_count
##   Length:16          Min.   :   61
##   Class :character   1st Qu.: 1033
##   Mode  :character   Median : 1411
##                      Mean   : 3101
##                      3rd Qu.: 2424
##                      Max.   :17438
```

```
ggplot(data = Total_origin) +
  geom_col(mapping = aes(x=UniqueCarrier, y=total_count),fill = "blue",
 alpha=0.6)+
  labs(title="The number of flights that departure from AUS",
       y="The number of flights",
       x = "Airlines")
```

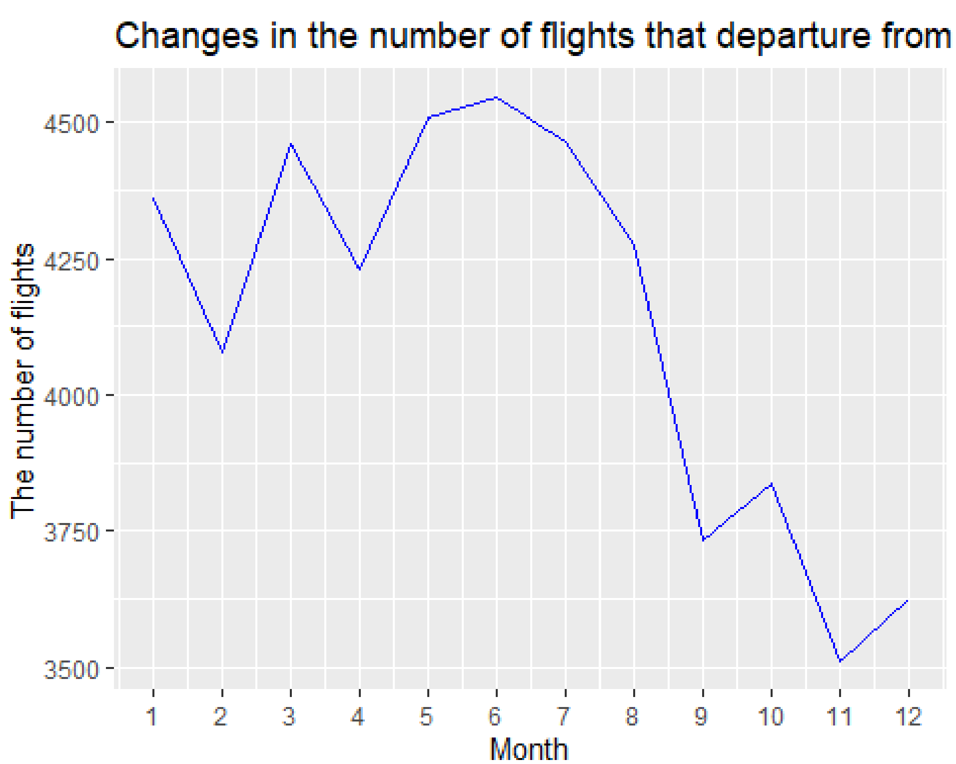


By month, the number of flights departing from Austin in 2008 peaked in June, and then continued to decline until September. Even though the number of flights increased in October, the number of flights in November reached the lowest point of the year.

```
origin_month = ABIA %>%
  filter(Origin == 'AUS') %>%
  group_by(Month)%>%
  summarize(flights = n())

origin_month
```

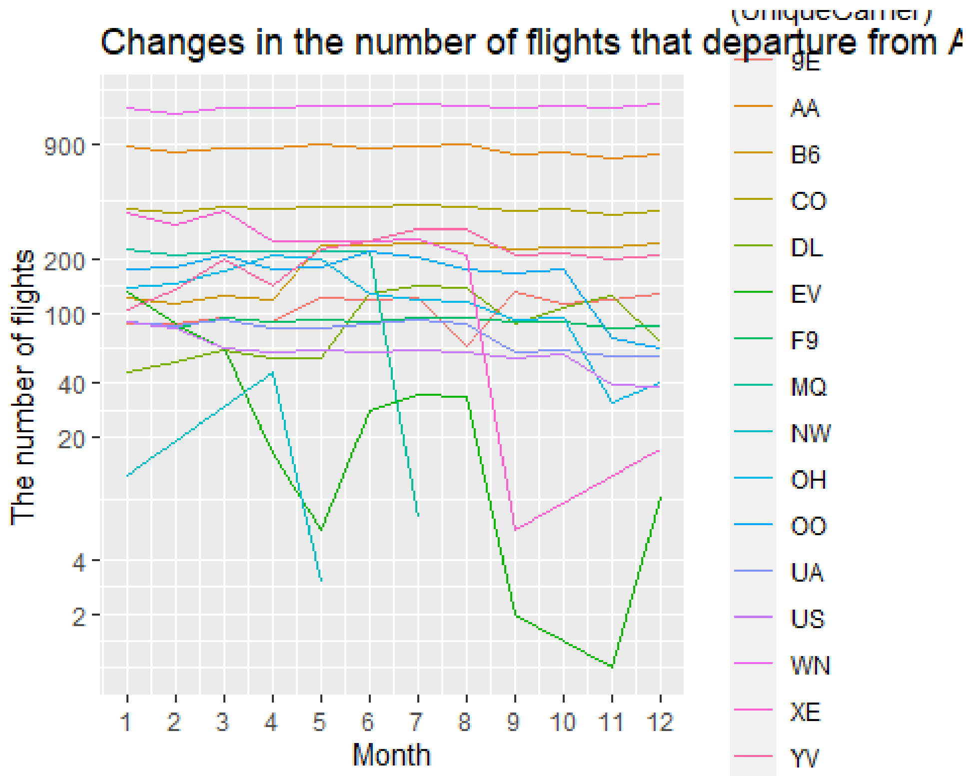
```
## # A tibble: 12 x 2
##    Month flights
##  * <int>   <int>
##  1     1    4361
##  2     2    4077
##  3     3    4459
##  4     4    4229
##  5     5    4507
##  6     6    4545
##  7     7    4465
##  8     8    4276
##  9     9    3733
## 10    10    3837
## 11    11    3510
## 12    12    3624
```

```
ggplot(data = origin_month) +
  geom_line(mapping = aes(x=Month, y=flights),color = "blue")+
  scale_x_continuous(breaks = 1:12)+
  labs(title="Changes in the number of flights that departure from AUS
(months)",
       y="The number of flights",
       x = "Month")
```



To specific, the main factor that influenced the number of flights by month is that some airlines have drastically reduced the number of flights after June, and even no longer provide services, like EV, NW, OH, etc.

```r
origin_month_carrier = ABIA %>%
  filter(Origin == 'AUS') %>%
  group_by(Month,UniqueCarrier)%>%
  summarize(flights = n())

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

origin_month_carrier

## # A tibble: 175 x 3
## # Groups:   Month [12]
##    Month UniqueCarrier flights
##    <int> <chr>           <int>
##  1     1 9E                 87
##  2     1 AA                864
##  3     1 B6                121
##  4     1 CO                382
##  5     1 DL                 46
##  6     1 EV                133
##  7     1 F9                 89
##  8     1 MQ                230
##  9     1 NW                 12
## 10     1 OH                139
## # ... with 165 more rows

ggplot(data = origin_month_carrier) +
  geom_line(mapping = aes(x=Month, y=flights,color=(UniqueCarrier)))+
  scale_x_continuous(breaks = 1:12)+
  scale_y_log10(breaks = c(2,4,20,40,100,200,900))+
  labs(title="Changes in the number of flights that departure from AUS
(airlines)",
       y="The number of flights",
       x = "Month")
```



Flight delay without cancellation

From the bar chart, we know that the airlines of EV and WN have the highest departure delay rates, and the departure delay rate of WN is even close to 50%.

```
ABIA = ABIA %>%
  mutate(if_delay = ifelse(DepDelay > 0, 1, 0))
head(ABIA)

##   Year Month DayofMonth DayOfWeek DepTime CRSDepTime ArrTime CRSArrT
ime
## 1 2008     1          1         2     120       1935     309        2
130
## 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 A
irTime
## 1            9E      5746  84129E               109            115
    88
```

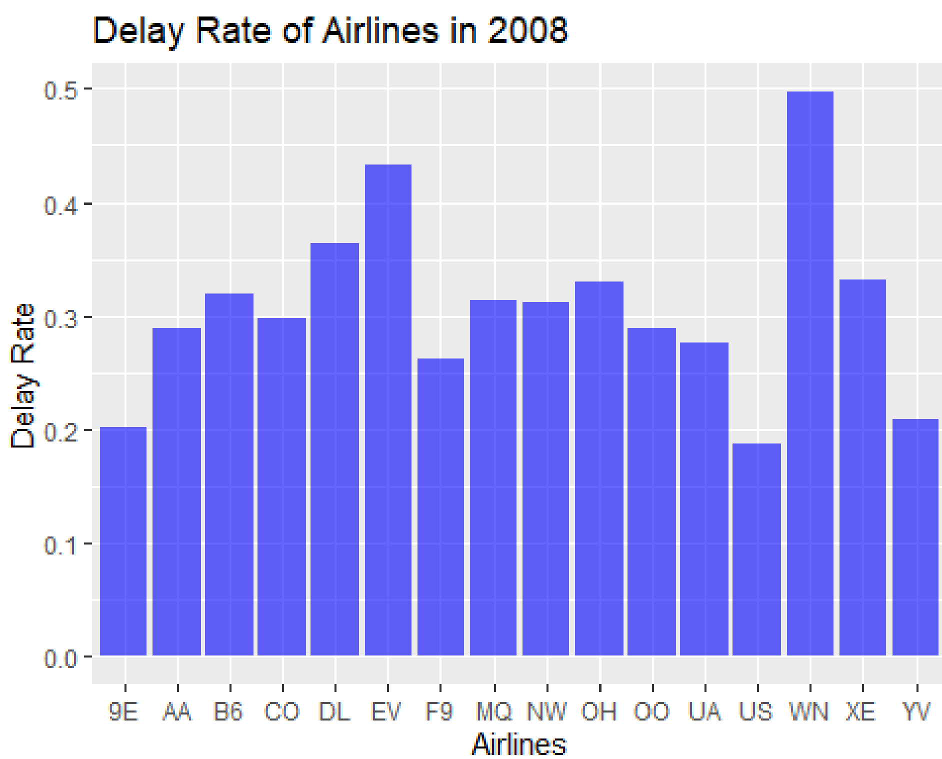
```
## 2             AA      1614  N438AA               151            155
   133
## 3             YV      2883  N922FJ               148            149
   125
## 4             9E      5743  89189E                86            105
    70
## 5             AA      1157  N4XAAA                53             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 Secur
ityDelay
## 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 if_delay
## 1                 0        1
## 2                NA        0
## 3                NA        0
## 4                NA        0
## 5                NA        1
## 6                NA        0
```

```
P1= ABIA %>%
  filter(Origin == 'AUS', Cancelled == 0) %>%
  group_by(UniqueCarrier) %>%
  summarize(total_count = n(),delay_num =sum(if_delay),delay_rate = del
ay_num/total_count)
P1
```

```
## # A tibble: 16 x 4
##    UniqueCarrier total_count delay_num delay_rate
##  * <chr>               <int>     <dbl>      <dbl>
##  1 9E                   1245       251      0.202
```

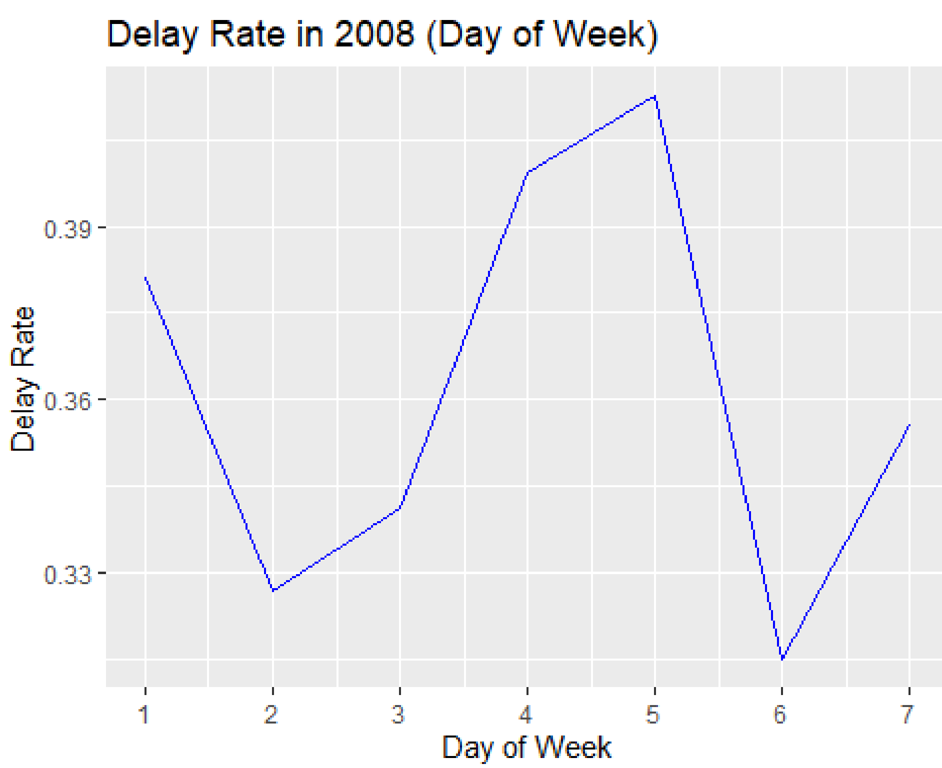
```
##   2 AA                     9709      2805      0.289
##   3 B6                     2367       757      0.320
##   4 CO                     4554      1357      0.298
##   5 DL                     1056       384      0.364
##   6 EV                      407       176      0.432
##   7 F9                     1064       279      0.262
##   8 MQ                     1245       390      0.313
##   9 NW                       61        19      0.311
##  10 OH                     1463       482      0.329
##  11 OO                     1976       570      0.288
##  12 UA                      923       255      0.276
##  13 US                      727       136      0.187
##  14 WN                    17343      8621      0.497
##  15 XE                     2296       762      0.332
##  16 YV                     2455       514      0.209
```

```
ggplot(data = P1) +
  geom_col(mapping = aes(x=UniqueCarrier, y=delay_rate),fill = "blue",
alpha=0.6)+
  labs(title="Delay Rate of Airlines in 2008",
       y="Delay Rate",
       x = "Airlines")
```



From the line chart, we find that the departure delay rates on Tuesday and Saturday are the lowest, and the departure delay rates on Thursday and Friday are relatively higher than others.

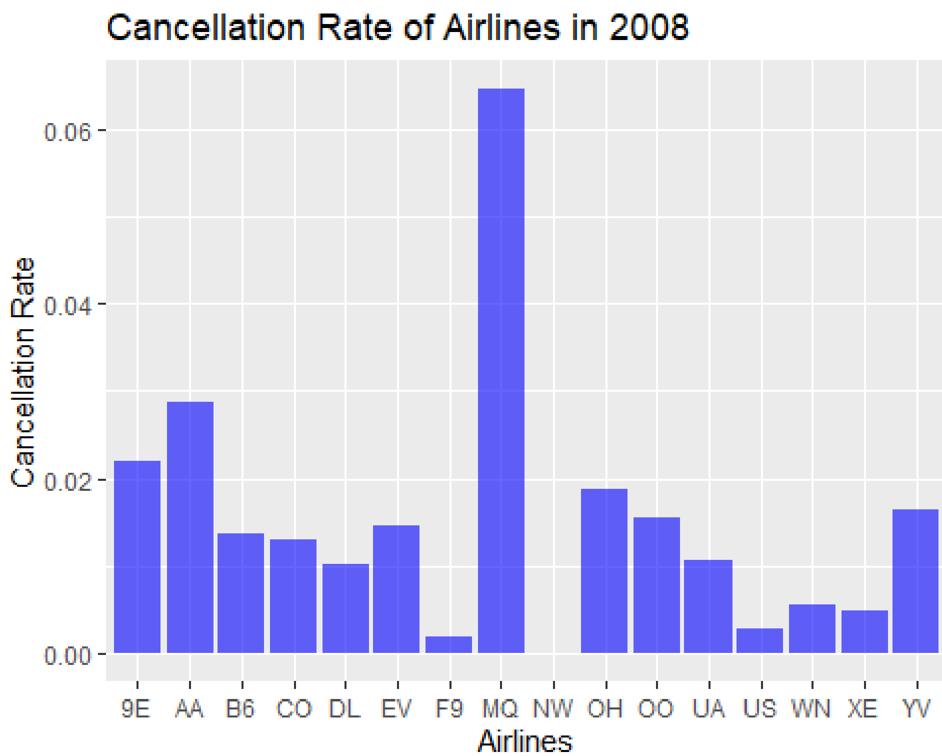
```
P2= ABIA %>%
  filter(Origin == 'AUS', Cancelled == 0) %>%
  group_by(DayOfWeek)%>%
  summarise(total_count = n(),delay_num =sum(if_delay),delay_rate = del
ay_num/total_count)
P2

## # A tibble: 7 x 4
##   DayOfWeek total_count delay_num delay_rate
## *     <int>       <int>     <dbl>      <dbl>
## 1         1        7299      2782      0.381
## 2         2        7265      2373      0.327
## 3         3        7294      2488      0.341
## 4         4        7274      2904      0.399
## 5         5        7270      3000      0.413
## 6         6        5618      1769      0.315
## 7         7        6871      2442      0.355

ggplot(data = P2) +
  geom_line(mapping = aes(x=DayOfWeek, y=delay_rate),color = "blue")+
  scale_x_continuous(breaks = 1:7)+
  labs(title="Delay Rate in 2008 (Day of Week) ",
       y="Delay Rate",
       x = "Day of Week")
```



The reason of cancellation

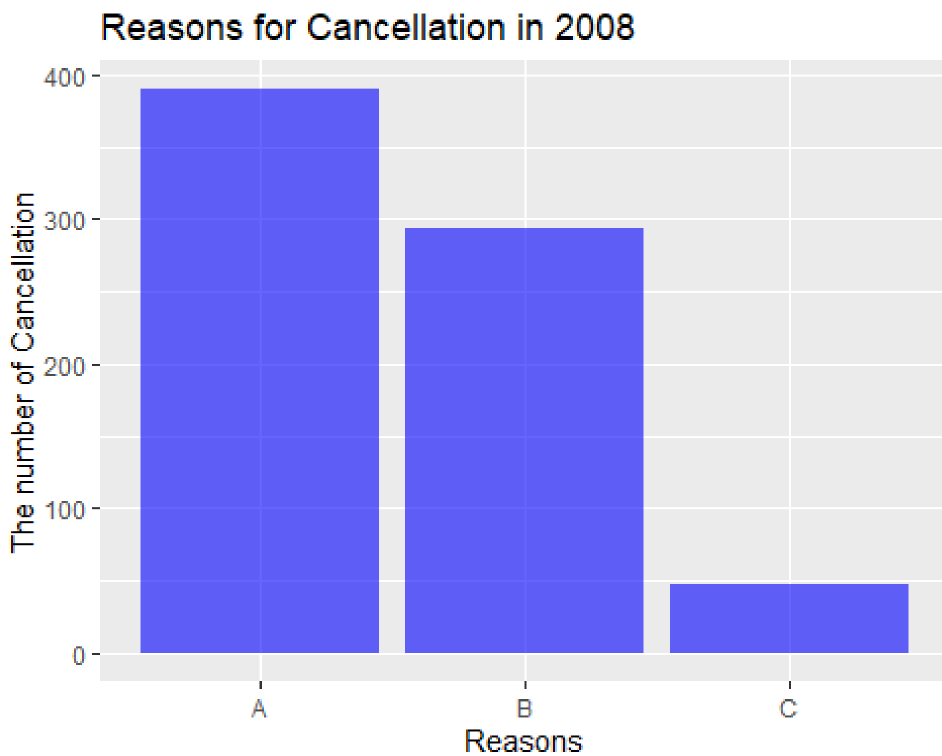
In 2008, the airline of EQ has the highest rate of cancellation, the airline of NW has the lowest rate of cancellation.

```
Total_origin_cancel = ABIA %>%
  filter(Origin == 'AUS') %>%
  group_by(UniqueCarrier)%>%
  summarize(total_count = n(),cancellation_num = sum(Cancelled), cancel
lation_rate = cancellation_num/total_count)

Total_origin_cancel

## # A tibble: 16 x 4
##    UniqueCarrier total_count cancellation_num cancellation_rate
##  * <chr>               <int>            <int>             <dbl>
##  1 9E                   1273               28           0.0220
##  2 AA                   9997              288           0.0288
##  3 B6                   2400               33           0.0138
##  4 CO                   4614               60           0.0130
##  5 DL                   1067               11           0.0103
##  6 EV                    413                6           0.0145
##  7 F9                   1066                2           0.00188
##  8 MQ                   1331               86           0.0646
##  9 NW                     61                0           0
## 10 OH                   1491               28           0.0188
## 11 OO                   2007               31           0.0154
## 12 UA                    933               10           0.0107
## 13 US                    729                2           0.00274
## 14 WN                  17438               95           0.00545
## 15 XE                   2307               11           0.00477
## 16 YV                   2496               41           0.0164

ggplot(data = Total_origin_cancel) +
  geom_col(mapping = aes(x=UniqueCarrier, y=cancellation_rate),fill = "
blue", alpha=0.6)+
  labs(title="Cancellation Rate of Airlines in 2008 ",
       y="Cancellation Rate",
       x = "Airlines")
```



In the bar chart, there are three cancellation reasons(A = carrier, B = weather, C = NAS). Obviously, most cancellation is due to carrier.

```
p4 = ABIA %>%
  filter(Origin == 'AUS',Cancelled==1) %>%
  group_by(CancellationCode)%>%
  summarize(total_count = n())
p4

## # A tibble: 3 x 2
##   CancellationCode total_count
## * <chr>                  <int>
## 1 A                        390
## 2 B                        294
## 3 C                         48

ggplot(data = p4) +
  geom_col(mapping = aes(x=CancellationCode,y=total_count),fill = "blue
", alpha=0.6)+
  labs(title="Reasons for Cancellation in 2008 ",
       y="The number of Cancellation",
       x = "Reasons")
```



Furthermore, among those carriers, we find that AA carrier has the largest number of cancellations. For carrier of AA, most of the reason for cancellation is because of the carrier itself; for carrier of WN, the number of cancellations due to reason B accounts for the majority.

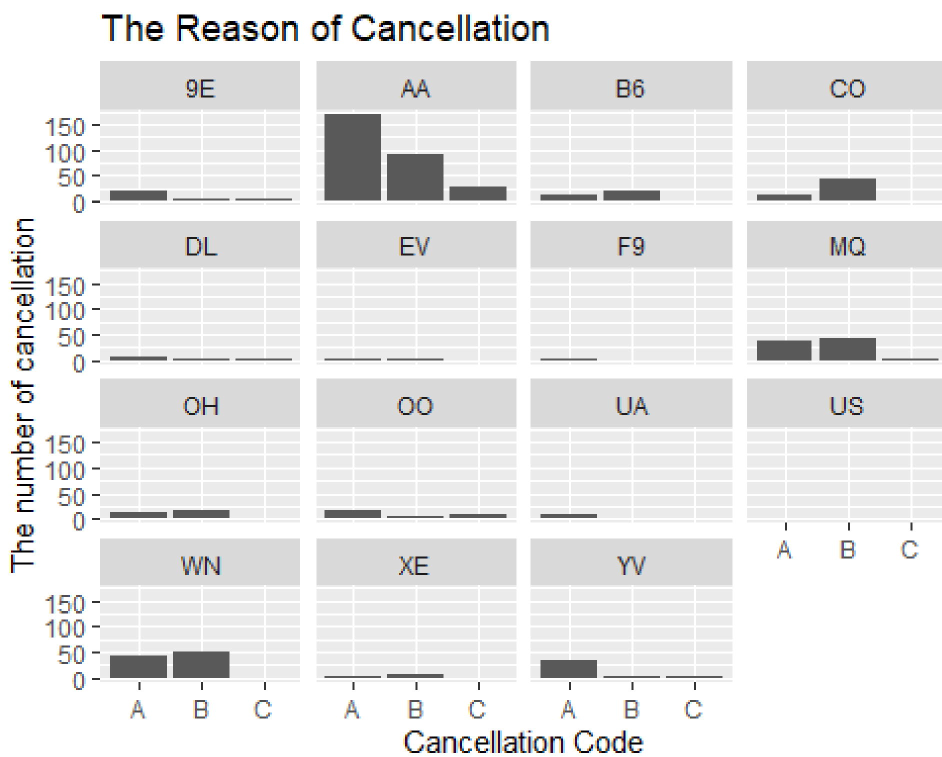
```
p5 = ABIA %>%
  filter(Origin == 'AUS',Cancelled==1) %>%
  group_by(CancellationCode,UniqueCarrier)%>%
  summarize(total_count = n())
```

```
## `summarise()` has grouped output by 'CancellationCode'. You can over
ride using the `.groups` argument.
```

```
p5
```

```
## # A tibble: 36 x 3
## # Groups:   CancellationCode [3]
##    CancellationCode UniqueCarrier total_count
##    <chr>            <chr>               <int>
##  1 A                9E                     21
##  2 A                AA                    170
##  3 A                B6                     12
##  4 A                CO                     14
##  5 A                DL                      6
##  6 A                EV                      2
##  7 A                F9                      2
##  8 A                MQ                     40
```

```
## 9 A                OH                      13
## 10 A               OO                      18
## # ... with 26 more rows

ggplot(data = p5) +
  geom_col(mapping = aes(x=CancellationCode,y=total_count))+
  facet_wrap(~UniqueCarrier)+
  labs(title="The Reason of Cancellation",
       y="The number of cancellation",
       x = "Cancellation Code",
       fill="CancellationCode")
```



In conclusion: if you want to depart from Austin by plane, you'd better avoid Tuesday and Saturday, and buy other airlines besides EV, WN and MQ.

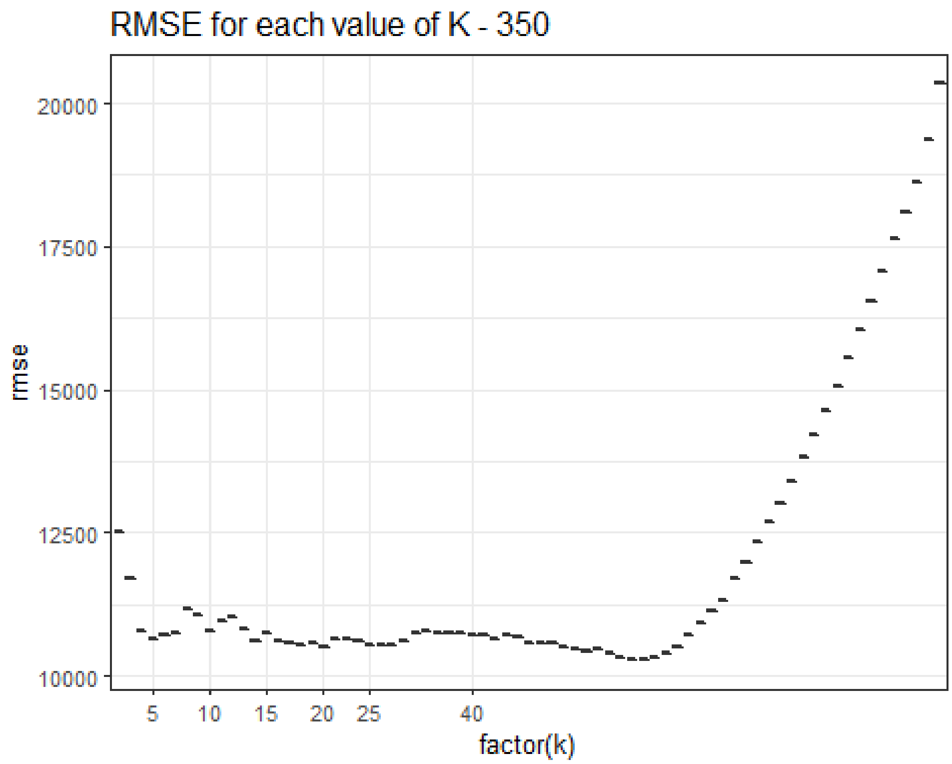
4) K-nearest neighbors

i. 350

```
library(tidyverse)
library(ggplot2)
library(mosaic)

## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2

##
## The 'mosaic' package masks several functions from core packages in o
```

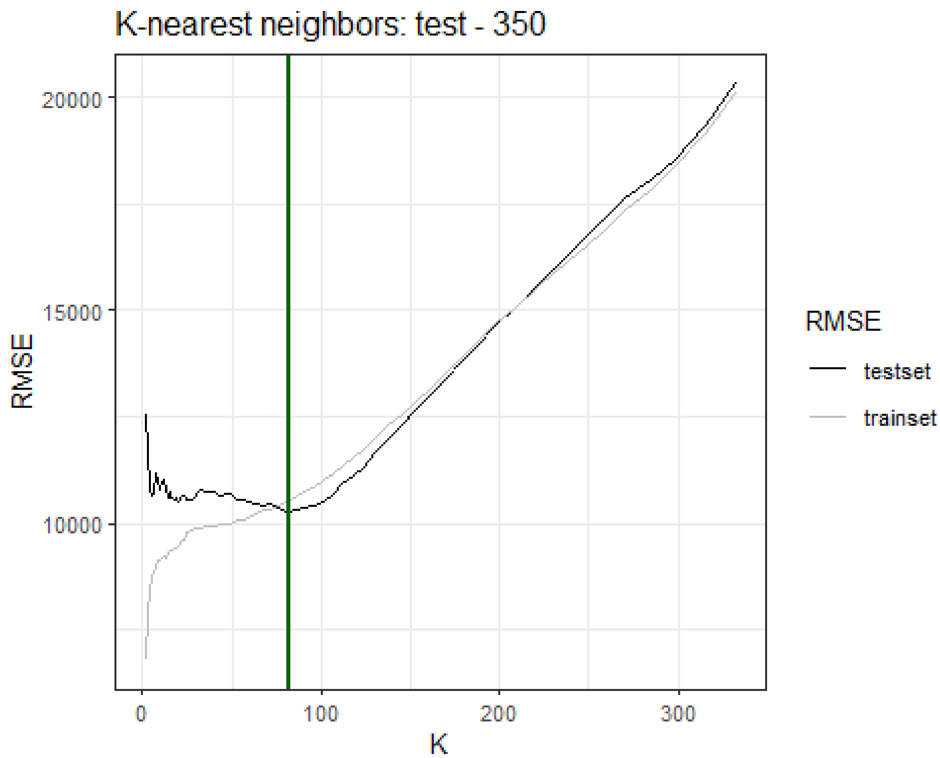
```
rder to add
## additional features.  The original behavior of these functions shoul
d not be affected by this.

##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##     mean

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

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

## The following object is masked from 'package:ggplot2':
##
##     stat

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum

library(FNN)
library(foreach)

##
## Attaching package: 'foreach'

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

library(rsample)
library(caret)

##
## Attaching package: 'caret'

## The following object is masked from 'package:mosaic':
##
##     dotPlot
```

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

library(modelr)

##
## Attaching package: 'modelr'

## The following object is masked from 'package:mosaic':
##
##     resample

## The following object is masked from 'package:ggformula':
##
##     na.warn

library(parallel)

sclass = read.csv('../data/sclass.csv')

sclass350 = subset(sclass, trim == '350')

# Split the data into a training and a testing set
sclass350_split = initial_split(sclass350, prop=0.9)
sclass350_train = training(sclass350_split)
sclass350_test  = testing(sclass350_split)

# RMSE for each value of K
N = nrow(sclass350)
N_train = floor(0.8*N)
k_grid = unique(round(exp(seq(log(N_train), log(2), length=100))))

rmse_out = foreach(k = k_grid, .combine='rbind') %dopar% {
  this_rmse = foreach(k = k_grid, .combine='c') %do% {
    knn_model = knnreg(price ~ mileage, data=sclass350_train, k = k, us
e.all=TRUE)
    modelr::rmse(knn_model, sclass350_test)
  }
  data.frame(k=k_grid, rmse=this_rmse)
}

## Warning: executing %dopar% sequentially: no parallel backend registe
red

rmse_out = arrange(rmse_out, k)
ggplot(rmse_out) +
  geom_boxplot(aes(x=factor(k), y=rmse)) +
  theme_bw(base_size=10) +
  scale_x_discrete(breaks=c(5,10,15,20,25,30,40,50,80,100)) +
  labs (titles = "RMSE for each value of K - 350")
```

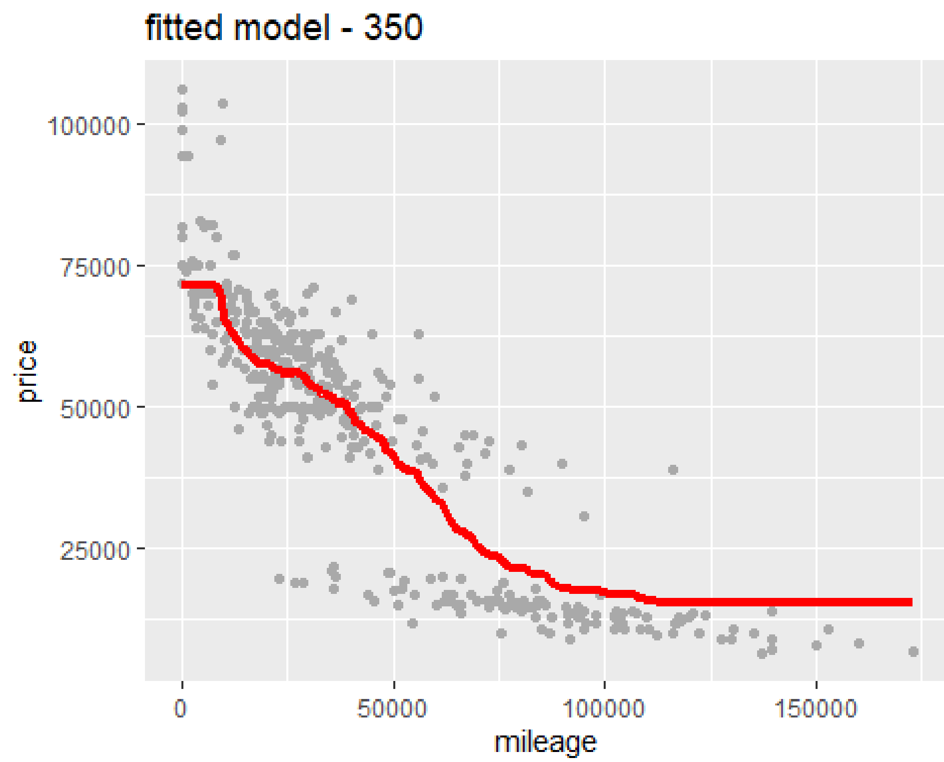



```
#  K-nearest-neighbors
rmse_grid_out = foreach(k = k_grid, .combine='c') %do% {
  knn_model = knnreg(price ~ mileage, data=sclass350_train, k = k, use.
all=TRUE)
  modelr::rmse(knn_model, sclass350_test)
}
rmse_grid_out = data.frame(K = k_grid, RMSE = rmse_grid_out)

p_out = ggplot(data=rmse_grid_out) +
  theme_bw(base_size = 10) +
  geom_path(aes(x=K, y=RMSE, color='testset'), size=0.5)

ind_best = which.min(rmse_grid_out$RMSE)
k_best = k_grid[ind_best]

rmse_grid_in = foreach(k = k_grid, .combine='c') %do% {
  knn_model = knnreg(price ~ mileage, data=sclass350_train, k = k, use.
all=TRUE)
  modelr::rmse(knn_model, sclass350_train)
}
rmse_grid_in = data.frame(K = k_grid, RMSE = rmse_grid_in)
p_out + geom_path(data=rmse_grid_in, aes(x=K, y=RMSE, color='trainset'),
size=0.5) +
  scale_colour_manual(name="RMSE",
                      values=c(testset="black", trainset="grey")) +
```

```
  geom_vline(xintercept=k_best, color='darkgreen', size=1) +
  labs (titles = "K-nearest neighbors: test - 350")
```



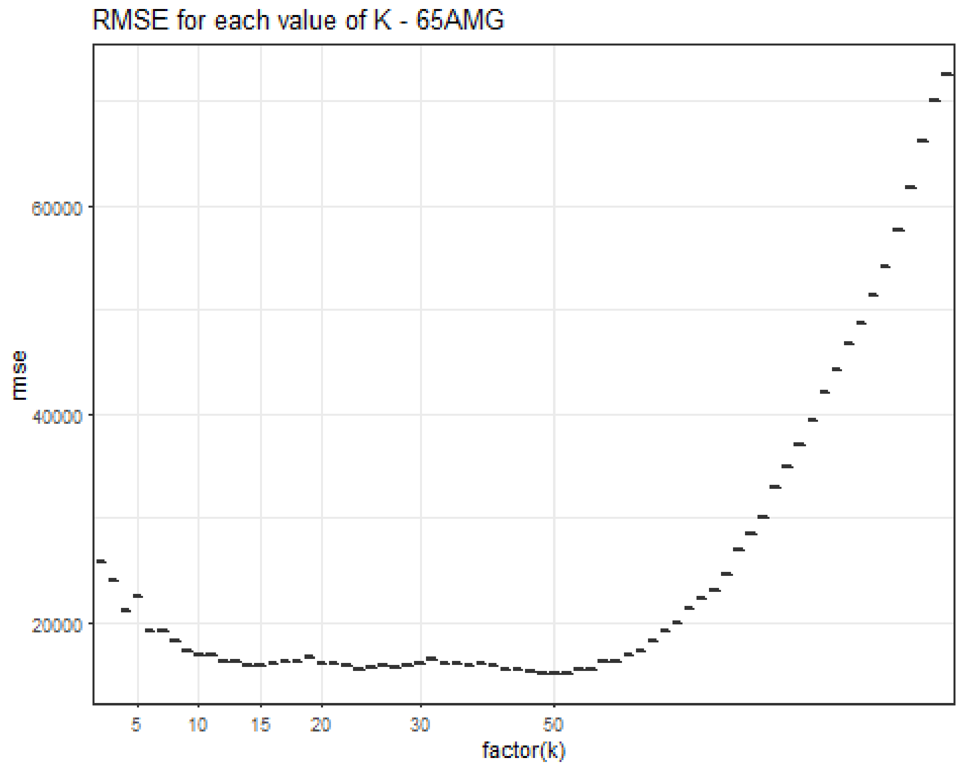

```
# fitted model
knn = knnreg(price ~ mileage, data=sclass350_train, k=k_best)
sclass350 = sclass350 %>%
  mutate(price_pre = predict(knn, sclass350))
g350 = ggplot(data = sclass350) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey')
g350 + geom_line(aes(x = mileage, y = price_pre), color='red', size=1.5)
 +
  labs (titles = "fitted model - 350")
```



ii. 65 AMG

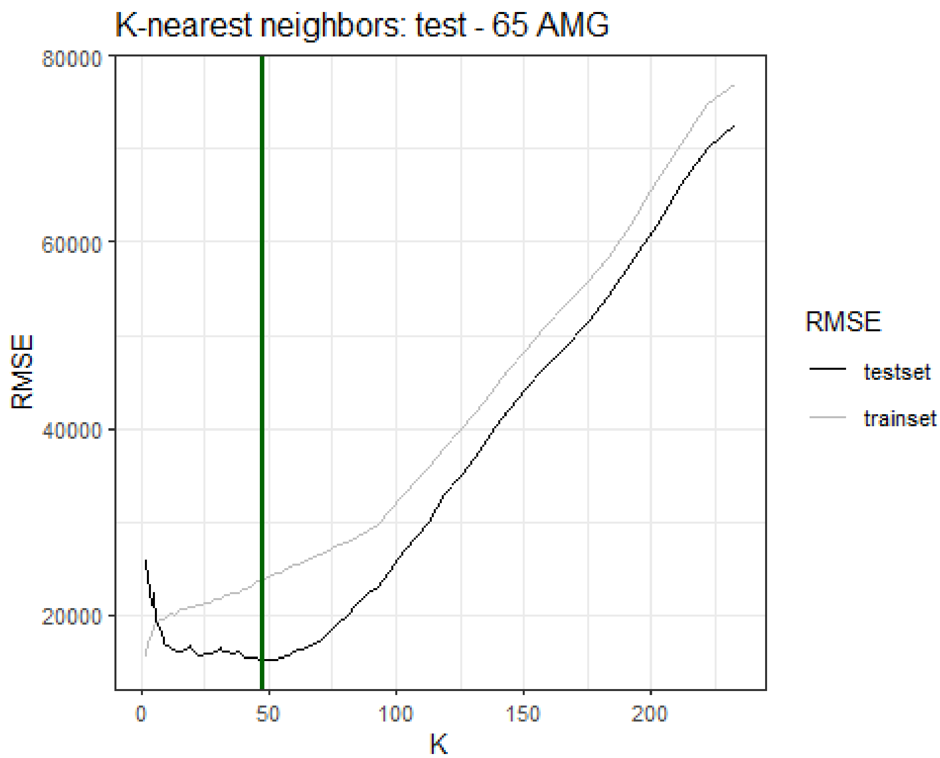
```
# Split the data into a training and a testing set
sclass65AMG = subset(sclass, trim == '65 AMG')

sclass65AMG_split = initial_split(sclass65AMG, prop=0.9)
sclass65AMG_train = training(sclass65AMG_split)
sclass65AMG_test  = testing(sclass65AMG_split)

# RMSE for each value of K
N65AMG = nrow(sclass65AMG)
N_train65AMG = floor(0.8*N65AMG)
k_grid65AMG = unique(round(exp(seq(log(N_train65AMG), log(2), length=10
0))))
rmse_out65AMG = foreach(k = k_grid65AMG, .combine='rbind') %dopar% {
  this_rmse = foreach(k = k_grid65AMG, .combine='c') %do% {
    knn_model = knnreg(price ~ mileage, data=sclass65AMG_train, k = k,
use.all=TRUE)
    modelr::rmse(knn_model, sclass65AMG_test)
  }
  data.frame(k=k_grid65AMG, rmse=this_rmse)
}
rmse_out65AMG = arrange(rmse_out65AMG, k)
ggplot(rmse_out65AMG) +
  geom_boxplot(aes(x=factor(k), y=rmse)) +
  theme_bw(base_size=8) +
```

```
  scale_x_discrete(breaks=c(5,10,15,20,25,30,40,50,80,100)) +
  labs (titles = "RMSE for each value of K - 65AMG")
```

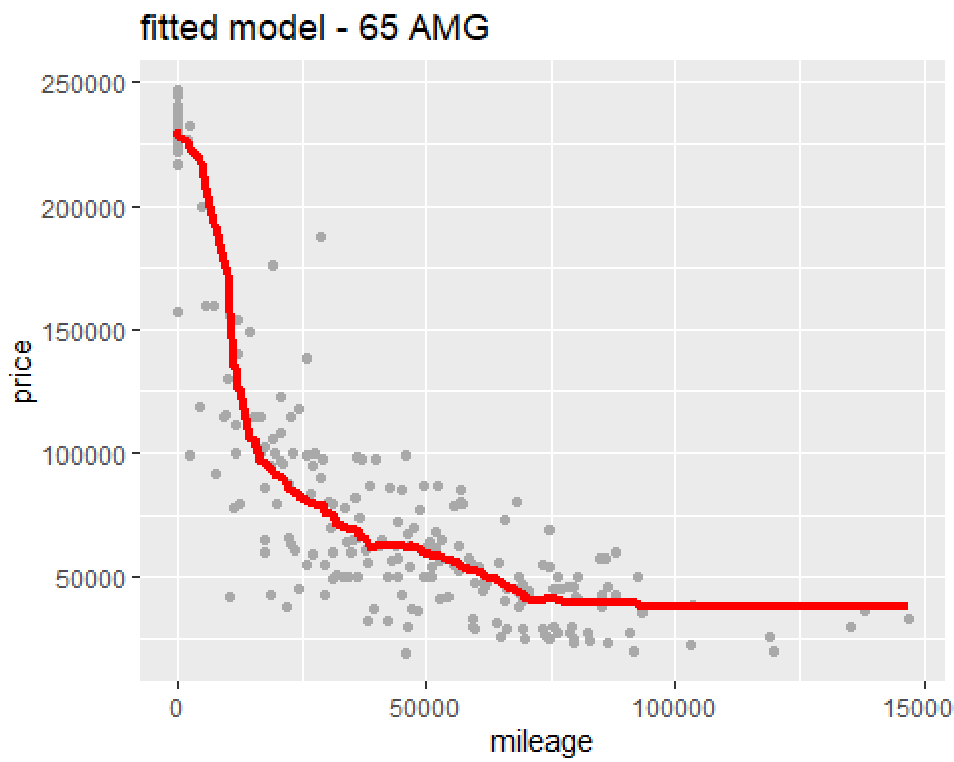



```
#  K-nearest-neighbors
rmse_grid_out65AMG = foreach(k = k_grid65AMG, .combine='c') %do% {
  knn_model = knnreg(price ~ mileage, data=sclass65AMG_train, k = k, us
e.all=TRUE)
  modelr::rmse(knn_model, sclass65AMG_test)
}
rmse_grid_out65AMG = data.frame(K = k_grid65AMG, RMSE = rmse_grid_out65
AMG)
p_out = ggplot(data=rmse_grid_out65AMG) +
  theme_bw(base_size = 10) +
  geom_path(aes(x=K, y=RMSE, color='testset'), size=0.5)

ind_best65AMG = which.min(rmse_grid_out65AMG$RMSE)
k_best65AMG = k_grid65AMG[ind_best65AMG]

rmse_grid_in2 = foreach(k = k_grid65AMG, .combine='c') %do% {
  knn_model = knnreg(price ~ mileage, data=sclass65AMG_train, k = k, us
e.all=TRUE)
  modelr::rmse(knn_model, sclass65AMG_train)
}
rmse_grid_in2 = data.frame(K = k_grid65AMG, RMSE = rmse_grid_in2)
p_out + geom_path(data=rmse_grid_in2, aes(x=K, y=RMSE, color='trainset
'),size=0.5) +
  scale_colour_manual(name="RMSE",
```

```
                              values=c(testset="black", trainset="grey")) +
  geom_vline(xintercept=k_best65AMG, color='darkgreen', size=1)+
  labs (titles = "K-nearest neighbors: test - 65 AMG")
```




```
# fitted model
knn65AMG = knnreg(price ~ mileage, data=sclass65AMG_train, k=k_best65AM
G)
sclass65AMG = sclass65AMG %>%
  mutate(price_pre = predict(knn65AMG, sclass65AMG))
g65AMG = ggplot(data = sclass65AMG) +
  geom_point(mapping = aes(x = mileage, y = price), color='darkgrey')
g65AMG + geom_line(aes(x = mileage, y = price_pre), color='red', size=1.
5) +
  labs (titles = "fitted model - 65 AMG")
```




```
k_best

## [1] 82

k_best65AMG

## [1] 48

dim(sclass350)

## [1] 416  18

dim(sclass65AMG)

## [1] 292  18
```

Trim 350 yields a larger optimal value of K. In the plot of RMSE versus K, trim 350 has the higher K. I reckon that it's due to trim 350 has more number of data than trim 65AMG. If the value of K is small, once there are noise components, they will have a greater impact on the prediction. When the value of K is large, it is equivalent to predicting with data in a larger neighborhood, and the approximate error of learning will increase. Because of dataset "sclass350" has more points , the optimal value of K can be larger in order to reduce the bias.