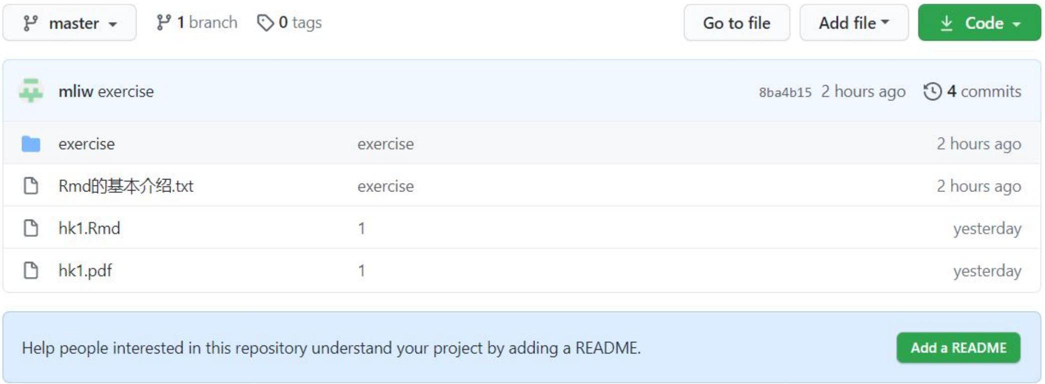
ECO395M STAT LEARNING Homework 1[*]

Mingwei Li, Xinyu Leng, Hongjin Long

Abstract

This document is the first homework of ECO395M STAT LEARNING. The projects **Gas Price**, **a bike share network**, **flights at ABIA** and **K-nearest neighbors** are included in this document. The whole project is available at here.



mliw exercise	8ba4b15 2 hours ago	4 commits
exercise	exercise	2 hours ago
Rmd的基本介绍.txt	exercise	2 hours ago
hk1.Rmd	1	yesterday
hk1.pdf	1	yesterday

Help people interested in this repository understand your project by adding a README.

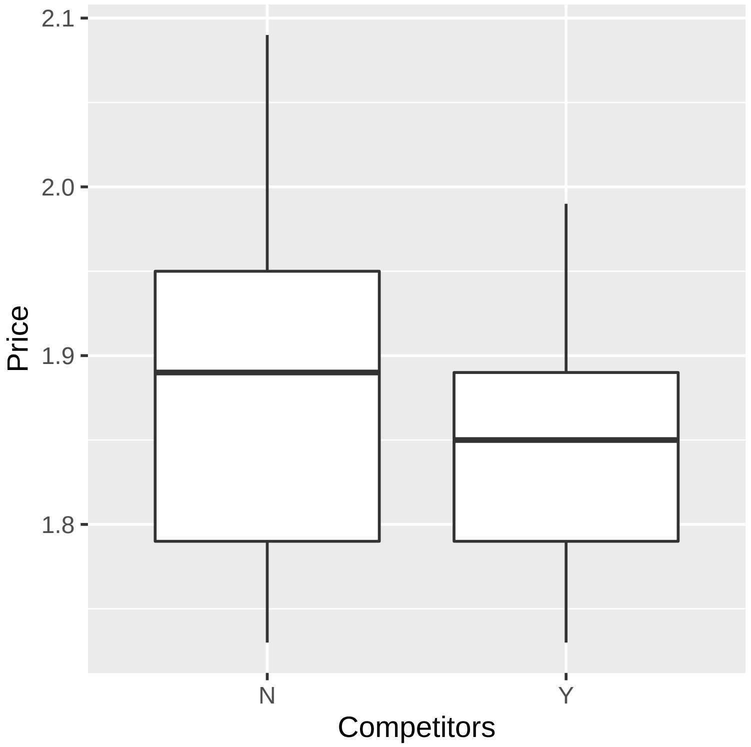
[*] Mingwei Li, Xinyu Leng and Hongjin Long are master students of economics, The University of Texas at Austin

Contents

1 Gas Price

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

```
p0 = ggplot(data=gasprice) +
  geom_boxplot(aes(x=Competitors, y=Price))
p0
```

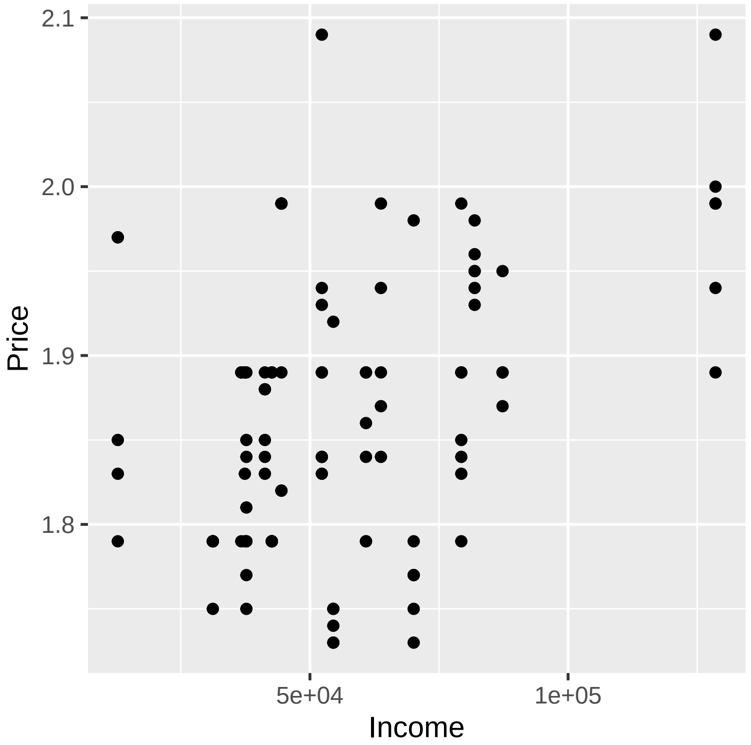


Claim: Gas stations charge more if they lack direct competition in sight

Conclusion: The theory is supported by the data. As the left box is higher than the right box.

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

```
p0 = ggplot(data=gasprice) +
  geom_point(aes(x=Income, y=Price))
p0
```

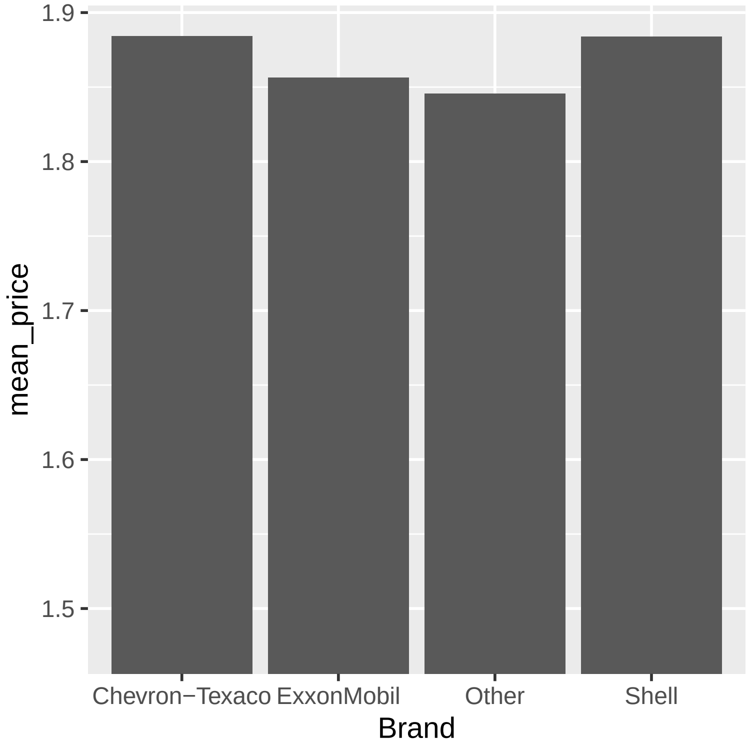


Claim: The richer the area, the higher the gas price

Conclusion: The theory is supported by the data. As the points in the figure show an increasing trend(this trend not very clear).

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

```
d4 = gasprice %>%
  group_by(Brand) %>%
  summarize(mean_price=mean(Price))
p0 = ggplot(data=d4) +
  geom_col(aes(x=Brand, y=mean_price))
p0 + coord_cartesian(ylim =c(min(d4$mean_price)*0.8, max(d4$mean_price)))
```

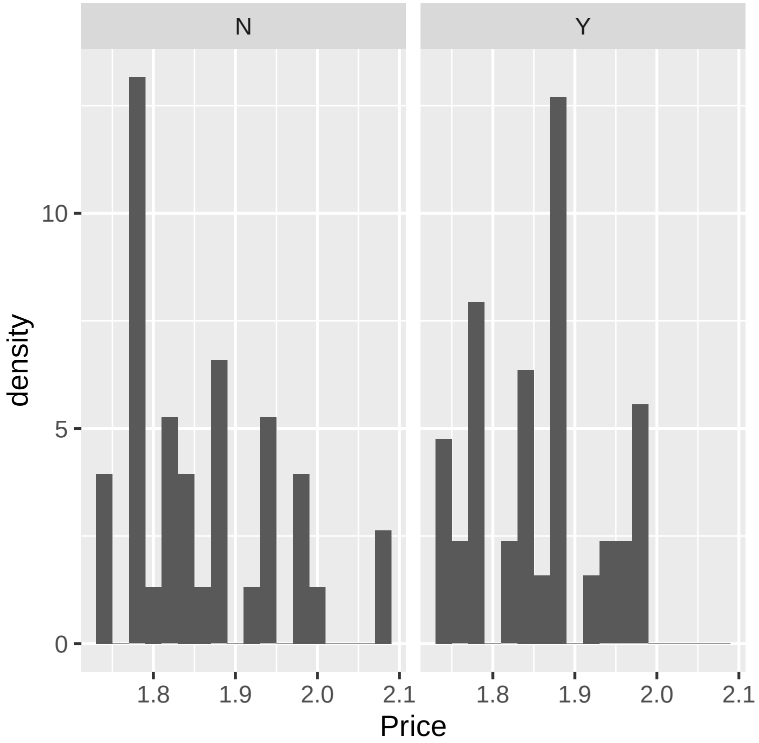


Claim: Shell charges more than other brands

Conclusion: The theory is supported by the data. As the bar of Shell is the highest (the same as Chevron-Texaco).

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

```
p0 = ggplot(data=gasprice) +
  geom_histogram(aes(x=Price, after_stat(density)),binwidth=0.02) +
  facet_wrap(~Stoplight)
p0
```



Claim: Gas stations at stoplights charge more

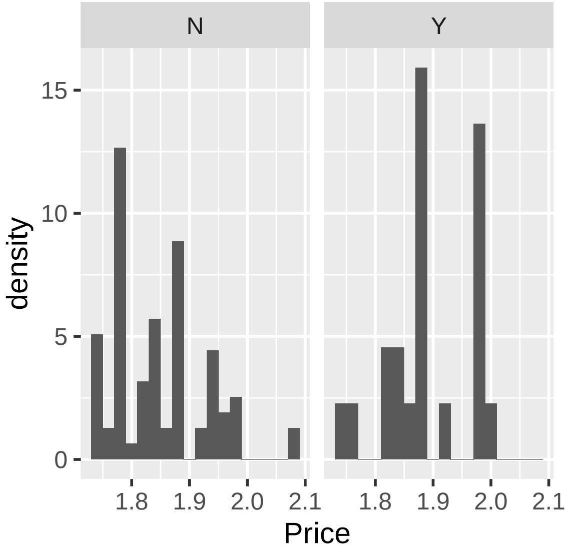
Conclusion: The theory is supported by the data. As more density is put on the right tail of gas stations at stoplights(However, the graph is not very clear in this problem)

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

faceted histogram and boxplot are used to solve this problem.

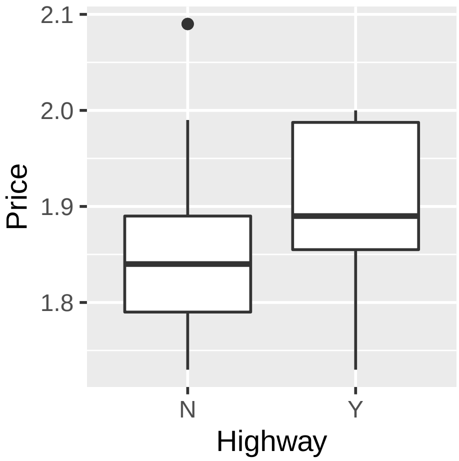
(1)faceted histogram:

```
p0 = ggplot(data=gasprice) +
  geom_histogram(aes(x=Price, after_stat(density)),binwidth=0.02) +
  facet_wrap(~Highway)
p0
```



(2)boxplot:

```
p0 = ggplot(data=gasprice) +
  geom_boxplot(aes(x=Highway, y=Price))
p0
```



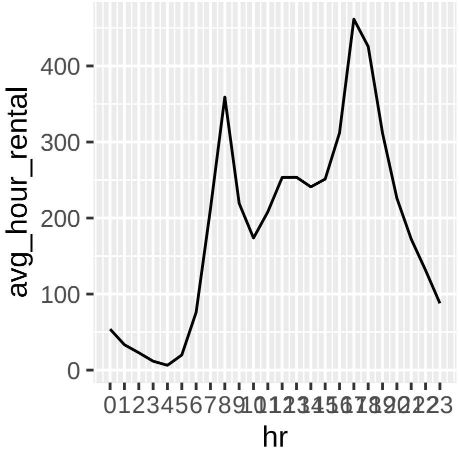
Claim: Gas stations with direct highway access charge more

Conclusion: The theory is supported by the data. As in histogram, more density is put on the right tail of gas stations with direct highway access. The boxplot reaches the same conclusion.

2 a bike share network

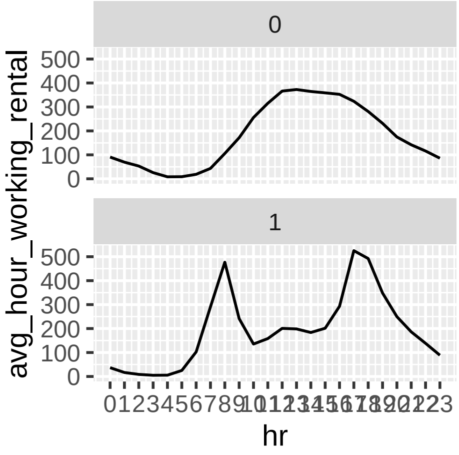
(a) Plot A: a line graph showing average bike rentals (total) versus hour of the day (hr).

```
avg_hr_rent=bikeshare %>%
  group_by(hr) %>%
  summarize(avg_hour_rental=mean(total))
ggplot(data=avg_hr_rent) +
  geom_line(aes(x=hr, y=avg_hour_rental)) +
  scale_x_continuous(breaks=0:23)
```



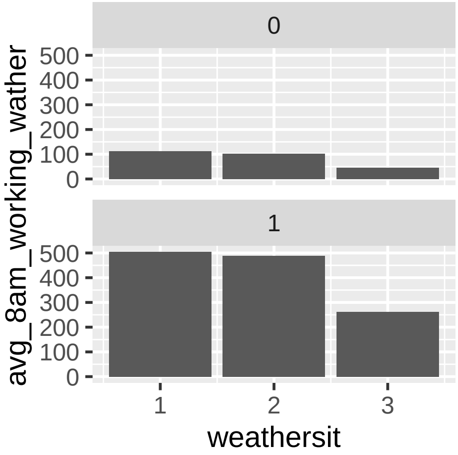
(b) Plot B: a faceted line graph showing average bike rentals versus hour of the day, faceted according to whether it is a working day (workingday).

```
avg_hr_work_rent=bikeshare %>%
  group_by(hr, workingday) %>%
  summarize(avg_hour_working_rental=mean(total))
ggplot(data=avg_hr_work_rent) +
  geom_line(aes(x=hr, y=avg_hour_working_rental)) +
  scale_x_continuous(breaks=0:23) +
  facet_wrap(~ workingday, nrow=2)
```



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

```
bikeshare_8am=bikeshare %>%
  filter(hr==8)
bike_8am_work_wather= bikeshare_8am %>%
  group_by(workingday, weathersit) %>%
  summarise(avg_8am_working_wather=mean(total))
ggplot(data=bike_8am_work_wather) +
  geom_col(aes(x=weathersit, y=avg_8am_working_wather)) +
  facet_wrap(~ workingday, nrow=2)
```



3 flights at ABIA

(1)Dataset and Pakages

```
ABIA <- read.csv("data/ABIA.csv", stringsAsFactors=TRUE)
library(ggplot2)
library(tidyverse)
```

(2)Consider the best time of day to fly to minimize delays

select the variables

```
ABIA2<-select(ABIA,Year,Month,DayofMonth,DayOfWeek,CRSDepTime,DepDelay,UniqueCarrier)
```

Filter the variables

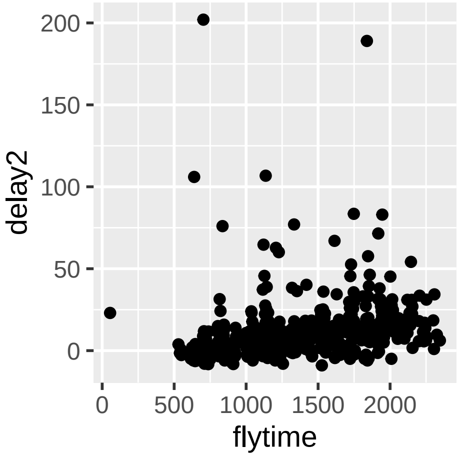
```
ABIA2<-filter(ABIA2, !is.na(DepDelay), !is.na(CRSDepTime))
```

group by the CRSDepTime

```
by_CRSDepTime<-group_by(ABIA2,CRSDepTime)
delay2<-summarise(by_CRSDepTime,count = n(),
                  flytime=mean(CRSDepTime , na.rm = TRUE),
                  delay2=mean(DepDelay, na.rm = TRUE))
```

(3)Data Visualization

```
ggplot(data = delay2) +
   geom_point(mapping = aes(x=flytime,y=delay2))
```



```
   geom_smooth(method='lm',mapping = aes(x=flytime,y=delay2))
```

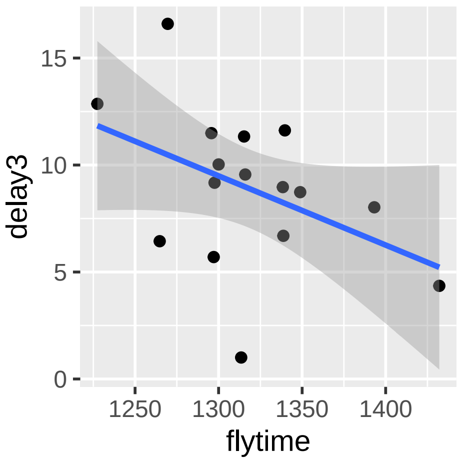
(4)Consider this change by airline?

Group by UniqueCarrier

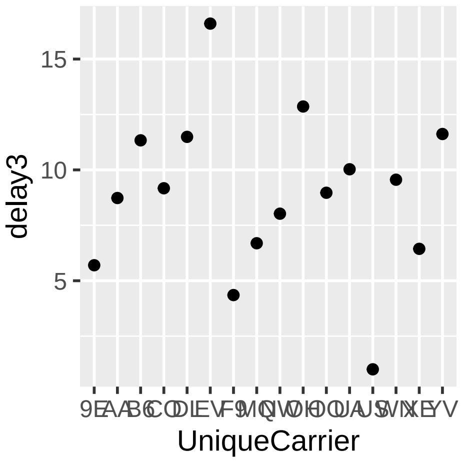
```
by_UniqueCarrier<-group_by(ABIA2,UniqueCarrier)
delay3<-summarise(by_UniqueCarrier,count = n(),
                  flytime=mean(CRSDepTime , na.rm = TRUE),
                  delay3=mean(DepDelay, na.rm = TRUE))
```

Data Visualization

```r
ggplot(data = delay3) +
  geom_point(mapping = aes(x=flytime,y=delay3)) +
   geom_smooth(method='lm',mapping = aes(x=flytime,y=delay3))
```



```r
ggplot(data = delay3) +
   geom_point(mapping = aes(x=UniqueCarrier,y=delay3)) +
   geom_smooth(method='lm',mapping = aes(x=UniqueCarrier,y=delay3))
```



Conclusion: As we can see in the graph, the best time of day to minimize delays is before 1000 and after 2000 (most of the average delay time is lower than 25). The airline could not change this because peopel would more likely to fly by day.

(5)Consider the correlation between fly distance and flight delay

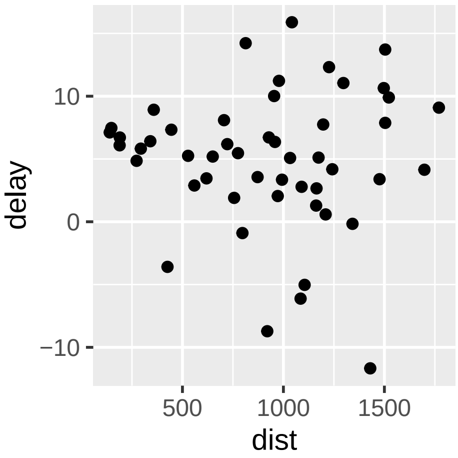
Select the variables

```r
ABIA1<-select(ABIA,Year,Month,DayofMonth,DayOfWeek,DepDelay,ArrDelay,Distance,Dest)
ABIA1<-filter(ABIA1, !is.na(DepDelay), !is.na(ArrDelay))
by_Dest<-group_by(ABIA1,Dest)
delay<-summarise(by_Dest,count = n(),
                 dist=mean(Distance , na.rm = TRUE),
                 delay=mean(ArrDelay, na.rm = TRUE))
```

Remove noise data

```
delay<-filter(delay,count > 20)
```

```
ggplot(data = delay) +
   geom_point(mapping = aes(x=dist,y=delay))
```



```
   geom_smooth(method = 'loess',mapping = aes(x=dist,y=delay))
```

Conclusion As we can see in the graph, when the distance is 0-1200, the average delay time have a negative trend with the distance, when the distance is larger than 1200, the average delay time have a positive trend with the distance.

4 K-nearest neighbors

(1)Packages and Datasets

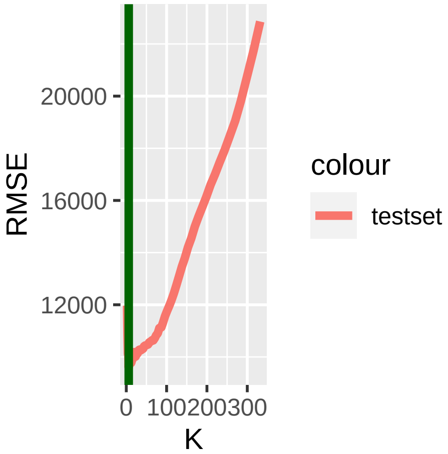
```r
library(mosaic)
library(tidyverse)
library(FNN)
library(foreach)
sclass <- read.csv("data/sclass.csv", stringsAsFactors=TRUE)
sclass350 = subset(sclass, trim == '350')
```

(2)Create a train/test split

```r
N = nrow(sclass350)
N_train = floor(0.8*N)
train_ind = sample.int(N, N_train, replace=FALSE)

sclass350_train = sclass350[train_ind,]
sclass350_test = sclass350[-train_ind,]

y_train_350 = sclass350_train$price
X_train_350 = data.frame(mileage = sclass350_train$mileage)
y_test_350 = sclass350_test$price
X_test_350 = data.frame(mileage = sclass350_test$mileage)

rmse = function(y, ypred) {
  sqrt(mean((y-ypred)^2))
}
k_grid = unique(round(exp(seq(log(N_train), log(2), length=100))))
rmse_grid_out = foreach(k = k_grid, .combine='c') %do% {
  knn_model = knn.reg(X_train_350, X_test_350, y_train_350, k = k)
  rmse(y_test_350, knn_model$pred)
}
rmse_grid_out = data.frame(K = k_grid, RMSE = rmse_grid_out)
```

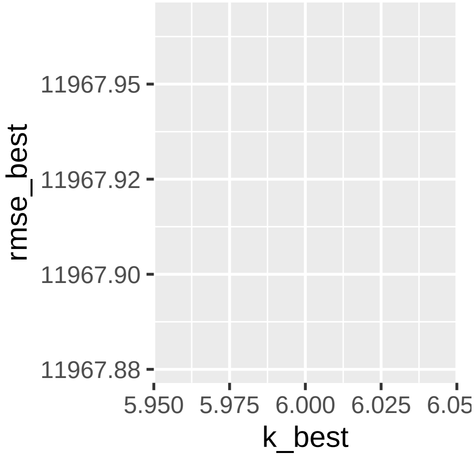
(3)Visualization

```
ind_best = which.min(rmse_grid_out$RMSE)
k_best = k_grid[ind_best]
ggplot(data=rmse_grid_out) +
geom_path(aes(x=K, y=RMSE, color='testset'), size=1.5) +
geom_vline(xintercept=k_best, color='darkgreen', size=1.5)
```



(4)Fit the model at the optimal k

```
knn_model_bestk = knn.reg(X_train_350, X_test_350, y_train_350, k = k_best)
rmse_best = rmse(y_test_350, knn_model$pred)
ggplot(data=rmse_grid_out)+
geom_path(mapping = aes(x=k_best, y=rmse_best), color='red', size=1.5)
```

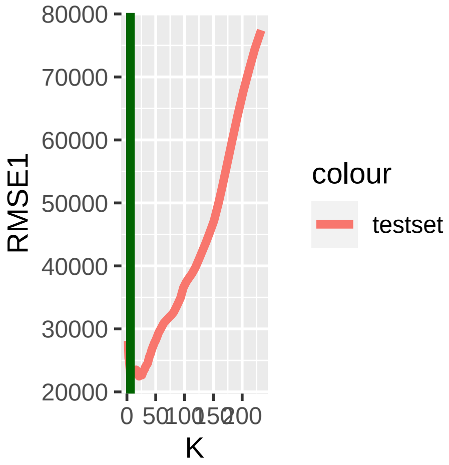


(5)Focus on second trim level: 65 AMG

```
sclass65AMG = subset(sclass, trim == '65 AMG')
# create a train/test split
N = nrow(sclass65AMG)
N_train = floor(0.8*N)
train_ind = sample.int(N, N_train, replace=FALSE)
```

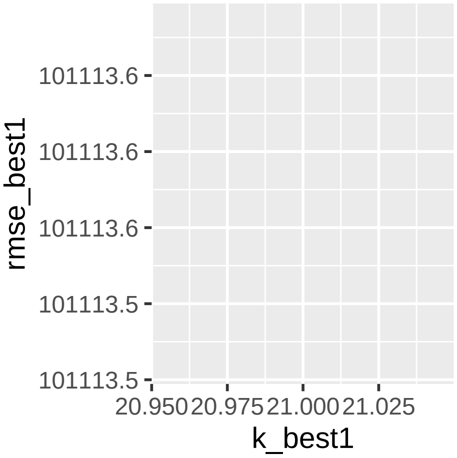
```
sclass65AMG_train = sclass65AMG[train_ind,]
sclass65AMG_test = sclass65AMG[-train_ind,]

y_train_65AMG = sclass65AMG_train$price
X_train_65AMG = data.frame(mileage = sclass65AMG_train$mileage)
y_test_65AMG = sclass65AMG_test$price
X_test_65AMG = data.frame(mileage = sclass65AMG_test$mileage)


rmse1 = function(y, ypred) {
  sqrt(mean((y-ypred)^2))
}

k_grid1 = unique(round(exp(seq(log(N_train), log(2), length=100))))
rmse1_grid_out = foreach(k = k_grid1, .combine='c') %do% {
  knn_model1 = knn.reg(X_train_65AMG, X_test_65AMG, y_train_65AMG, k = k)
  rmse1(y_test_65AMG, knn_model1$pred)
}

rmse1_grid_out = data.frame(K = k_grid1, RMSE1 = rmse1_grid_out)
```

```
ind_best = which.min(rmse1_grid_out$RMSE1)
k_best1 = k_grid1[ind_best]
ggplot(data=rmse1_grid_out) +
geom_path(aes(x=K, y=RMSE1, color='testset'), size=1.5) +
geom_vline(xintercept=k_best, color='darkgreen', size=1.5)
```



(6)Fit the model at the optimal k

```
knn_model_besk1 = knn.reg(X_train_65AMG, X_test_65AMG, y_train_65AMG, k = k_best1)
rmse_best1 = rmse(y_test_65AMG, knn_model$pred)
ggplot(data=rmse1_grid_out)+
geom_path(mapping = aes(x=k_best1, y=rmse_best1), color='blue', size=1.5)
```



Conclusion: $K\_best > k\_best1$, 350 has a larger optimal value of k.