Excercise1

Xiaohan Sun / Liyuan Zhang / Evelyn Cheng

2021/2/6

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

## 1) Data visualization: gas prices

library(tidyverse)

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

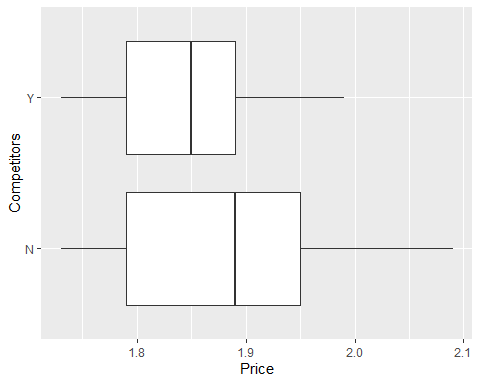
## √ ggplot2 3.3.3 √ purrr 0.3.4  
## √ tibble 3.0.6 √ dplyr 1.0.3  
## √ tidyr 1.1.2 √ stringr 1.4.0  
## √ readr 1.4.0 √ forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## 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 Intersection  
## 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))



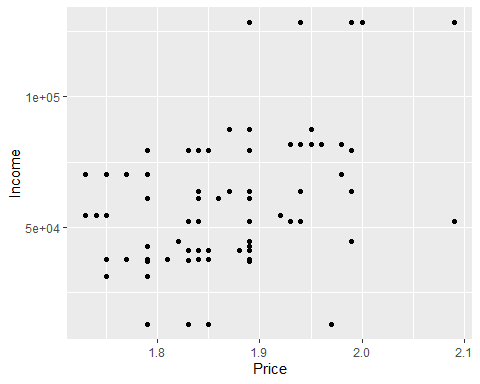
**Claim:** Gas stations charge more if they lack direct competition in sight.

**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

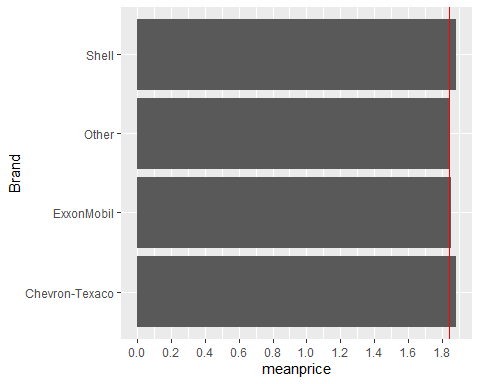
ggplot(data=GasPrices) +   
 geom\_point(aes(x=Price, y=Income))+  
 ylim(12780,128560)



**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.

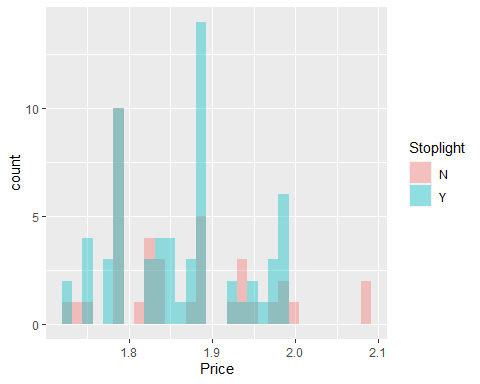
GasPricesC = GasPrices %>%  
 group\_by(Brand) %>%  
 summarise(meanprice=mean(Price))  
ggplot(data=GasPricesC) +   
 geom\_col(aes(x=meanprice, y=Brand)) +   
 scale\_x\_continuous(breaks=seq(0,2,0.2)) +  
 geom\_vline(xintercept=1.84,col="red")



**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.

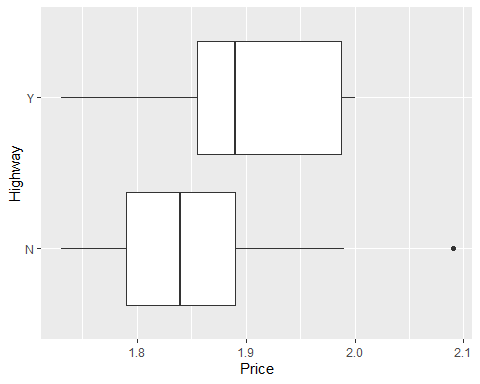
ggplot(data=GasPrices,aes(x=Price,fill=Stoplight))+geom\_histogram(bins = 30, alpha=0.4, position = "identity")



**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.

## 4) K-nearest neighbors

1. 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 order to add   
## additional features. The original behavior of these functions should 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'

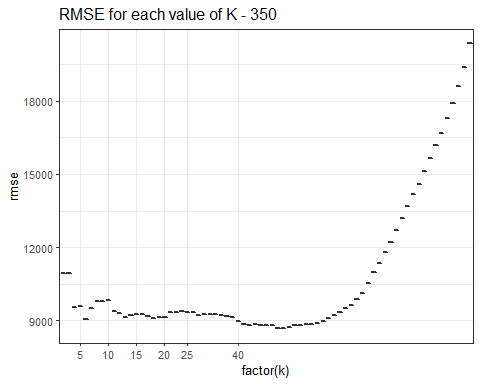
## The following object is masked from 'package:mosaic':  
##   
## resample

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

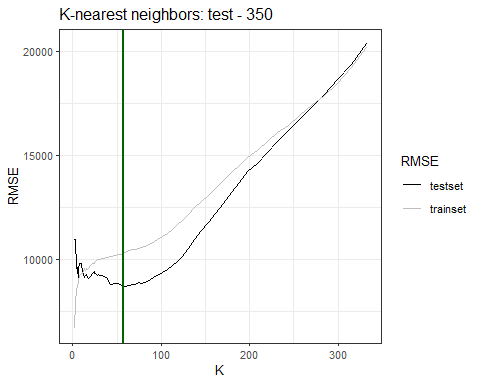
library(parallel)  
  
sclass = read.csv('../data/sclass.csv')  
  
sclass350 = subset(sclass, trim == '350')  
  
# Split the data into a training and a testing set  
sclass350\_split = initial\_split(sclass350, prop=0.9)  
sclass350\_train = training(sclass350\_split)  
sclass350\_test = testing(sclass350\_split)  
  
# 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, use.all=TRUE)  
 modelr::rmse(knn\_model, sclass350\_test)  
 }  
 data.frame(k=k\_grid, rmse=this\_rmse)  
}

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

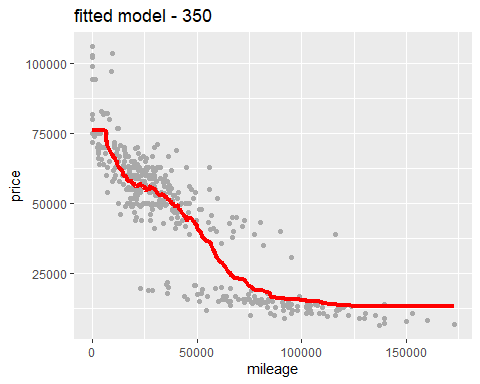
rmse\_out = arrange(rmse\_out, k)  
ggplot(rmse\_out) +   
 geom\_boxplot(aes(x=factor(k), y=rmse)) +   
 theme\_bw(base\_size=10) +  
 scale\_x\_discrete(breaks=c(5,10,15,20,25,30,40,50,80,100)) +  
 labs (titles = "RMSE for each value of K - 350")



# 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")

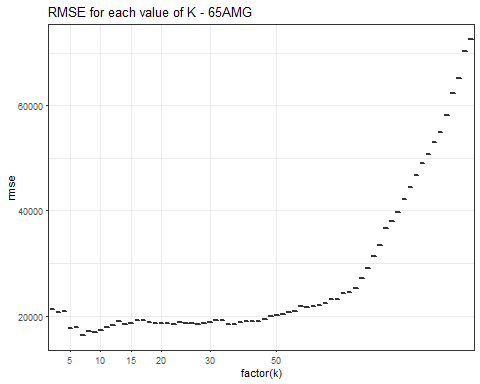


# fitted model  
knn = knnreg(price ~ mileage, data=sclass350\_train, k=k\_best)  
sclass350 = sclass350 %>%  
 mutate(price\_pre = predict(knn, sclass350))  
g350 = ggplot(data = sclass350) +   
 geom\_point(mapping = aes(x = mileage, y = price), color='darkgrey')  
g350 + geom\_line(aes(x = mileage, y = price\_pre), color='red', size=1.5) +  
 labs (titles = "fitted model - 350")

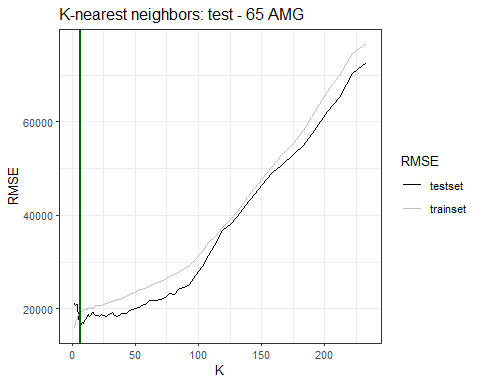


1. 65 AMG

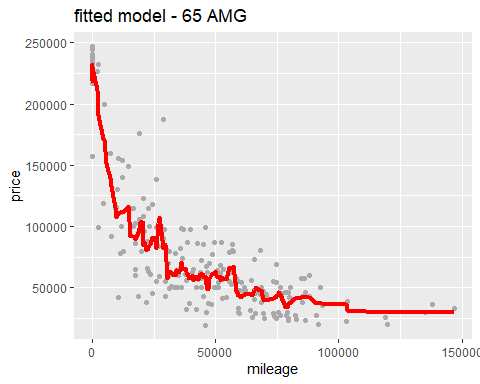
# Split the data into a training and a testing set  
sclass65AMG = subset(sclass, trim == '65 AMG')  
  
sclass65AMG\_split = initial\_split(sclass65AMG, prop=0.9)  
sclass65AMG\_train = training(sclass65AMG\_split)  
sclass65AMG\_test = testing(sclass65AMG\_split)  
  
# RMSE for each value of K  
N65AMG = nrow(sclass65AMG)  
N\_train65AMG = floor(0.8\*N65AMG)  
k\_grid65AMG = unique(round(exp(seq(log(N\_train65AMG), log(2), length=100))))  
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, use.all=TRUE)  
 modelr::rmse(knn\_model, sclass65AMG\_test)  
}  
rmse\_grid\_out65AMG = data.frame(K = k\_grid65AMG, RMSE = rmse\_grid\_out65AMG)  
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, use.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\_best65AMG)  
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] 57

k\_best65AMG

## [1] 7

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.