PS1

Seunghoon Choi, Yuri Lee

2021 2 4

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

# 1. Gas Price

library(tidyverse) library(ggplot2)

## A. Competition & Price

ggplot(data=GasPrices) + geom\_boxplot(aes(x = Competitors, y=Price)) ### The bar plot above shows the gas prices of gas providers which have competitors is lower than the price of the providers which do not have competitors.

ggplot(data=GasPrices) + geom\_boxplot(aes(x = Competitors, y=Price)) + facet\_wrap(~Name, nrow=2) ### However, it would be hard to generalize the relation between the price and the existence of competitors. Only three providers show lower price when they have competitors than without-competitors cases among the eight eligible cases out of twenty, whose with-competitor prices and without-copetitors prices can be compared.

## B. Income & Price

ggplot(data=GasPrices) + geom\_point(mapping = aes(x=Income, y=Price)) ### We can see upward shape of dots in this graph, which means the gas prices and income of people who live in those areas where the gas stations are located have a positive relation.

ggplot(data=GasPrices) + geom\_point(mapping = aes(x=Income, y=Price)) + facet\_wrap(~Name, nrow=4) ### On the graph of each company, several companies such as 7-Eleven, Exxon, and Shell represent these positive relations obviously. On the contrary, we can see that some companies like Costco sticks to one-price polices.

## C. Price of Shell vs Other sellers

GasPrices = GasPrices %>% mutate(class = ifelse(Name == ‘Shell’, ‘Shell’, ‘others’))

d1 = GasPrices %>% group\_by(class) %>% summarise(mean\_price = mean(Price)) d1

ggplot(data=d1) + geom\_col(mapping=aes(x=class, y=mean\_price), position = ‘dodge’) ### The average gas price of Shell(1.88) is a little bit higher than that of other providers’ one(1.86).

d2 = GasPrices %>% group\_by(Name) %>% summarise(mean\_price = mean(Price)) d2

ggplot(data=d2) + geom\_col(mapping=aes(x=Name, y=mean\_price), position = ‘dodge’) ### The 12 out of 19 providers have lower average gas prices than that of Shell.

## d. stoplights’ effects on Price

ggplot(data = GasPrices) + geom\_histogram(aes(x=Price)) + facet\_wrap(~ Stoplight) ### Gas stations nearby stoplights generally have higher gas prices. Prices of gas stations without stoplight nearby are concentrated around 1.8 the most, while gas stations near stoplight have a lot of cases around 1.8~1.9.

## e. The effect of Highway access on Price

d3 = GasPrices %>% group\_by(Highway) %>% summarise(mean\_price = mean(Price)) d3

ggplot(data=d3) + geom\_col(mapping = aes(x=Highway, y=mean\_price)) ### Gas stations which is accesible to highways tend to set gas prices higher than gas stations which is far from highways.

d4 = GasPrices %>% group\_by(Highway, Name) %>% summarize(mean\_price = mean(Price)) d4

ggplot(data = d4) + geom\_col(mapping = aes(x = Name,y = mean\_price, fill=Highway), position = ‘dodge’) + facet\_wrap(~Name, nrow=4)

ggplot(data=GasPrices) + geom\_boxplot(aes(x = Highway, y=Price)) + facet\_wrap(~Name, nrow=2) ### These plots show that companies usually set the higher gas prices for the highway accessible gas stations.

# 2. Bike Share

## plot A

bikeshare\_a = bikeshare %>% group\_by(hr) %>% summarise(average\_rental=mean(total))

head(bikeshare\_a)

ggplot(data=bikeshare\_a) + geom\_line(aes(x=hr, y=average\_rental)) + scale\_x\_continuous(breaks=0:23) + labs(title=“average bike rentals per hour”) ### A lot of bike share users rented bikes during rush hours, around 8:00, 17:00.

## plot B

bikeshare\_b = bikeshare %>% mutate(work = ifelse(workingday==1, “working”, “dayoff”)) %>% group\_by(hr, work) %>% summarise(average\_rental=mean(total))

head(bikeshare\_b)

ggplot(data=bikeshare\_b) + geom\_line(aes(x=hr, y=average\_rental)) + facet\_wrap(~work) + scale\_x\_continuous(breaks=0:23) + labs(title=“average bike rentals per hour: working or not”) ### In working days, most people used bikes during rush hours, around 8:00, 17:00, while, in day offs, most people rented bikes in the afternoon.

## plot C

bikeshare\_c = bikeshare %>% filter(hr==8) %>% mutate(work = ifelse(workingday==1, “working”, “dayoff”)) %>% group\_by(weathersit, work) %>% summarise(average\_rental=mean(total))

ggplot(data=bikeshare\_c) + geom\_col(aes(x=factor(weathersit), y=average\_rental)) + facet\_wrap(~work) + labs(title=“average bike rentals at 8:00 under weather situation: working or not”) ### A bike rental demand at 8:00 in working days are approximately 5 times larger than that in dayoffs. When the weather is bad, the demand decreses by half in both working days and dayoffs.

# 3. Flight at ABIA: Which day of a week is the worst departure/arrival(long delay) in Austin?

## Departure Delay

d1 = ABIA %>% filter(Origin == ‘AUS’) %>% filter(!is.na(DepDelay)) %>% group\_by(DayOfWeek) %>% summarise(mean\_depdelay = mean(DepDelay)) d1

ggplot(data=d1) + geom\_col(aes(x=DayOfWeek, y=mean\_depdelay), position = ‘dodge’) ### If you leave from Austin by airplain, Wednesday is the best, which you can minimize your departure delay, the average departure delay is 5 minutes, while Friday gives the longest delay.

d2 = ABIA %>% filter(Origin == ‘AUS’) %>% filter(!is.na(DepDelay)) %>% group\_by(DayOfWeek, UniqueCarrier) %>% summarise(mean\_depdelay = mean(DepDelay)) d2

ggplot(data=d2) + geom\_col(aes(x=DayOfWeek, y=mean\_depdelay), position = ‘dodge’) + facet\_wrap(~UniqueCarrier) ### However, each airline has different delay pattern by day of week. So, if you plan airline trip, you might need to consider which day of week is best and worst for your airline.

## Arrival Delay

d3 = ABIA %>% filter(Dest==‘AUS’) %>% filter(!is.na(ArrDelay)) %>% group\_by(DayOfWeek) %>% summarise(mean\_ArrDelay = mean(ArrDelay)) d3

ggplot(data=d3) + geom\_col(aes(x=DayOfWeek, y=mean\_ArrDelay), position = ‘dodge’) ### The arrival delay is also the longest on Friday like the departure delay in Austin.

d4 = ABIA %>% filter(Dest==‘AUS’) %>% filter(!is.na(ArrDelay)) %>% group\_by(DayOfWeek, UniqueCarrier) %>% summarise(mean\_ArrDelay = mean(ArrDelay)) d4

ggplot(data=d4) + geom\_col(aes(x=DayOfWeek, y=mean\_ArrDelay), position = ‘dodge’) + facet\_wrap(~UniqueCarrier) ### Each airline has different shape of arrival delay by day of week. The interesting thing is NW airline shows high peak in departure and arrival delay in the middle of week, while US airline has very low, and stable delay.

# 4. K-nearest neighbors

library(tidyverse) library(ggplot2) library(rsample) library(caret) library(modelr) library(parallel) library(foreach)

## filtering two trims

sclass %>% filter(trim==“350” | trim==“65 AMG”)

s3 = subset(sclass, trim==“350”) s6 = subset(sclass, trim==“65 AMG”)

plot(price ~ mileage, data = s3) plot(price ~ mileage, data = s6)

## split data

s3\_split = initial\_split(s3, prop=0.8) s3\_train = training(s3\_split) s3\_test = testing(s3\_split)

s6\_split = initial\_split(s6, prop=0.8) s6\_train = training(s6\_split) s6\_test = testing(s6\_split)

## KNN for trim 350

## k=2,5,10,15,20,50,100,200

knn = knnreg(mileage ~ price, data=s3\_train, k=2) knn5 = knnreg(mileage ~ price, data=s3\_train, k=5) knn10 = knnreg(mileage ~ price, data=s3\_train, k=10) knn15 = knnreg(mileage ~ price, data=s3\_train, k=15) knn20 = knnreg(mileage ~ price, data=s3\_train, k=20) knn50 = knnreg(mileage ~ price, data=s3\_train, k=50) knn100 = knnreg(mileage ~ price, data=s3\_train, k=100) knn200 = knnreg(mileage ~ price, data=s3\_train, k=200)

s3\_test = s3\_test %>% mutate(price\_pred = predict(knn2, s3\_test)) %>% mutate(price\_pred = predict(knn5, s3\_test)) %>% mutate(price\_pred = predict(knn10, s3\_test)) %>% mutate(price\_pred = predict(knn15, s3\_test)) %>% mutate(price\_pred = predict(knn20, s3\_test)) %>% mutate(price\_pred = predict(knn50, s3\_test)) %>% mutate(price\_pred = predict(knn100, s3\_test)) %>% mutate(price\_pred = predict(knn200, s3\_test)) %>%

## Calculating RMSE

modelr::rmse(knn2, s3\_test) modelr::rmse(knn5, s3\_test) modelr::rmse(knn10, s3\_test) modelr::rmse(knn15, s3\_test) modelr::rmse(knn20, s3\_test) modelr::rmse(knn50, s3\_test) modelr::rmse(knn100, s3\_test) modelr::rmse(knn200, s3\_test)

## KNN for 65 AMG

## k=2,5,10,15,20,50,100,200

knn2 = knnreg(mileage ~ price, data=s6\_train, k=2) knn5 = knnreg(mileage ~ price, data=s6\_train, k=5) knn10 = knnreg(mileage ~ price, data=s6\_train, k=10) knn15 = knnreg(mileage ~ price, data=s6\_train, k=15) knn20 = knnreg(mileage ~ price, data=s6\_train, k=20) knn50 = knnreg(mileage ~ price, data=s6\_train, k=50) knn100 = knnreg(mileage ~ price, data=s6\_train, k=100) knn200 = knnreg(mileage ~ price, data=s6\_train, k=200)

s6\_test = s6\_test %>% mutate(price\_pred = predict(knn2, s6\_test)) %>% mutate(price\_pred = predict(knn5, s6\_test)) %>% mutate(price\_pred = predict(knn10, s6\_test)) %>% mutate(price\_pred = predict(knn15, s6\_test)) %>% mutate(price\_pred = predict(knn20, s6\_test)) %>% mutate(price\_pred = predict(knn50, s6\_test)) %>% mutate(price\_pred = predict(knn100, s6\_test)) %>% mutate(price\_pred = predict(knn200, s6\_test)) %>%

## Calculating RMSE

modelr::rmse(knn2, s6\_test) modelr::rmse(knn5, s6\_test) modelr::rmse(knn10, s6\_test) modelr::rmse(knn15, s6\_test) modelr::rmse(knn20, s6\_test) modelr::rmse(knn50, s6\_test) modelr::rmse(knn100, s6\_test) modelr::rmse(knn200, s6\_test)