midtermDoc

Young Jin LEE, 21300551

4/18/2018

## Basic R skills

### 1 파일로부터 데이터를 R로 읽어오시오. 데이터 프레임 이름은 car\_df

### 2 변수는 몇 개며 observation은 몇 개 입니까?

dim(car\_df)

## [1] 428 19

#### 변수는 19개, observation은 428개가 있다.

### 3 name columns은 자동차 모델의 이름을 담고 있습니다. character와 factor type 중 어떤 것이 적절할까요? 필요하다면 type 변환을 하시오.

#car\_df[car\_df$name == 'Infiniti G35 4dr',]  
car\_df$name <- as.factor(car\_df$name)  
#car\_df$ncyl <- as.factor(car\_df$ncyl)

#### 같은 모델의 이름이 여러개 있는 경우가 있으며, 나중에 브랜드 별로 나눠서 분석을 할 수도 있기 때문에, factor type이 적절하다.

### 4 mrsp는 소비자 권장 가격입니다. dealer\_cost와 비교해서 평균적으로 차이가 얼마나 나나요?

mean(car\_df$msrp)

## [1] 32774.86

mean(car\_df$dealer\_cost)

## [1] 30014.7

mean(car\_df$msrp) - mean(car\_df$dealer\_cost)

## [1] 2760.154

#### 소비자 권장가격은 32774.86, dealer\_cost는 30014.7로, 소비자 권장가격이 dealer\_cost보다 2760.154 높다.

### 5 city\_mpg값이 가장 큰 자동차 모델은 무엇인가요? 그 차의 city\_mpg 와 hwy\_mpg의 차이는 얼마인가요? 이 차는 hwy\_mpg가 가장 높은 차와 같은 차종인가 요?

idx <- which.max(car\_df$city\_mpg)  
  
car\_df[idx,]$name

## [1] Honda Insight 2dr (gas/electric)  
## 425 Levels: Acura 3.5 RL 4dr Acura 3.5 RL w/Navigation 4dr ... Volvo XC90 T6

car\_df[idx,]$city\_mpg

## [1] 60

car\_df[idx,]$hwy\_mpg

## [1] 66

car\_df[idx,]$hwy\_mpg - car\_df[idx,]$city\_mpg

## [1] 6

idx2 <- which.max(car\_df$hwy\_mpg)  
car\_df[idx2,]$name

## [1] Honda Insight 2dr (gas/electric)  
## 425 Levels: Acura 3.5 RL 4dr Acura 3.5 RL w/Navigation 4dr ... Volvo XC90 T6

car\_df[idx,]$name == car\_df[idx2,]$name

## [1] TRUE

#### Honda Insight 2dr (gas/electric) 가 city\_mpg 값이 60으로 가장 높다. 이 차의 hwy\_mpg는 66, city\_mpg는 60으로 hwy\_mpg가 6 높다. 이 차는 hwy\_mpg가 가장 높은 차와 같은 차종이다.

### 6 각각의 자동차 종류(sport car, suv, wagon, minivan, pickup)마다자동차 모델이 몇 개씩 있나요? 어떤 종류에도 속하지 않는 자동차는 몇 개나 있나요?

sapply(car\_df[,c(2:6)], function(x){sum(x)})

## sports\_car suv wagon minivan pickup   
## 49 60 30 20 24

a <- car\_df[car\_df$sports\_car == FALSE, ]  
b <- a[a$suv == FALSE, ]  
c <- b[b$wagon == FALSE, ]  
d <- c[c$minivan == FALSE, ]  
e <- d[d$pickup == FALSE, ]  
dim(e)[1]

## [1] 245

# 자동차 모델 별 sports\_car: 49, suv: 60, wagon: 30, minivan:20, pickup:24 개가 있다.  
# 어떤 종류에도 속하지 않는 자동차는 245개이다.

#### 자동차 모델 별 sports\_car: 49, suv: 60, wagon: 30, minivan:20, pickup:24 개가 있다. 어떤 종류에도 속하지 않는 자동차는 245개이다.

### 7 SUV와 minivan 차종의 무게를 비교하시오. 평균적으로 어떤 차종이 더 무겁나요?

suv <- car\_df[car\_df$suv == T,]  
summary(suv$weight)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2866 3787 4438 4444 4990 7190

# SUV 무게 분포  
suv\_weight <- mean(suv$weight)   
suv\_weight

## [1] 4444.433

minivan <- car\_df[car\_df$minivan == T,]  
summary(minivan$weight)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3699 3996 4225 4194 4346 4802

# minivan 무게 분포  
minivan\_weight <- mean(minivan$weight)  
minivan\_weight

## [1] 4193.6

suv\_weight - minivan\_weight

## [1] 250.8333

#mean(suv$weight - minivan$weight)

#### SUV의 무게 평균은 4444.433, minivan의 평균은 4193.6 으로 SUV의 무게가 평균적으로 250.8333 높다.

### 8 새로운 column “avg\_mpg”를 추가하시오. avg\_mpg는 city\_mpg와 hwy\_mpg의 평균입니다.

car\_df$avg\_mpg <- (car\_df$city\_mpg + car\_df$hwy\_mpg) / 2

#### 새로운 column “avg\_mpg”를 추가함

### 9 “eco\_grade”라는 columns을 추가하시오. avg\_mpg 상위 20%에는 “good” 하위 20%에는 “bad”, 나머지에는 “normal” 값을 부여하시오.

cut\_points <- quantile(car\_df$avg\_mpg, c(0, 0.2, 0.8, 1), na.rm = TRUE)  
car\_df$eco\_grade <- cut(car\_df$avg\_mpg, breaks = cut\_points, include.lowest = T)  
#head(car\_df[,c('avg\_mpg', 'eco\_grade')])  
(levels(car\_df$eco\_grade) <- c('bad', 'normal', 'good'))

## [1] "bad" "normal" "good"

#head(car\_df[,c('avg\_mpg', 'eco\_grade')],8)  
#table(car\_df$eco\_grade)

#### “eco\_grade”라는 columns을 추가함. avg\_mpg 상위 20%에는 “good” 하위 20%에는 “bad”, 나머지에는 “normal” 값을 부여함

### 10 4륜 구동 자동차와 후륜구동 자동차의 마력을 비교하시오

allwheel <- car\_df[car\_df$all\_wheel == T,]  
rearwheel <- car\_df[car\_df$rear\_wheel == T,]  
# 4륜 구동 자동차의 마력 분포  
summary(allwheel$horsepwr)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 130.0 190.0 222.5 235.1 295.0 345.0

# 후륜 구동 자동차의 마력 분포  
summary(rearwheel$horsepwr)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 130.0 215.0 239.0 262.6 302.0 500.0

mean(rearwheel$horsepwr) - mean(allwheel$horsepwr)

## [1] 27.4749

#### 4륜 구동 자동차의 평균 마력은 약 235.0978, 후륜구동 자동차의 평균 마력은 약 262.5727이다. 따라서 평균적으로 후륜 구동 자동차의 마력이 27.4749 높다고 할 수 있다.

## Data Science Project Pipeline

### 1 데이터 과학자에게 communication과 teamwork 능력이 중요한 이유는 무엇인가요?

#### 데이터 과학은 생명, 경영, 마케팅, 산업공학 등 많은 분야에서 필요로 한다. 데이터 과학자가 모든 도메인 지식을 정통하는 것은 불가능에 가깝기 때문에 해당 프로젝트마다 도메인 사람들과의 협업이 중요시 된다. 서로 자주 소통함으로써 데이터에 대한 이해를 높일 수 있고, 더 의미 있고 가치있는 결과가 나올 가능성이 커진다. 또, 데이터 과학 팀 내에서도 커뮤니케이션과 팀워크를 통해 많은 시간과 노력을 줄일 수 있다. 예를 들어, 각자가 하는 작업을 다른 팀원이 쉽게 알아볼 수 있도록 하는 것이 중요하며, 미팅을 통해 방향을 잃지 않고 한 공간에 결과물을 일관성있게 결과물을 정리해 놓음으로써 효율적이게 업무를 수행할 수 있다.

### 2 데이터 탐색 단계에 대해 설명하시오. 그것이 왜 필요한가요?

#### 데이터 탐색은 해당 데이터의 성격과 데이터가 담고 있는 특징을 파악하도록 도와준다. 데이터가 어떻게 저장되어있는지부터 각 변수, 케이스가 의미하는 것이 무엇인지, 또 데이터의 전반적인 특징을 파악함으로써 향후 발생 할 수 있는 문제를 줄여주며 데이터를 이해하는데 도움을 준다. 예를 들면, 수치형 이어야 하는 데이터에 결측치가 NA가 아닌 character형태로 “NA”라고 들어가 있는 경우가 있다. 이 경우, R에서는 이 변수를 integer가 아닌 character, 혹은 factor로 읽어들이는데, 어떤 값이 들어가 있는지 확인을 하고 “NA”를 따로 처리해 주어야한다. 만약 탐색과정에서 이 부분을 발견하지 못했다면, 향후 문제가 될 가능성이있다. 이 뿐만 아니라, scatter plot, box plot등 시각화를 통해 경향성을 파악할 수 있다. 평균, 최대 값, 최소 값등을 확인 할 수 있으며, 더 나아가 feature engineering을 하는데도 도움이 된다.

### 3 client에게서 받은 data를 cleaning 해야하는 이유는 무엇입니까?

#### 데이터 내의 노이즈를 제거하고 일관성 결여를 교정하며 결측 값을 처리하고 불필요한 변수를 제거함으로써 일관성있는 데이터 분석을 할 수 있기 때문에 data cleaning을 해야한다. 많은 경우 client에게서 받은 data는 cleaning을 할 필요가 있다.

## Data Cleaning

### 1 dataset은 tidy인가 아닌가? 이유를 설명하시오.

#### 날짜 별 측정된 날씨 정보인데, 연,월,일 중 일자가 변수로 들어가 있으며, 숫자가 아닌 앞에 X값이 들어가 있다. 또, NA 값과 비어있는 셀이 존재한다. 또, 범주형 변수가 character형태로 들어가 있어 요약 확인 시 어려움이 있다. 따라서 tidy한 형태의 dataset이라고 할 수 없다.

### 2 dataset이 tidy가 아니라면 tidy한 형태로 변환하시오.

### 3 데이터셋에 불필요한 column이 있는가? 있다면 무엇인가? 불필요한 column을 제거하시오

weather.tidy <- weather %>% gather(day, Value, -year, -month, -measure, -X)  
weather.tidy$day <- gsub("X", "", weather.tidy$day)  
  
weather.tidy$day <- str\_pad(weather.tidy$day, width = 2, side = "left", pad = "0")  
  
weather.tidy$X <- NULL   
weather.tidy <- weather.tidy %>% spread(measure, Value)

### 4 데이터에 year month day 세 column이 있는데 이를 하나로 합쳐서 date column을 추가하시오. date column은 Date data type이어야합니다. 그리고 year month day 세 column은 제거하시오

weather.tidy <- weather.tidy %>% unite(date, year,month,day)  
weather.tidy$date <- ymd(weather.tidy$date)

### 5 PrecipitationIn(강수량) 변수를 보면 “T”라는 값이 있는데 이는 Trace 비가 아주 미량왔다는 의미이다. 해당 변수를 숫자형으로 변환할 수 있도록, “T”를 숫자 0으로 변환하시오.

idx <- which(str\_detect(weather.tidy$PrecipitationIn, "T"))  
weather.tidy[idx,]$PrecipitationIn <- 0

### 7 데이터셋에 missing values가 있나요? 몇 개나 있나요? 각 변수 별로 몇 개씩 있나요?

sum(is.na(weather.tidy))

## [1] 827

sapply(weather.tidy, function(x) {sum(is.na(x))})

## date CloudCover   
## 7 37   
## Events Max.Dew.PointF   
## 37 37   
## Max.Gust.SpeedMPH Max.Humidity   
## 43 37   
## Max.Sea.Level.PressureIn Max.TemperatureF   
## 37 37   
## Max.VisibilityMiles Max.Wind.SpeedMPH   
## 37 37   
## Mean.Humidity Mean.Sea.Level.PressureIn   
## 37 37   
## Mean.TemperatureF Mean.VisibilityMiles   
## 37 37   
## Mean.Wind.SpeedMPH MeanDew.PointF   
## 37 37   
## Min.DewpointF Min.Humidity   
## 37 37   
## Min.Sea.Level.PressureIn Min.TemperatureF   
## 37 37   
## Min.VisibilityMiles PrecipitationIn   
## 37 37   
## WindDirDegrees   
## 37

weather.tidy <- weather.tidy[complete.cases(weather.tidy), ] ## 결측치가 있는 케이스 삭제

#### 데이터셋에 missing value는 총 827개가 있다. date 변수에는 7개, Max.Fust.SpeedMPH 변수에는 43개 그리고 나머지 모든 변수에 missing value가 37개 씩 있다. 결측치가 있는 케이스는 삭제하였다.

### 6 각 변수의 data type을 적절한 것으로 변환하시오.

weather.tidy$CloudCover <- as.factor(weather.tidy$CloudCover)  
weather.tidy[,4:23] <- sapply(weather.tidy[,4:23], function(x) {as.integer(x)})

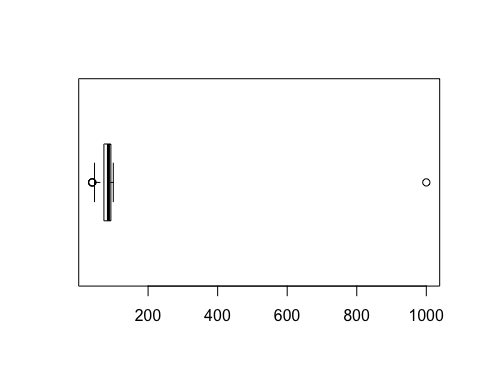
#### date는 Date type, cloudcover 는 0~8까지 factor 형 변수이며, Events는 character 형 이라 생각하며 나며지는 수치형 이기 때문에 그에 알맞게 변환하였다.

### 8 Max.Humidity(최대 습도) 변수를 보시오. outlier가 있나요? outlier 값이 실수로 0이 하나 더 붙어 나온 값이라고 합시다. 해당 outlier를 적절한 값으로 고치시오

par(mfrow = c(1,1))  
summary(weather.tidy$Max.Humidity)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 39.00 73.00 86.00 85.64 93.00 1000.00

boxplot(weather.tidy$Max.Humidity, horizontal = TRUE)



outlier\_values <- boxplot.stats(weather.tidy$Max.Humidity)$out  
outlier\_values

## [1] 39 40 39 1000

length(outlier\_values)

## [1] 4

indexx <- which(weather.tidy$Max.Humidity == 1000)  
weather.tidy[indexx,]$Max.Humidity <- 100

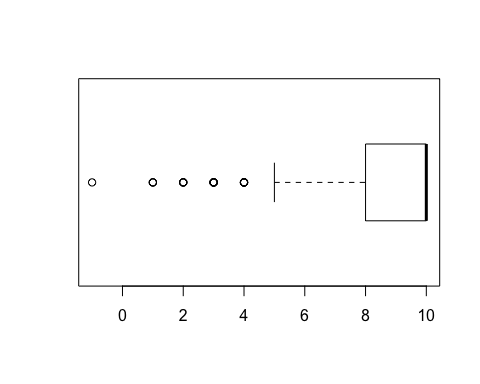
#### 1.5IQR rule에 의한 outlier 값은 39, 40, 1000이 4개 있다. outlier값이 실수로 0이 하나 더 붙어 나온 값이라면 그에 해당하는 값은 1000으로 판단되어 이 값을 100으로 바꾸어 주었다.

### 9 Mean.VisibilityMiles(평균시야거리) 변수를 보시오. outlier가 있나요? outlier를 적절한 값으로 고치시오.

summary(weather.tidy$Mean.VisibilityMiles)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -1.000 8.000 10.000 8.847 10.000 10.000

boxplot(weather.tidy$Mean.VisibilityMiles, horizontal = TRUE)



outlier\_values2 <- boxplot.stats(weather.tidy$Mean.VisibilityMiles)$out  
outlier\_values2

## [1] 4 2 3 1 4 3 1 2 2 1 3 4 4 2 4 3 -1 3 4 3 4

length(outlier\_values2)

## [1] 21

mean(weather.tidy$Mean.VisibilityMiles)

## [1] 8.847222

index1 <- which(weather.tidy$Mean.VisibilityMiles == -1)  
weather.tidy[index1,]$Mean.VisibilityMiles <- 8.847222  
index2 <- which(weather.tidy$Mean.VisibilityMiles == 1)  
weather.tidy[index2,]$Mean.VisibilityMiles <- 8.847222  
index3 <- which(weather.tidy$Mean.VisibilityMiles == 2)  
weather.tidy[index3,]$Mean.VisibilityMiles <- 8.847222  
index4 <- which(weather.tidy$Mean.VisibilityMiles == 3)  
weather.tidy[index4,]$Mean.VisibilityMiles <- 8.847222  
index5 <- which(weather.tidy$Mean.VisibilityMiles == 4)  
weather.tidy[index5,]$Mean.VisibilityMiles <- 8.847222

#### 1.5IQR rule에 의한 outlier 값은 -1, 1, 2, 3, 4 가 총 21개 있다. 일단 음수 값을 포함해 이 이상치들은 평균 시야거리로서 부적절하다는 판단하에 이들을 모두 평균으로 대체하였다. 가능하다면 이 데이터를 넘겨준 사람들에게 어떻게 처리하면 좋을지 물어보겠다.

### 10 Event변수를 보면 공백문자 “”가 포함되어있습니다. 비나 안개 같은 특별한 event가 없는 날이라는 표시인데, 더욱 명백하게 표현하는 것이 좋습니다. 공백문자를 “None”으로 바꾸시오.

weather.tidy$Events[weather.tidy$Events == ""] <- 'None'

#### 바꿈

### 11 data frame의 column name은 모두 소문자로 하는 것이 좋습니다. 나중에 대문자인지 소문자인지기억하지 않아도 되기 때문입니다. data frame에서 column name을 모두 소문자로 바꾸시오.

colnames(weather.tidy) <- tolower(colnames(weather.tidy))

#### data frame에서 column name을 모두 소문자로 바꿈