Data Science Homework 2

21600685

조예성

Data Exploration and Cleaning

Q1.

str()을 통해 관측치의 수와 변수의 수를 파악한다.

정답 : observation 1000개, variable 11개

Q2.

cust.df의 state.of.res, sex, marital.stat, housing.type 이상 이 4가지의 변수는 단순한 스트링의 의미가 아닌 범주형의 의미를 가져서 factor형 변수로 전환하는 것이 더 적절하다.

정답 :

cust.df$state.of.res <- as.factor(cust.df$state.of.res)

cust.df$sex <- as.factor(cust.df$sex)

cust.df$marital.stat <- as.factor(cust.df$marital.stat)

cust.df$housing.type <- as.factor(cust.df$housing.type)

다음 4개의 변수를 factor형으로 전환

Q3.

총 8자리의 스트링 ID를 구축할 때,

맨 앞쪽은 c로 되어 있고 그 나머지가 7자리의 숫자로 통일을 해야 하는데 앞쪽이 빈 경우 0 으로 채워야 하므로 str\_pad()를 활용해서 이를 해결한다.

정답 :

cust.df$custid <- as.character(cust.df$custid)

cust.df$custid <- str\_pad(cust.df$custid, width = 7, side = "left", pad = "0")

cust.df$custid <- str\_pad(cust.df$custid, width = 8, side = "left", pad = "c")

Q4.

colSums(is.na())를 통해 각 변수당 na의 개수를 파악한 뒤

length() 함수를 통해 변수들의 총 관측치의 수를 파악한다.

정답 : 모든 변수당 모든 관측치 수 – 1000개

Is.employed – 328개 – 328/1000 x 100 – 32.8%

Income – 328개 – 32.8%

housing.type 56개 – 56/1000 x 100 – 5.6%

recent.move 56개 – 56/1000 x 100 – 5.6%

num.vehicles = 56개 56/1000 x 100 – 5.6%

Q5.

NA값들이 있는 데이터셋을 cust.dfs라는 새로운 데이터 프레임으로 저장

이후 rowSums(is.na()) 테크닉을 활용해 개수의 동향 파악

이러한 NA값들을 갖는 고객의 수 파악.

정답 : 0, 0, 3, 0, 0, .. 식으로 NA값이 존재하는 경우는 다른 변수들 또한 NA값들이 존재했으며 NA가 존재하지 않는 경우는 다른 어떤 변수에도 NA값이 존재하지 않았다.

NA값을 포함하는 고객의 수는 총 56명이었다.

Q6.

우선 char형으로 is.employed변수를 다시 변환시켜준다.

그 다음 TRUE나 FALSE같은 string형 데이터는 str\_replace을 이용해 간단하게 바꾸어 주지만

NA는 string이 아니므로 str\_replace가 제한이 된다.

이 경우 직접 지정을 해서 변환을 해주는 방법을 사용한다.

cust.df$is.employed[which(is.na(cust.df$is.employed))] <- "missing"

Q7.

state.of.res 변수를 char형으로 전환해준 뒤

aggregate()함수를 써서 주별 소득의 평균값과 중앙값을 계산해준다.

그리고 avg\_income이라는 새로운 데이터프레임에 aggregate한 결과를 mean.income, median.income이라는 새로운 열을 추가하여 구축해준다.

코드 :

cust.df$state.of.res <- as.character(cust.df$state.of.res)

str(cust.df)

# 평균값

aggregate(Income~state.of.res, data= cust.df, mean)

# 중앙값

aggregate(Income~state.of.res, cust.df, median)

## df 구축

avg\_income <- aggregate(Income~state.of.res, data= cust.df, mean)

colnames(avg\_income)[2] <- "mean.income"

avg\_income$median.income <- aggregate(Income~state.of.res, data = cust.df, median)[, 'Income']

avg\_income

Q8.

cust\_income이라는 것은 주별 평균 소득을 의미하는 데이터프레임

이를 기존 cust\_df에 결합시켜주어 새로운 데이터프레임 cust\_df2를 만듦

이 때 병합 과정을 merge를 이용해서 해준다. 다만, key값을 state.of.res기준으로 결합을 해주고 all = T를 통해 모든 열항목에 대해 더해주어 주 이름이 중복되더라도 그 value들을 다 추가해준다.

고객별 소득(Income.x)과 주별 평균 소득(Income.y)이 나란히 데이터셋에 있는 점을 활용해 NA값들은 그 값을 추가해준다.

이후에 평균값을 저장하는 열을 삭제해준다.

정답 :

코드 :

cust\_income <- aggregate(Income~state.of.res, data= cust.df, mean)

cust\_df2 <- merge(cust.df, cust\_income, all = T, by = 'state.of.res')

cust\_df2

cust\_df2$Income.x <- ifelse(is.na(cust\_df2$Income.x), cust\_df2$Income.y, cust\_df2$Income.x)

cust\_df2$Income.x

colnames(cust\_df2)[5] <- "Income" # 다시 이름 변경

cust\_df2 <- cust\_df2[, -12]

# 정답 : 주별 평균값으로 소득 NA값 대체

head(cust\_df2)

Q9.

비슷한 방법으로

cust\_aggregate라는 데이터셋은 주별 소득 중앙값을 정리한 데이터프레임

이를 방금 전 cust\_df2에 추가하여 cust\_df3를 만든다.

동일하게 merge함수를 이용해 key = ‘state.of.res’로 하고 all = T를 통해 중복 결합을 수행한다.

이후 중간값 소득 대비 고객별 소득을 구한 새로운 열 income.relative를 추가해주어 연산을 마무리한다.

정답 :

코드 :

cust\_aggregate <- aggregate(Income~state.of.res, cust.df, median)

cust\_df3 <- merge(cust\_df2, cust\_aggregate, all = T, by = 'state.of.res')

head(cust\_df3)

cust\_df3$income.relative <- cust\_df3$Income.x/cust\_df3$Income.y

cust\_df3$income.relative

colnames(cust\_df3)[5] <- "Income"

cust\_df3 <- cust\_df3[, -12]

head(cust\_df3)

Q10.

summary(cust\_df3)

summary를 통해 살펴보면

age의 Max가 146.7

Income의 경우 Min이 -8700인 경우가 있었다.

평균 수명을 고려하여 146살의 경우와 소득이 -8700인 경우는 outlier로 간주 할 수 있다.

보다 정확히 boxplot을 그려보면 Q1-1.5IQR, Q3+1.5IQR에 각각 벗어나는 이상치임을 알 수 있다.

boxplot(cust\_df3$Income)

boxplot(cust\_df3$age)

Q11.

Q11.

우선 histogram을 이용하면 이를 정확히 분석할 수 있다.

hist(cust\_df3$age)

사람의 나이의 경우 145살의 경우 현재 평균 수명을 고려하면 생존하기는 어려운 수명이다.

cust\_df3$age

이 외에도 123살, 126살 등 예상으로는 23살 26살을 입력하는 과정에서 1을 앞에 붙여 발생한 오류로 추정해볼 수 있다.

개인 소득을 산출할 경우 보통 음수의 경우는 고려하지 않는다.

hist(cust\_df3$Income)

-8700도 8700소득을 입력하는 과정에서 발생한 오타로 추정해볼 수 있다.

2. Tidy Data

Q1.

행의 경우 보통 관측치를 의미하고 열의 경우 그 관측치에 대한 변수가 위치한다.

이 경우 행에는 corpID를 통해 기업의 ID를 표시하고 그 기업의 각종 리스크를 설명하고자 하는 데이터프레임이다.

하지만 변수가 되어야할 리스크의 종류들이 열에 위치하지 않고 행에 분포해 있으므로 tidy한 데이터 셋으로 보기 어렵다.

Q2.

관측치인 corpID를 기준으로 변수들의 모임인 열 index를 spread시켜줌으로 tidy하게 전환해준다.

정답 :

코드 :

bankruptcy\_df <- bankruptcy\_df[, -1]

bankruptcy\_df <-spread(bankruptcy\_df, index, rating)

bankruptcy\_df

save(bankruptcy\_df, file = "bankruptcy\_df.RData")

Q3.

str\_replace()을 이용한다. 다만 다소 비효율적으로 짠 단점이 있다.

정답 :

코드 :

bankruptcy\_df$Class <- str\_replace(bankruptcy\_df$Class, 'B', 'Bankruptcy')

bankruptcy\_df$Class <- str\_replace(bankruptcy\_df, 'NB', 'Non-Bankruptcy')

str(bankruptcy\_df)

bankruptcy\_df$Competitiveness

bankruptcy\_df$Competitiveness <- str\_replace(bankruptcy\_df$Competitiveness, 'A', 'Average')

bankruptcy\_df$Competitiveness <- str\_replace(bankruptcy\_df$Competitiveness, 'P', 'Positive')

bankruptcy\_df$Competitiveness <- str\_replace(bankruptcy\_df$Competitiveness, 'N', 'Negative')

bankruptcy\_df$Credibility

bankruptcy\_df$Credibility<- str\_replace(bankruptcy\_df$Credibility, 'A', 'Average')

bankruptcy\_df$Credibility <- str\_replace(bankruptcy\_df$Credibility, 'P', 'Positive')

bankruptcy\_df$Credibility <- str\_replace(bankruptcy\_df$Credibility, 'N', 'Negative')

bankruptcy\_df$`Financial Flexibility`

#bankruptcy\_df$Financial Flexibility

bankruptcy\_df$`Financial Flexibility`<- str\_replace(bankruptcy\_df$`Financial Flexibility`, 'A', 'Average')

bankruptcy\_df$`Financial Flexibility` <- str\_replace(bankruptcy\_df$`Financial Flexibility`, 'P', 'Positive')

bankruptcy\_df$`Financial Flexibility` <- str\_replace(bankruptcy\_df$`Financial Flexibility`, 'N', 'Negative')

bankruptcy\_df$`Industrial Risk`

bankruptcy\_df$`Industrial Risk`<- str\_replace(bankruptcy\_df$`Industrial Risk`, 'A', 'Average')

bankruptcy\_df$`Industrial Risk` <- str\_replace(bankruptcy\_df$`Industrial Risk`, 'P', 'Positive')

bankruptcy\_df$`Industrial Risk` <- str\_replace(bankruptcy\_df$`Industrial Risk`, 'N', 'Negative')

bankruptcy\_df$`Management Risk`

bankruptcy\_df$`Management Risk`<- str\_replace(bankruptcy\_df$`Management Risk`, 'A', 'Average')

bankruptcy\_df$`Management Risk` <- str\_replace(bankruptcy\_df$`Management Risk`, 'P', 'Positive')

bankruptcy\_df$`Management Risk` <- str\_replace(bankruptcy\_df$`Management Risk`, 'N', 'Negative')

bankruptcy\_df$`Operating Risk`

bankruptcy\_df$`Operating Risk`<- str\_replace(bankruptcy\_df$`Operating Risk`, 'A', 'Average')

bankruptcy\_df$`Operating Risk` <- str\_replace(bankruptcy\_df$`Operating Risk`, 'P', 'Positive')

bankruptcy\_df$`Operating Risk` <- str\_replace(bankruptcy\_df$`Operating Risk`, 'N', 'Negative')