

PR5 - Data Frame

김서준

17 10월, 2023

Dataframe

1. 벡터를 이용해 데이터프레임 만들기

```
getwd()
```

```
## [1] "C:/UnivStudy/UnivLectures/23-2/R-programming/Works/PR5"
```

```
name <- c("Boil", "Tom", "Ravindra", "Bob", "Sobia")
gender <- c("M", "M", "F", "M", "F")
age <- c(17, 21, 33, 12, 37)
marriage <- c(F, T, F, F, T)

customer <- data.frame(name, gender, age, marriage, stringsAsFactors = T)
str(customer)
```

```
## 'data.frame': 5 obs. of 4 variables:
## $ name : Factor w/ 5 levels "Bob","Boil","Ravindra",...: 2 5 3 1 4
## $ gender : Factor w/ 2 levels "F","M": 2 2 1 2 1
## $ age : num 17 21 33 12 37
## $ marriage: logi FALSE TRUE FALSE FALSE TRUE
```

```
customer <- data.frame(name, gender, age, marriage)
str(customer)
```

```
## 'data.frame': 5 obs. of 4 variables:
## $ name : chr "Boil" "Tom" "Ravindra" "Bob" ...
## $ gender : chr "M" "M" "F" "M" ...
## $ age : num 17 21 33 12 37
## $ marriage: logi FALSE TRUE FALSE FALSE TRUE
```

```
str(customer)
```

```
## 'data.frame': 5 obs. of 4 variables:
## $ name : chr "Boil" "Tom" "Ravindra" "Bob" ...
## $ gender : chr "M" "M" "F" "M" ...
## $ age : num 17 21 33 12 37
## $ marriage: logi FALSE TRUE FALSE FALSE TRUE
```

```
names(customer)
```

```
## [1] "name"      "gender"    "age"       "marriage"
```

```
rownames(customer)
```

```
## [1] "1" "2" "3" "4" "5"
```

2. Data Frame 변수명 바꾸기

```
colnames(customer)
```

```
## [1] "name"      "gender"    "age"       "marriage"
```

```
rownames(customer)
```

```
## [1] "1" "2" "3" "4" "5"
```

```
colnames(customer) <- c("cust_name", "cust_gend", "cust_age", "cust_mrg")
rownames(customer) <- c('a', 'b', 'c', 'd', 'e')
customer
```

```
##   cust_name cust_gend cust_age cust_mrg
## a      Boil         M      17   FALSE
## b       Tom         M      21    TRUE
## c  Ravindra         F      33   FALSE
## d       Bob         M      12   FALSE
## e      Sobia         F      37    TRUE
```

3. Data Frame 데이터 추출

```
customer[1,] ; customer['a', ] #첫번째 행 숫자 및 rowname 으로 추출
```

```
##   cust_name cust_gend cust_age cust_mrg
## a      Boil         M      17   FALSE
```

```
##   cust_name cust_gend cust_age cust_mrg
## a      Boil         M      17   FALSE
```

```
customer[customer$cust_name == "Tom",] #cust_name 컬럼이 Tom 인 row 만 추출
```

```
##   cust_name cust_gend cust_age cust_mrg
## b       Tom         M      21    TRUE
```

```
customer[2:5, ] ; customer[-1, ]
```

```
##   cust_name cust_gend cust_age cust_mrg
## b      Tom      M      21      TRUE
## c Ravindra      F      33     FALSE
## d      Bob      M      12     FALSE
## e     Sobia      F      37      TRUE
```

```
##   cust_name cust_gend cust_age cust_mrg
## b      Tom      M      21      TRUE
## c Ravindra      F      33     FALSE
## d      Bob      M      12     FALSE
## e     Sobia      F      37      TRUE
```

```
customer[customer$cust_name!="Tom",]
```

```
##   cust_name cust_gend cust_age cust_mrg
## a      Boil      M      17     FALSE
## c Ravindra      F      33     FALSE
## d      Bob      M      12     FALSE
## e     Sobia      F      37      TRUE
```

```
customer[c("b", "c"),]
```

```
##   cust_name cust_gend cust_age cust_mrg
## b      Tom      M      21      TRUE
## c Ravindra      F      33     FALSE
```

4. Data Frame 에 데이터추가

```
#이름으로 추가
customer$cust_height <- c("185", "165", "156", "174", "155")
customer["f", ] <- list("jack", "M", 50, T, "167")
customer
```

```
##   cust_name cust_gend cust_age cust_mrg cust_height
## a      Boil      M      17     FALSE         185
## b      Tom      M      21      TRUE         165
## c Ravindra      F      33     FALSE         156
## d      Bob      M      12     FALSE         174
## e     Sobia      F      37      TRUE         155
## f      jack      M      50      TRUE         167
```

```
# cbind, rbind 로 추가
customer <- cbind(customer, weight = c(80, 70, 65, 48, 55, 100))
customer <- rbind(customer, g=list("Merry", "F", 42, F, "172", 60))
customer <- rbind(customer, h = c("Merry", "F", 42, F, "172", 60))
customer
```

```
##   cust_name cust_gend cust_age cust_mrg cust_height weight
## a      Boil         M      17   FALSE        185      80
## b       Tom         M      21    TRUE        165      70
## c  Ravindra         F      33   FALSE        156      65
## d       Bob         M      12   FALSE        174      48
## e     Sobia         F      37    TRUE        155      55
## f      jack         M      50    TRUE        167     100
## g     Merry         F      42   FALSE        172      60
## h     Merry         F      42   FALSE        172      60
```

5. Data Frame 에 데이터 삭제

```
customer <- customer[, -5] #5번째 칼럼 빼고 다시
customer <- customer[-7, ] #7번째 로우를 없애고 다시 할당
customer$weight<-NULL
```

6.Data 조건문을 활용해 조작하기

```
customer[customer$cust_gend == "M",] #customer라는 이름의 데이터 프레임 안에 cust_gend열에서 데이터 값이 M인 행들만 추출
```

```
##   cust_name cust_gend cust_age cust_mrg
## a      Boil         M      17   FALSE
## b       Tom         M      21    TRUE
## d       Bob         M      12   FALSE
## f      jack         M      50    TRUE
```

```
customer[customer$cust_gend != "F",] #customer라는 이름의 데이터 프레임 안에 csut_gend열에서 데이터 값이 F가 아닌것들만 추출 (!= 는 ~~가 아니다 라는 뜻)
```

```
##   cust_name cust_gend cust_age cust_mrg
## a      Boil         M      17   FALSE
## b       Tom         M      21    TRUE
## d       Bob         M      12   FALSE
## f      jack         M      50    TRUE
```

```
nrow(customer[customer$cust_gend == "M",]) #customer라는 이름의 데이터 프레임 안에 cust_gend열에서 데이터 값이 M인 행들의 갯수
```

```
## [1] 4
```

```
customer[customer$cust_name == "Bob", c("cust_age", "cust_mrg")] #cust_name이 Bob인 행의 "cust_age"와 "cust_mrg"를 추출
```

```
##   cust_age cust_mrg
## d      12   FALSE
```

```
customer[customer$cust_name == "Tom" | customer$cust_name == "Ravindra", ] #cust_name에서 Tom
이거나 Ravindra인 데이터를 추출
```

```
##   cust_name cust_gend cust_age cust_mrg
## b      Tom      M      21     TRUE
## c Ravindra      F      33     FALSE
```

```
customer[customer$cust_gend=="M" & customer$cust_age>24, ] # cust_gend가 M이고 동시에 cust_age
가 24보다 큰 데이터들 추출
```

```
##   cust_name cust_gend cust_age cust_mrg
## f      jack      M      50     TRUE
```

7. Data frame 정렬하기

```
order(customer$cust_age)
```

```
## [1] 4 1 2 3 5 7 6
```

```
customer[order(customer$cust_age),]
```

```
##   cust_name cust_gend cust_age cust_mrg
## d      Bob      M      12     FALSE
## a      Boil      M      17     FALSE
## b      Tom      M      21     TRUE
## c Ravindra      F      33     FALSE
## e      Sobia      F      37     TRUE
## h      Merry      F      42     FALSE
## f      jack      M      50     TRUE
```

```
order(customer$cust_age, decreasing = F)
```

```
## [1] 4 1 2 3 5 7 6
```

```
customer[order(customer$cust_age, decreasing = F),]#내림차순 decreasing = T
```

```
##   cust_name cust_gend cust_age cust_mrg
## d      Bob      M      12     FALSE
## a      Boil      M      17     FALSE
## b      Tom      M      21     TRUE
## c Ravindra      F      33     FALSE
## e      Sobia      F      37     TRUE
## h      Merry      F      42     FALSE
## f      jack      M      50     TRUE
```

8. Data frame 기타 함수

```
head(customer) #상위 6개 row
```

```
##   cust_name cust_gend cust_age cust_mrg
## a      Boil         M      17   FALSE
## b       Tom         M      21    TRUE
## c  Ravindra         F      33   FALSE
## d       Bob         M      12   FALSE
## e     Sobia         F      37    TRUE
## f      jack         M      50    TRUE
```

```
head(customer, 2) #상위 2개 row
```

```
##   cust_name cust_gend cust_age cust_mrg
## a      Boil         M      17   FALSE
## b       Tom         M      21    TRUE
```

```
tail(customer, 2) #하위 2개 row
```

```
##   cust_name cust_gend cust_age cust_mrg
## f      jack         M      50    TRUE
## h     Merry         F      42   FALSE
```

#파일 입출력

1. 내장데이터 불러오기

#MASS 패키지에는 다양한 데이터가 들어있음

```
#install.packages("MASS")
library(MASS)
# iris 데이터셋
# 붓꽃의 종과 Sepal 과 Petal 의 너비와 길이에 대한 데이터
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4          0.2  setosa
## 2          4.9         3.0          1.4          0.2  setosa
## 3          4.7         3.2          1.3          0.2  setosa
## 4          4.6         3.1          1.5          0.2  setosa
## 5          5.0         3.6          1.4          0.2  setosa
## 6          5.4         3.9          1.7          0.4  setosa
```

```
str(iris)
```

```
## 'data.frame':   150 obs. of  5 variables:
## $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
## $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
## $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
## $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
## $ Species      : Factor w/ 3 levels "setosa","versicolor",...: 1 1 1 1 1 1 1 1 1 1 ...
```

```
#mtcars 데이터셋
# 자동차 차종별 상세스펙에 대한 데이터
head(mtcars)
```

```
##           mpg cyl disp  hp drat   wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710      22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive  21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant         18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
str(mtcars)
```

```
## 'data.frame':   32 obs. of  11 variables:
## $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
## $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
## $ disp: num  160 160 108 258 360 ...
## $ hp : num  110 110 93 110 175 105 245 62 95 123 ...
## $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
## $ wt : num  2.62 2.88 2.32 3.21 3.44 ...
## $ qsec: num  16.5 17 18.6 19.4 17 ...
## $ vs : num  0 0 1 1 0 1 0 1 1 1 ...
## $ am : num  1 1 1 0 0 0 0 0 0 0 ...
## $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
## $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
#USArrests 데이터셋
# 1973년도 50 개 주에서 수집된 범죄기록 데이터
head(USArrests)
```

```
##           Murder Assault UrbanPop Rape
## Alabama      13.2      236      58 21.2
## Alaska       10.0      263      48 44.5
## Arizona        8.1      294      80 31.0
## Arkansas        8.8      190      50 19.5
## California     9.0      276      91 40.6
## Colorado       7.9      204      78 38.7
```

```
str(USArrests)
```

```
## 'data.frame':   50 obs. of  4 variables:
## $ Murder   : num  13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4 ...
## $ Assault  : int  236 263 294 190 276 204 110 238 335 211 ...
## $ UrbanPop: int   58 48 80 50 91 78 77 72 80 60 ...
## $ Rape     : num  21.2 44.5 31 19.5 40.6 38.7 11.1 15.8 31.9 25.8 ...
```

2.file 로 저장된 데이터 불러오기

```
csv <- read.csv("read_csv.csv", fileEncoding = 'EUC-KR') ; csv
```

```
##   X..연습.테이블.입니다.      X      X.1      X.2
## 1                      1   Daredevil   Hawkeye   Loki
## 2                      2   Deadpool    Hulk      Luke Cage
## 3                      3 Doctor Strange Human Torch .
## 4                      6              Invisible Woman Ms. Marvel
## 5                      5              Iron Man Nightcrawler
## 6                      7   Ghost Rider   Jean Grey   Psylocke
##           X.3      X.4
## 1   Punisher   Storm
## 2 Rocket Raccoon Taskmaster
## 3 Scarlet Witch Thing
## 4 Silver Surfer   Thor
## 5           N.A.  Wolverine
## 6 Squirrel Girl  Barricade
```

```
str(csv)
```

```
## 'data.frame':   6 obs. of  6 variables:
## $ X..연습.테이블.입니다.: int  1 2 3 6 5 7
## $ X                      : chr  "Daredevil" "Deadpool" "Doctor Strange" "" ...
## $ X.1                    : chr  "Hawkeye" "Hulk" "Human Torch" "Invisible Woman" ...
## $ X.2                    : chr  "Loki" "Luke Cage" "." "Ms. Marvel" ...
## $ X.3                    : chr  "Punisher" "Rocket Raccoon" "Scarlet Witch" "Silver Surfer"
## ...
## $ X.4                    : chr  "Storm" "Taskmaster" "Thing" "Thor" ...
```

```
getwd()
```

```
## [1] "C:/UnivStudy/UnivLectures/23-2/R-programming/Works/PR5"
```

```
csv2 <- read.csv("read_csv.csv", header = F) ; csv2
```



```
##                                V1                V2
## 1 # WxbfWxacWxbdWxc0 Wxc5Wxd7Wxc0Wxed Wxc070Wxd9.
## 2                                1          Daredevil
## 3                                2          Deadpool
## 4                                3 Doctor Strange
## 5                                6
## 6                                5
## 7                                7          Ghost Rider
##                                V3                V4                V5                V6
## 1
## 2          Hawkeye          Loki          Punisher          Storm
## 3          Hulk          Luke Cage Rocket Raccoon Taskmaster
## 4          Human Torch          .          Scarlet Witch          Thing
## 5 Invisible Woman          Ms. Marvel Silver Surfer          Thor
## 6          Iron Man Nightcrawler          N.A. Wolverine
## 7          Jean Grey          Psylocke Squirrel Girl Barricade
```

```
str(csv2)
```

```
## 'data.frame':  7 obs. of  6 variables:
## $ V1: chr  "# WxbfWxacWxbdWxc0 Wxc5Wxd7Wxc0Wxed Wxc070Wxd9." "1" "2" "3" ...
## $ V2: chr  "" "Daredevil" "Deadpool" "Doctor Strange" ...
## $ V3: chr  "" "Hawkeye" "Hulk" "Human Torch" ...
## $ V4: chr  "" "Loki" "Luke Cage" "." ...
## $ V5: chr  "" "Punisher" "Rocket Raccoon" "Scarlet Witch" ...
## $ V6: chr  "" "Storm" "Taskmaster" "Thing" ...
```

```
csv3 <- read.csv("csv_NA.csv", header = F, na.strings = c(".", "N.A.", "")) ; csv3
```

```
##                                V1                V2                V3                V4
## 1 #연습 테이블 입니다.          <NA>          <NA>          <NA>
## 2                                1          Daredevil          Hawkeye          Loki
## 3                                2          Deadpool          Hulk          Luke Cage
## 4                                3 Doctor Strange          Human Torch          <NA>
## 5                                6          <NA> Invisible Woman          Ms. Marvel
## 6                                5          <NA>          Iron Man Nightcrawler
## 7                                7          Ghost Rider          Jean Grey          Psylocke
##                                V5                V6
## 1          <NA>          <NA>
## 2          Punisher          Storm
## 3 Rocket Raccoon Taskmaster
## 4 Scarlet Witch          Thing
## 5 Silver Surfer          Thor
## 6          <NA> Wolverine
## 7 Squirrel Girl Barricade
```

```
str(csv3)
```

```
## 'data.frame': 7 obs. of 6 variables:
## $ V1: chr "#연습 테이블 입니다." "1" "2" "3" ...
## $ V2: chr NA "Daredevil" "Deadpool" "Doctor Strange" ...
## $ V3: chr NA "Hawkeye" "Hulk" "Human Torch" ...
## $ V4: chr NA "Loki" "Luke Cage" NA ...
## $ V5: chr NA "Punisher" "Rocket Raccoon" "Scarlet Witch" ...
## $ V6: chr NA "Storm" "Taskmaster" "Thing" ...
```

```
csv4 <- read.csv("csv_NA.csv", header = F, stringsAsFactors = F, encoding = "UTF-8") ; csv4
```

```
##           V1           V2           V3           V4
## 1 #연습 테이블 입니다.
## 2           1      Daredevil      Hawkeye      Loki
## 3           2      Deadpool      Hulk      Luke Cage
## 4           3 Doctor Strange      Human Torch      .
## 5           6              Invisible Woman      Ms. Marvel
## 6           5              Iron Man Nightcrawler
## 7           7      Ghost Rider      Jean Grey      Psylocke
##           V5           V6
## 1
## 2      Punisher      Storm
## 3 Rocket Raccoon Taskmaster
## 4  Scarlet Witch      Thing
## 5 Silver Surfer      Thor
## 6           N.A.  Wolverine
## 7 Squirrel Girl  Barricade
```

```
str(csv4)
```

```
## 'data.frame': 7 obs. of 6 variables:
## $ V1: chr "#연습 테이블 입니다." "1" "2" "3" ...
## $ V2: chr "" "Daredevil" "Deadpool" "Doctor Strange" ...
## $ V3: chr "" "Hawkeye" "Hulk" "Human Torch" ...
## $ V4: chr "" "Loki" "Luke Cage" "." ...
## $ V5: chr "" "Punisher" "Rocket Raccoon" "Scarlet Witch" ...
## $ V6: chr "" "Storm" "Taskmaster" "Thing" ...
```

```
table <- read.table("read_csv.csv", header = F, sep = ",", stringsAsFactors = F)
head(table)
```

```
##  V1           V2           V3           V4           V5           V6
## 1  1      Daredevil      Hawkeye      Loki      Punisher      Storm
## 2  2      Deadpool      Hulk      Luke Cage Rocket Raccoon Taskmaster
## 3  3 Doctor Strange      Human Torch      .  Scarlet Witch      Thing
## 4  6              Invisible Woman      Ms. Marvel Silver Surfer      Thor
## 5  5              Iron Man Nightcrawler              N.A.  Wolverine
## 6  7      Ghost Rider      Jean Grey      Psylocke Squirrel Girl  Barricade
```

3. 웹에 있는 표를 읽어 오기 readHTMLTable()

```
library(XML)

url <- "http://www.worldometers.info/world-population/"

library(httr)

html_source <- GET(url)
tabs <- readHTMLTable(rawToChar(html_source$content), stringAsFactors = F)

world_pop <- tabs$popbycountry
head(world_pop)
```

```
##   # Country (or dependency) Population(2023) YearlyChange NetChange
## 1 1 India 1,428,627,663 0.81 % 11,454,490
## 2 2 China 1,425,671,352 -0.02 % -215,985
## 3 3 United States 339,996,563 0.5 % 1,706,706
## 4 4 Indonesia 277,534,122 0.74 % 2,032,783
## 5 5 Pakistan 240,485,658 1.98 % 4,660,796
## 6 6 Nigeria 223,804,632 2.41 % 5,263,420
## Density (P/Km² ) Land Area (Km² ) Migrants(net) Fert.Rate Med.Age UrbanPop %
## 1 481 2,973,190 -486,136 1.999 28 36.3 %
## 2 152 9,388,211 -310,220 1.19 39 65 %
## 3 37 9,147,420 999,700 1.662 38 82.9 %
## 4 153 1,811,570 -49,997 2.134 30 59.1 %
## 5 312 770,880 -165,988 3.347 21 34.7 %
## 6 246 910,770 -59,996 5.063 17 53.9 %
## WorldShare
## 1 17.8 %
## 2 17.7 %
## 3 4.2 %
## 4 3.4 %
## 5 3 %
## 6 2.8 %
```

4. 데이터 저장하기

```
table
```

```
##   V1      V2      V3      V4      V5      V6
## 1 1 Daredevil Hawkeye Loki Punisher Storm
## 2 2 Deadpool Hulk Luke Cage Rocket Raccoon Taskmaster
## 3 3 Doctor Strange Human Torch . Scarlet Witch Thing
## 4 6 Invisible Woman Ms. Marvel Silver Surfer Thor
## 5 5 Iron Man Nightcrawler N.A. Wolverine
## 6 7 Ghost Rider Jean Grey Psylocke Squirrel Girl Barricade
```

```
#write.table(table, "PR_table.csv")
#write.table(table, "PR_table1.csv", row.names = F)
#write.csv(table, "PR_table2.csv", row.names = F)
```

PR5 연습문제

데이터 출처: <https://www.bigdata-telecom.kr/invoke/SOKBP2603/?goodsCode=LTCFOOD>
<https://www.bigdata-telecom.kr/invoke/SOKBP2603/?goodsCode=LTCFOOD>

```
Sys.setlocale("LC_ALL", "C")
```

```
## [1] "C"
```

```
data <- read.csv('temp.csv', encoding = "UTF-8")
Sys.setlocale("LC_ALL", "Korean")
```

```
## Warning in Sys.setlocale("LC_ALL", "Korean"): using locale code page other than
## 65001 ("UTF-8") may cause problems
```

```
## [1] "LC_COLLATE=Korean_Korea.949;LC_CTYPE=Korean_Korea.949;LC_MONETARY=Korean_Korea.949;LC_N
UMERIC=C;LC_TIME=Korean_Korea.949"
```

```
data[which(is.na(data$agrd_code)), "agrd_code"] <- '결측'
table(data$agrd_code)
```

```
##
##      1      2      3      4      5      6      7
## 144 144 144 144 144 144 144
```

1. 데이터 내에서 연령 코드 변수의 데이터를 연령대로 변경하시오

```
data$agrd_code[data$agrd_code == 1] <- "20대 미만"
sum(data$agrd_code == "20대 미만")
```

```
## [1] 144
```

2. '한식' 업종만을 추출하여 korean_food 라는 변수에 할당하시오.

```
korean_food <- data[data$induty_nm == "한식", ]
head(korean_food)
```

```
##      X.U.FEFF.stdr_ym induty_nm sexdstn_code agrde_code settle_cascnt settle_amount
## 1      202101      한식      1  20대 미만      532      16571885
## 2      202101      한식      1      2      114328      3505999370
## 3      202101      한식      1      3      356410      10941412956
## 4      202101      한식      1      4      635087      20334597598
## 5      202101      한식      1      5      534371      17253687369
## 6      202101      한식      1      6      215463      6721233600
```

3. korean_food 데이터 프레임에서 결제건수가 많은 순서대로 정렬하고 상위 5개 데이터만 추출하시오

```
order(korean_food$settle_cascnt, decreasing = T)
```

```
##      [1] 137  67 138 151  81  60 130 152  68 109  53  74  82 144  95  39 110 102
##      [19]  46  32  54  88  96 165 166  40 158 131 145  61 103 123  75 116 159  89
##      [37] 124  47  33  11   4  25  18 117  12  26   5  19  59  73  45 129  31 143
##      [55] 101  66 136  87  80 157 150  52 115  38 108   3 139  94  17 153 122  83
##      [73] 164  69 111 132 146  55  97 104 167  62  76  41  24 160  90  10  48 125
##      [91]  34 118   6  13  20  27  65 149 135  79  58  37  51 142  30  72  44 128
##     [109] 107 163  93 156  86 100 121 114  23  16   2   9 133 147  77 140 105  63
##     [127] 154  84 161  91  70  49  35 112 119  98  56 168  42 126   7  21  28  14
##     [145] 141 127  29  43 148 162 155 134  78  64  57  92 106  36  71  99 113  50
##     [163]  85  15   1 120   8  22
```

```
head(korean_food[order(korean_food$settle_cascnt, decreasing = T),])
```

```
##      X.U.FEFF.stdr_ym induty_nm sexdstn_code agrde_code settle_cascnt
## 767      202110      한식      2      4      872619
## 347      202105      한식      2      4      859192
## 768      202110      한식      2      5      840573
## 851      202111      한식      2      4      836191
## 431      202106      한식      2      4      833957
## 340      202105      한식      1      4      831066
##      settle_amount
## 767  28675840970
## 347  28739569759
## 768  28351648149
## 851  28850127457
## 431  26576440246
## 340  30047900854
```

마크다운 오류방지 용

```
print("마크다운 오류방지")
```

```
## [1] "마크다운 오류방지"
```

4. 결제건수가 10000회 이상, 150000회 이하이며 20대가 주문한 데이터들의 날짜를 추출하시오.

```
data[data$settle_cascnt > 10000 & data$settle_cascnt < 150000 & data$agrde_code == 2, "X.U.FEFF.st
dr_ym" ]
```

```
## [1] 202101 202101 202101 202101 202101 202101 202101 202101 202101 202101 202102
## [11] 202102 202102 202102 202102 202102 202102 202102 202102 202102 202102 202103
## [21] 202103 202103 202103 202103 202103 202103 202103 202103 202103 202103 202104
## [31] 202104 202104 202104 202104 202104 202104 202104 202104 202104 202104 202105
## [41] 202105 202105 202105 202105 202105 202105 202105 202105 202106 202106 202106
## [51] 202106 202106 202106 202106 202106 202106 202107 202107 202107 202107 202107
## [61] 202107 202107 202107 202107 202107 202107 202108 202108 202108 202108 202108
## [71] 202108 202108 202108 202108 202108 202108 202109 202109 202109 202109 202109
## [81] 202109 202109 202109 202109 202109 202109 202110 202110 202110 202110 202110
## [91] 202110 202110 202110 202110 202110 202111 202111 202111 202111 202111 202111
## [101] 202111 202111 202111 202111 202112 202112 202112 202112 202112 202112 202112
## [111] 202112 202112 202112 202112
```

```
#devtools::install_github("JaseZiv/worldfootballR", ref = "main")
```

```
library(worldfootballR)
```

5. 아래의 코드를 실행시켰을 때 얻는 데이터는 지난 10월 1일 토트넘과 리버풀의 경기 내용이다. 골을 넣은 토트넘 선수들의 이름을 모두 출력하시오.

```
match_summary <- fb_match_summary(match_url = "https://fbref.com/en/matches/ec4145b4/Tottenham-
Hotspur-Liverpool-September-30-2023-Premier_League")
match_summary[match_summary$Home_Away == "Home" & match_summary$Event_Type == "Goal", "Event_Players"]
```

```
## [1] "Son Heung-min Assist: Richarlison"
```

6. 아래의 코드를 실행시켰을 때 얻는 데이터는 지난 10월 1일 토트넘과 리버풀의 경기에서 발생한 슈팅 정보를 담고 있다. 해당 경기에서 Son Heung-min 선수와 James Maddison 선수의 슈팅 데이터를 추출하시오.

```
shooting <- fb_match_shooting("https://fbref.com/en/matches/ec4145b4/Tottenham-Hotspur-Liverpool-September-30-2023-Premier_League")
shooting[(shooting$Player == "Son Heung-min" | shooting$Player == "James Maddison"),]
```

| ## | Date | Squad | Home_Away | Match_Half | Minute | Player |
|-------|------------|-----------|-----------|------------|--------|----------------|
| ## 3 | 2023-09-30 | Tottenham | Home | 1 | 30 | James Maddison |
| ## 5 | 2023-09-30 | Tottenham | Home | 1 | 36 | Son Heung-min |
| ## 7 | 2023-09-30 | Tottenham | Home | 1 | 45+1 | Son Heung-min |
| ## 10 | 2023-09-30 | Tottenham | Home | 2 | 48 | Son Heung-min |
| ## 12 | 2023-09-30 | Tottenham | Home | 2 | 49 | James Maddison |
| ## 13 | 2023-09-30 | Tottenham | Home | 2 | 51 | Son Heung-min |
| ## 17 | 2023-09-30 | Tottenham | Home | 2 | 73 | James Maddison |
| ## 22 | 2023-09-30 | Tottenham | Home | 2 | 89 | James Maddison |

| ## | Player_Href | xG | PSxG | Outcome | Distance | Body Part |
|-------|-------------------------------------|------|------|---------|----------|------------|
| ## 3 | /en/players/ee38d9c5/James-Maddison | 0.03 | 0.14 | Saved | 25 | Left Foot |
| ## 5 | /en/players/92e7e919/Son-Heung-min | 0.70 | 0.47 | Goal | 8 | Right Foot |
| ## 7 | /en/players/92e7e919/Son-Heung-min | 0.03 | 0.34 | Saved | 28 | Right Foot |
| ## 10 | /en/players/92e7e919/Son-Heung-min | 0.04 | | Blocked | 23 | Right Foot |
| ## 12 | /en/players/ee38d9c5/James-Maddison | 0.02 | 0.53 | Saved | 21 | Left Foot |
| ## 13 | /en/players/92e7e919/Son-Heung-min | 0.11 | 0.43 | Saved | 17 | Right Foot |
| ## 17 | /en/players/ee38d9c5/James-Maddison | 0.03 | | Blocked | 23 | Right Foot |
| ## 22 | /en/players/ee38d9c5/James-Maddison | 0.07 | | Blocked | 25 | Right Foot |

| ## | Notes | Player_SCA_1 | Event_SCA_1 | Player_SCA_2 | Event_SCA_2 |
|-------|--------|------------------|-------------|-----------------|-------------|
| ## 3 | | Pedro Porro | Pass (Live) | James Maddison | Pass (Live) |
| ## 5 | | Richarlison | Pass (Live) | James Maddison | Pass (Live) |
| ## 7 | | Dejan Kulusevski | Pass (Live) | | |
| ## 10 | | Richarlison | Pass (Live) | Richarlison | Take-On |
| ## 12 | | Cristian Romero | Pass (Live) | James Maddison | Pass (Live) |
| ## 13 | Volley | Destiny Udogie | Pass (Live) | Cristian Romero | Pass (Live) |
| ## 17 | | Manor Solomon | Pass (Live) | James Maddison | Pass (Live) |
| ## 22 | | Dejan Kulusevski | Pass (Live) | James Maddison | Pass (Live) |