PR5 - Data.Frame

강현승

2022 10 7

#Dataframe

1. 벡터를 이용해 데이터 프레임 만들기

data.frame 함수를 사용하여 데이터 프레임 만들기 stringsAsFactors 인수에 T값을 할당하면 각 열이 factor형으로 저장됨 R version 4 이후부터는 Default 값이 F이다.

```
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)

# stringsAsFactors = T 사용해서 만들기
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
```

```
# stringsAsFactors 인수 없이 만들기
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
```

```
# data.frame 함수와 관련된 다양한 함수 사용하기
str(customer) # 데이터 프레임의 구조를 확인
```

```
## '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) # 데이터 프레임의 열 이름을 확인
```

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

```
rownames(customer) # 데이터 프레임의 행 이름을 확인
```

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

2. DataFrame 변수명 바꾸기

```
# colnames, rownames 함수로 변수명 변환 및 확인


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. DataFrame 데이터 추출

```
# 접근 방식은 matrix와 동일
# [행,열] 연산자 및 $ 연산자 활용하여 데이터에 접근하기
customer[1, ]
```

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

```
customer["a", ] # 첫 번째 행 숫자 및 rowname으로 추출
```

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

```
customer[customer$cust_name == "Tom", ] # cust_name 컬럼이 Tom인 row만 추출
```

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

```
customer[2:5, ]
```

```
##   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[-1, ] # 2 ~ 5 행
```

```
##   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 컬럼이 Tom이 아닌 row
```

```
## [1] cust_name cust_gend cust_age  cust_mrg
## <0 rows> (or 0-length row.names)
```

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

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

4. DataFrame에 데이터 추가

```
# 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, rbind로 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("Meerry", "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    Meerry         F       42    FALSE         172     60
```

5. DataFrame에 데이터 삭제

```
customer = customer[, -5] # 1 번째 컬럼을 빼고 나머지만 다시 할당
customer = customer[-7,] # 7 번째 로우를 빼고 나머지만 다시 할당
customer$weight = NULL # weight 컬럼 삭제
```

6. Data 조건문을 활용해 조작하기

```
# 이 부분은 모든 코드에 주석 달 것!
# &와 | 연산자로 여러 개의 조건을 사용할 수 있음

customer[customer$cust_gend == "M", ]
```

```
##   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", ]
```

```
##   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", ]) # nrow는 행의 개수를 보여줌
```

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

```
customer[customer$cust_name == "Bob", c("cust_age", "cust_mrg")]
```

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

```
customer[customer$cust_name == "Tom" |
           customer$cust_name == "Ravindra", ]
```

```
##   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_name cust_gend cust_age cust_mrg
## f      Jack         M       50     TRUE
```

7. Dataframe 정렬하기

```
# order함수를 활용해 순서를 구하여, row조건에 넣어서 정렬
# decreasing = T 인수를 활용하여 오름차순, 내림차순 변경 가능
order(customer$cust_age) # order함수로 age에 대한 순서를 구함
```

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

```
customer[order(customer$cust_age), ] # row의 조건에 위에서 구한 순서를 넣음
```

```
##   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    Meerry         F       42    FALSE
## f      Jack         M       50     TRUE
```

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

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

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

```
##   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    Meerry         F       42    FALSE
## f      Jack         M       50     TRUE
```

8. Dataframe 기타 함수

```
# head, tail함수는 데이터 프레임이 상위, 하위 row를 출력함
# 기본 6 개를 출력하며, row 수를 지정할 수 있음
head(customer) # 상위 6 개 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 2 개 row
```

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

```
tail(customer, ) # 하위 2 개 row
```

```
##   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
## f      Jack         M       50     TRUE
## h    Meerry         F       42    FALSE
```

파일 입출력

1. 내장 데이터 불러오기

```
# MASS 패키지에는 다양한 데이터가 들어있음
# install.packages("MASS")
library(MASS)

# iris 데이터 셋
# 붓꽃의 종과 Sepal과 Petal의 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 데이터 셋
# 자동차 차종 별 상세 스펙에 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 데이터 셋
# 1973년도 50개 주에서 수집된 범죄 기록 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. file로 저장된 데이터 불러오기

```
# 블랙보드에서 실습과제에 첨부되어 있는 파일을 다운받아 사용할 것
# setwd함수로 해당 작업폴더 지정해주기 /setwd("c:/PR")
# 지정하지 않으면 내 문서가 기본 작업폴더
# read.csv() 함수 (첫 행 컬럼명으로 사용) (첫 열 로우명으로 사용) "" (입력된 데이터를 구분해주는 기호
)
# / header = T / row.names = 1 / sep = ,
# na.strings = c("Na", "nan") (NA 값으로 처리할 문자열 정의) / fileEncoding="UTF-8" (문자
열을 특정 형식으로 재인코딩) / encoding = "UTF-8" (불러들일 file의 인코딩을 미리 선언)

#그냥읽어오기
csv = read.csv("read_csv.csv", fileEncoding = 'EUC-KR')
csv
```

```
##   X1      Daredevil         Hawkeye          Loki       Punisher      Storm
## 1  2       Deadpool            Hulk     Luke Cage Rocket Raccoon Taskmaster
## 2  3 Doctor Strange     Human Torch             .  Scarlet Witch      Thing
## 3  6                Invisible Woman    Ms. Marvel  Silver Surfer       Thor
## 4  5                       Iron Man Nightcrawler           N.A.  Wolverine
## 5  7    Ghost Rider       Jean Grey      Psylocke  Squirrel Girl  Barricade
```

```
str(csv)
```

```
## 'data.frame':    5 obs. of  6 variables:
##  $ X1       : int  2 3 6 5 7
##  $ Daredevil: chr  "Deadpool" "Doctor Strange" "" "" ...
##  $ Hawkeye  : chr  "Hulk" "Human Torch" "Invisible Woman" "Iron Man" ...
##  $ Loki     : chr  "Luke Cage" "." "Ms. Marvel" "Nightcrawler" ...
##  $ Punisher : chr  "Rocket Raccoon" "Scarlet Witch" "Silver Surfer" "N.A." ...
##  $ Storm    : chr  "Taskmaster" "Thing" "Thor" "Wolverine" ...
```

```
# header, stringsAsFactors 사용
# 불러온 데이터가 어떻게 바뀌는지 확인해보세요
csv2 = read.csv("read_csv.csv", header = F)
csv2
```

```
##   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
```

```
str(csv2)
```

```
## 'data.frame':    6 obs. of  6 variables:
##  $ V1: int  1 2 3 6 5 7
##  $ V2: chr  "Daredevil" "Deadpool" "Doctor Strange" "" ...
##  $ V3: chr  "Hawkeye" "Hulk" "Human Torch" "Invisible Woman" ...
##  $ V4: chr  "Loki" "Luke Cage" "." "Ms. Marvel" ...
##  $ V5: chr  "Punisher" "Rocket Raccoon" "Scarlet Witch" "Silver Surfer" ...
##  $ V6: chr  "Storm" "Taskmaster" "Thing" "Thor" ...
```

```
# 결측 값 처리하기
# (".", "N.A.", "") 3 가지 문자를 모두 NA로 인식하도록 함
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" ...
```

```
# 인코딩 문제 해결하기
# 불러올 파일의 인코딩을 UTF-8로 지정
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" ...
```

```
# read.table() 함수
# table 형태로 저장된 2차원의 데이터를 불러옴
# txt파일이나 CSV파일을 불러올 수 있음
# 불러온 데이터는 데이터프레임으로 생성
# read.csv() 함수와 동일하게 인수를 사용
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 읽어오기 readHTMLTable()

```
# install.packages(c("XML","httr")) # 해당 패키지가 없다면 설치부터
library(XML)

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

library(httr)

html_source = GET(url) # html 전체 소스를 받아옴
tabs = readHTMLTable(rawToChar(html_source$content), stringsAsFactors =
                       F) # html의 콘텐츠 중에서 테이블만 추출

world_pop = tabs$popbycountry # 추출된 테이블들 중에서 원하는 테이블 선택 및 저장
head(world_pop)
```

```
##   # Country (or dependency) Population(2020) YearlyChange NetChange
## 1 1                Honduras        9,904,607       1.63 %   158,490
## 2 2    United Arab Emirates        9,890,402       1.23 %   119,873
## 3 3                Djibouti          988,000       1.48 %    14,440
## 4 4        Saint Barthelemy            9,877        0.3 %        30
## 5 5              Seychelles           98,347       0.62 %       608
## 6 6     Antigua and Barbuda           97,929       0.84 %       811
##   Density (P/Km²) Land Area (Km²) Migrants(net) Fert.Rate Med.Age UrbanPop %
## 1              89         111,890        -6,800    2.4872      24     57.3 %
## 2             118          83,600        40,000      1.42      33     86.4 %
## 3              43          23,180           900    2.7577      27       79 %
## 4             470              21                    N.A.    N.A.        0 %
## 5             214             460          -200      2.46      34     56.2 %
## 6             223             440             0         2      34     26.2 %
##   WorldShare
## 1      0.1 %
## 2      0.1 %
## 3        0 %
## 4        0 %
## 5        0 %
## 6        0 %
```

4. 데이터 저장하기

```
# write.table 또는 write.csv 함수 사용
# row.names = F는, 해당 인수를 T로 줄 경우 행 이름이 첫 열로 이동하여 저장되기 때문
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)
```

연습문제

업종 카드소비 트렌드 데이터 설정

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

```
## [1] "LC_CTYPE=C;LC_NUMERIC=C;LC_TIME=C;LC_COLLATE=C;LC_MONETARY=C;LC_MESSAGES=e
n_US.UTF-8;LC_PAPER=en_US.UTF-8;LC_NAME=C;LC_ADDRESS=C;LC_TELEPHONE=C;LC_MEASUREME
NT=en_US.UTF-8;LC_IDENTIFICATION=C"
```

```
data = read.csv("trend.csv", encoding = 'UTF-8')
data[which(is.na(data$agrde_code)), 'agrde_code']
```

```
## integer(0)
```

```
table(data$agrde_code)
```

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

연습 1

```
data$agrde_code[is.na(data$agrde_code)] = 0
data$agrde_code = factor(
  data$agrde_code,
  levels = c(0, 1, 2, 3, 4, 5, 6, 7),
  labels = c(
    '-',
    '20대 미만',
    "20세~29세",
    '30세~39세',
    '40세~49세',
    '50세~59세',
    '60세~69세',
    '70세 이상'
  ),

)
data$agrde_code[is.na(data$agrde_code)] = '-'
```

연습 2

```
korean_food = factor(levels(factor(data$induty_nm)))
# \uD55C\uC2DD 한식
korean_food = data[korean_food == levels(korean_food)[(levels(korean_food) == '\uD
55C\uC2DD')],]
```

연습 3

```
head(korean_food[order(korean_food$setle_cascnt, decreasing = T), ], 5)
```

```
##     X.U.FEFF.stdr_ym        induty_nm sexdstn_code            agrde_code
## 768           202110 <U+D55C><U+C2DD>            2 50<U+C138>~59<U+C138>
## 852           202111 <U+D55C><U+C2DD>            2 50<U+C138>~59<U+C138>
## 348           202105 <U+D55C><U+C2DD>            2 50<U+C138>~59<U+C138>
## 432           202106 <U+D55C><U+C2DD>            2 50<U+C138>~59<U+C138>
## 600           202108 <U+D55C><U+C2DD>            2 50<U+C138>~59<U+C138>
##     setle_cascnt setle_amount
## 768       840573  28351648149
## 852       814744  28523376792
## 348       804837  26641576662
## 432       797904  25830722700
## 600       780094  24673982277
```

연습 4

```
korean_food[korean_food$agrde_code == levels(data$agrde_code)[3] &
              korean_food$setle_cascnt >= 10000 &
              korean_food$setle_cascnt <= 150000, 'X.U.FEFF.stdr_ym']
```

```
##  [1] 202101 202102 202103 202104 202105 202106 202107 202108 202109 202110
## [11] 202111 202112
```

연습 5

```
# install.packages('devtools')
# library(devtools)
# devtools::install_github('JaseZiv/worldfootballR', ref = 'main')
library(worldfootballR)

match_summary = fb_match_summary(match_url = "https://fbref.com/en/matches/74aed880/Ajax-Napoli-October-4-2022-Champions-League")
match_summary[match_summary$Home_Away == 'Away' &
                match_summary$Event_Type == 'Goal', 'Event_Players']
```

```
## [1] "Giacomo Raspadori Assist: Math<U+00ED>as Olivera"
## [2] "Giovanni Di Lorenzo Assist: Khvicha Kvaratskhelia"
## [3] "Piotr Zieli<U+0144>ski Assist: Andre-Frank Zambo Anguissa"
## [4] "Giacomo Raspadori Assist: Andre-Frank Zambo Anguissa"
## [5] "Khvicha Kvaratskhelia Assist: Giacomo Raspadori"
## [6] "Giovanni Simeone Assist: Tanguy Ndombele"
```

연습 6

```
shooting = fb_match_shooting(
  "https://fbref.com/en/matches/2f44d120/Eintracht-Frankfurt-Totterham-Hotspur-October-4-2022-Champions-League"
)
shooting[shooting$Shooting_Player == 'Son Heung-min' |
           shooting$Shooting_Player == 'Harry Kane', ]
```

```
##          Date     Squad Home_Away Match_Half Minute Shooting_Player    Outcome
## 15 2022-10-04 Tottenham      Away          1     25      Harry Kane    Wayward
## 16 2022-10-04 Tottenham      Away          1     28      Harry Kane Off Target
## 17 2022-10-04 Tottenham      Away          1     40   Son Heung-min Off Target
## 20 2022-10-04 Tottenham      Away          2     51      Harry Kane    Blocked
## 21 2022-10-04 Tottenham      Away          2     54   Son Heung-min Off Target
## 22 2022-10-04 Tottenham      Away          2     81      Harry Kane      Saved
## 23 2022-10-04 Tottenham      Away          2     83   Son Heung-min Off Target
##    Distance  Body_Part             Shot_Notes               SCA1_Player
## 15        6      Other              Open goal              Son Heung-min
## 16       22 Right Foot                                     Son Heung-min
## 17       19 Right Foot                                        Harry Kane
## 20        5 Right Foot                          Ivan Peri<U+0161>i<U+0107>
## 21       17 Right Foot                                      Richarlison
## 22       31 Right Foot Deflected, Half volley
## 23        4 Right Foot                                   Ryan Sessegnon
##     SCA1_Event            SCA2_Player  SCA2_Event
## 15 Pass (Live)            Richarlison Pass (Live)
## 16 Pass (Live) Pierre H<U+00F8>jbjerg Pass (Live)
## 17 Pass (Live) Pierre H<U+00F8>jbjerg Pass (Live)
## 20 Pass (Live)          Son Heung-min Pass (Live)
## 21 Pass (Live)
## 22
## 23 Pass (Live)             Harry Kane Pass (Live)
```

연습 7

```
man_city_url = "https://fbref.com/en/squads/b8fd03ef/Manchester-City-Stats"
man_city_logs = fb_team_match_log_stats(team_urls = man_city_url, stat_type =
                                        "passing")
man_city_logs[man_city_logs$Result == 'W' &
                man_city_logs$PPA > 10, ]
```

```
##                                                    Team_Url            Team
## 3  https://fbref.com/en/squads/b8fd03ef/Manchester-City-Stats Manchester City
## 5  https://fbref.com/en/squads/b8fd03ef/Manchester-City-Stats Manchester City
## 6  https://fbref.com/en/squads/b8fd03ef/Manchester-City-Stats Manchester City
## 8  https://fbref.com/en/squads/b8fd03ef/Manchester-City-Stats Manchester City
## 9  https://fbref.com/en/squads/b8fd03ef/Manchester-City-Stats Manchester City
## 11 https://fbref.com/en/squads/b8fd03ef/Manchester-City-Stats Manchester City
## 12 https://fbref.com/en/squads/b8fd03ef/Manchester-City-Stats Manchester City
## NA                                                      <NA>            <NA>
##    ForAgainst       Date  Time           Comp       Round  Day Venue Result
## 3         For 2022-08-13 15:00 Premier League Matchweek 2  Sat  Home      W
## 5         For 2022-08-27 15:00 Premier League Matchweek 4  Sat  Home      W
## 6         For 2022-08-31 19:30 Premier League Matchweek 5  Wed  Home      W
## 8         For 2022-09-06 21:00   Champions Lg Group stage  Tue  Away      W
## 9         For 2022-09-14 20:00   Champions Lg Group stage  Wed  Home      W
## 11        For 2022-10-02 14:00 Premier League Matchweek 9  Sun  Home      W
## 12        For 2022-10-05 20:00   Champions Lg Group stage  Wed  Home      W
## NA       <NA>       <NA>  <NA>           <NA>        <NA> <NA>  <NA>   <NA>
```

```
##      GF   GA        Opponent Cmp_Total Att_Total Cmp_percent_Total
## 3     4    0     Bournemouth       667       730               91.4
## 5     4    2  Crystal Palace       734       819               89.6
## 6     6    0 Nott'ham Forest       740       820               90.2
## 8     4    0          Sevilla       569       643               88.5
## 9     2    1        Dortmund       662       755               87.7
## 11    6    3  Manchester Utd       494       562               87.9
## 12    5    0   FC Copenhagen       824       896               92.0
## NA <NA> <NA>            <NA>        NA        NA                 NA
##    TotDist_Total PrgDist_Total Cmp_Short Att_Short Cmp_percent_Short Cmp_Medium
## 3          12941          3109       254       266               95.5         307
## 5          14444          3494       282       302               93.4         341
## 6          14699          3627       273       289               94.5         356
## 8          10634          2971       243       258               94.2         254
## 9          13377          3230       246       269               91.4         298
## 11         10403          2759       155       173               89.6         244
## 12         15178          3500       350       371               94.3         344
## NA            NA            NA        NA        NA                 NA          NA
##    Att_Medium Cmp_percent_Medium Cmp_Long Att_Long Cmp_percent_Long Ast  xA KP
## 3         330                93.0       92      114              80.7   3 1.2 17
## 5         376                90.7      103      127              81.1   4 1.8 15
## 6         381                93.4      106      136              77.9   3 1.9 12
## 8         272                93.4       64       89              71.9   4 3.5 20
## 9         327                91.1      111      142              78.2   2 0.8 12
## 11        259                94.2       86      108              79.6   6 2.2 15
## 12        366                94.0      108      129              83.7   3 2.0 22
## NA         NA                  NA       NA       NA                NA  NA  NA NA
##    Final_Third PPA CrsPA Prog
## 3           80  15     3   38
## 5           77  22     9   57
## 6           57  13     1   47
## 8           45  11     3   43
## 9           72  11     4   52
## 11          17  15     2   36
## 12          88  21     3   53
## NA          NA  NA    NA   NA
```

도전문제

아래 문제는 “업종 카드소비 트렌드.csv”를 활용합니다.

월 별 판매액 총계를 구하고 당월 판매액이 큰 순으로 기준년월을 5 개 나타내시오. 월 별 판매액 총계의 평균, 분산, 표준편차를 구하시오.

```
data.levels = levels(factor(data$X.U.FEFF.stdr_ym)) # 년월로 팩터를 만든 다음, 레벨을 불러온다
data.sum_by_month_list = integer(length(data.levels)) # 월 별 합계를 저장하기 위한 integer 벡터를 data.levels길이로 만큼 만듦.
names(data.sum_by_month_list) = data.levels # levels로 data.sum_by... 의 이름을 지정
for (levelName in levels(factor(data$X.U.FEFF.stdr_ym))) {
  # level 이름을 for 문으로 반복
  # data.sum_by...의 하나의 요소에 이전에 지정했던 이름인 level이름으로 접근한다.
  data.sum_by_month_list[data.levels == levelName] = sum(data$setle_amount[data$X.U.FEFF.stdr_ym == levelName])
  # data$setle_amount 열을 년월로 추출하여 더한다.
}
head(names(data.sum_by_month_list)[order(data.sum_by_month_list)], 5) # data.sum_by...로 정렬하여 상위 5개 데이터를 출력한다.
```

```
## [1] "202101" "202102" "202109" "202107" "202104"
```

```
mean(data.sum_by_month_list) # 평균
```

```
## [1] 213502449451
```

```
var(data.sum_by_month_list) # 분산
```

```
## [1] 7.609594e+20
```

```
sd(data.sum_by_month_list) # 표준편차
```

```
## [1] 27585492156
```