R 기초문법

정민지

2018-10-01

Contents

0.1 데이타 불러오기

0.1.1 base 데이타 셋 불러오기

```r
library(datasets)
data("mtcars")
mtcars
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

0.2 기본연산

```
# An addition
5 + 5
```

```
## [1] 10
```

```
# A subtraction
5 - 5
```

```
## [1] 0
```

```
# A multiplication
3 * 5
```

```
## [1] 15
```

```
# A division
(5 + 5) / 2
```

```
## [1] 5
```

```
# Exponentiation
2^3
```

```
## [1] 8
```

```
# Modulo
15%%3
```

```
## [1] 0
```

0.3 변수할당

```
t <- 42
t
```

```
## [1] 42
```

0.4 데이타포맷

0.4.1 데이타 타입 확인

```
my_numeric <- 42
my_character <- "universe"
my_logical <- FALSE

# Check class of my_numeric
class(my_numeric)
```

```
## [1] "numeric"
```

```
# Check class of my_character
class(my_character)
```

```
## [1] "character"
```

```
# Check class of my_logical
class(my_logical)
```

```
## [1] "logical"
```

0.4.2 벡터

```
numeric_vector <- c(1, 10, 49)
numeric_vector
```

```
## [1]  1 10 49
```

```
character_vector <- c("a", "b", "c", "d", "e")
character_vector
```

```
## [1] "a" "b" "c" "d" "e"
```

```
boolean_vector <-c(TRUE, FALSE,TRUE)
boolean_vector
```

```
## [1]  TRUE FALSE  TRUE
```

```
# vector
vec <- character_vector[3]
vec
```

```
## [1] "c"
```

```
mid <- character_vector[c(2:4)]
mid
```

```
## [1] "b" "c" "d"
```

0.4.3 팩터(Factor)

벡터를 출력한 이후 level도 출력한다. *레벨(level): factor변수의 unique 값 - Group by와 동일한 기능

```
# options
## order : levels
## levels :
temperature_vector <- c("High", "Low", "High","Low", "Medium")
factor_temperature_vector <- factor(temperature_vector,
                                    order = TRUE,
                                    levels = c("Low", "Medium", "High"))
factor_temperature_vector
```

```
## [1] High   Low    High   Low    Medium
## Levels: Low < Medium < High
```

```
# force to change levels which is not fit in
## have to be same size with origin levels,
levels(factor_temperature_vector) <- c("J","K","M")
factor_temperature_vector
```

```
## [1] M J M J K
## Levels: J < K < M
```

0.4.4

0.4.5

0.4.5.1 벡터 컬럼명 부여

```
# Poker winnings from Monday to Friday
poker_vector <- c(140, -50, 20, -120, 240)
# Roulette winnings from Monday to Friday
roulette_vector <- c(-24, -50, 100, -350, 10)
# The variable days_vector
days_vector <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")

# Assign the names of the day to roulette_vector and poker_vector
names(poker_vector) <-   days_vector
poker_vector
```

```
##    Monday   Tuesday Wednesday  Thursday    Friday
##       140       -50        20      -120       240
```

```
names(roulette_vector) <- days_vector
roulette_vector
```

```
##    Monday   Tuesday Wednesday  Thursday    Friday
##       -24       -50       100      -350        10
```

```
# vector using column names
poker_vector[c("Monday","Thursday")]
```

```
##   Monday Thursday
##      140     -120
```

0.4.6

0.4.7

0.4.7.1 벡터 연산

```
A_vector <- c(1, 2, 3)
B_vector <- c(4, 5, 6)

# Take the sum of A_vector and B_vector
total_vector <- A_vector + B_vector

# Print out total_vector
total_vector
```

```
## [1] 5 7 9
```

```
# vector : single number
t_vector <- B_vector[B_vector > 4]
t_vector
```

```
## [1] 5 6
```

0.4.8 행렬(Matrix)

0.4.8.1 행렬

```
# declear matrix
## byrow : TRUE - record order
num.m <- matrix(1:9,byrow =TRUE, nrow = 3)
## byrow : FALSE - tupple
matrix(1:9,byrow =FALSE, nrow = 3)
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    7
## [2,]    2    5    8
## [3,]    3    6    9
```

```
# set names
colnames(num.m) <- c("1st","2nd","3rd")
rownames(num.m) <- c("4th","5th","6th")

num.m
```

```
##     1st 2nd 3rd
## 4th   1   2   3
## 5th   4   5   6
## 6th   7   8   9
```

```
# 4th Row
num.m[1,]
```

```
## 1st 2nd 3rd
##   1   2   3
```

```
# 2nd Col
num.m[,2]
```

```
## 4th 5th 6th
##   2   5   8
```

```
# 1st & 5th
num.m[2,1]
```

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

0.4.9

0.4.9.1 행렬

0.4.9.1.1 bind 함수

행렬 컬럼 병합

```
box_office <- c(460.998, 314.4, 290.475, 247.900, 309.306, 165.8)
star_wars_matrix <- matrix(box_office, nrow = 3, byrow = TRUE,
                           dimnames = list(c("A New Hope", "The Empire Strikes Back", "Return of the Je
                                             c("US", "non-US")))

# The worldwide box office figures
worldwide_vector <- rowSums(star_wars_matrix)

# cbind - column bind
all_wars_matrix <- cbind(star_wars_matrix,worldwide_vector)
all_wars_matrix
```

```
##                              US non-US worldwide_vector
## A New Hope              460.998  314.4          775.398
## The Empire Strikes Back 290.475  247.9          538.375
## Return of the Jedi      309.306  165.8          475.106
```

```
# rbind - row bind
all_wars_matrix2 <- rbind(star_wars_matrix,worldwide_vector)
```

```
## Warning in rbind(star_wars_matrix, worldwide_vector): number of columns of
## result is not a multiple of vector length (arg 2)
```

```
all_wars_matrix2
```

```
##                              US  non-US
## A New Hope              460.998 314.400
## The Empire Strikes Back 290.475 247.900
## Return of the Jedi      309.306 165.800
## worldwide_vector        775.398 538.375
```

0.4.10 데이타 프레임(Data Frame)

0.4.10.1 기본구조

```
library(datasets)
data("mtcars")
#
mtcars
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
#
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 ...
```