R basics

SIT22009: Data Science

Presented by Hyebong Choi

Useful functions

sample(X, #sample, replace = FALSE, ...)

- sample
  - random-sampling with and without(default) replacement

```
set.seed(2018)

x <- 1:20
sample(x, 10)
##  [1]  7  9  2  4  8  5 19 18 12 20
sample(x, 10, replace = TRUE)
##  [1]  8 14 20 14 17 13  6 12 15 17
sample(x, 10, replace = FALSE)
##  [1]  6 11  3  2 13  9  5 16  8  1
```

Random Shuffling with **sample()**

```
# random shuffling
x <- 1:10
sample(x, length(x))
```

```
##  [1]  2  8  1  4  7  6  9  5 10  3
```

```
women_shuffle <- women[sample(1:nrow(women), nrow(women)), ]
```

```
head(women)
```

```
##   height weight
## 1     58    115
## 2     59    117
## 3     60    120
## 4     61    123
## 5     62    126
## 6     63    129
```

```
head(women_shuffle)
```

```
##    height weight
## 14     71    159
## 11     68    146
## 8      65    135
## 13     70    154
## 7      64    132
## 1      58    115
```

Split

split(df, split_var, ...)

- Split a data frame into a list of data frames with split variable

```
split(mtcars, mtcars$cyl)
```

```
## $`4`
##                 mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## 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
...
##
## $`6`
##                 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
...
##
## $`8`
##                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
...
```

Split

split(df, split_var, ...)

- Split a data frame into a list of data frames with split variable

```
split(mtcars, mtcars$mpg > 20)
```

```
## $`FALSE`
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 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
...
##
## $`TRUE`
##                  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
...
```

Subset

subset(df, condition, ...)

- Find a subset of dataframe with a criteria

```
subset(mtcars, mpg > 25)
```

```
##                 mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 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
## 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
```

equivalent to

```
mtcars[mtcars$mpg > 25, ]
```

```
##                 mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## 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
## 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
```

Merge

Merge(df1, df2, ...)

- join two data frames into one with common variables

```
x <- data.frame( name = c("John", "Bob", "Carol"),
                 math = c(70,80,90))
y <- data.frame( name = c("John", "Bob", "Alice"),
                 history = c(100,55,75))
x
```

```
##    name math
## 1  John   70
## 2   Bob   80
## 3 Carol   90
```

```
y
```

```
##    name history
## 1  John     100
## 2   Bob      55
## 3 Alice      75
```

```
merge(x,y)
```

```
##   name math history
## 1  Bob   80      55
## 2 John   70     100
```

```
merge(x,y,all = T)
```

```
##    name math history
## 1   Bob   80      55
## 2 Carol   90      NA
## 3  John   70     100
## 4 Alice   NA      75
```

which

- Find positions of elements that satisfy the condition

```
x <- c(5,1,2,6,3,17,8,9, 12)
myindex <- which( x > 10)
myindex
```

```
## [1] 6 9
```

```
x[myindex]
```

```
## [1] 17 12
```

which.max which.min

- Find positions of maximum and minimum elements

```
x
## [1]  5  1  2  6  3 17  8  9 12
which.max(x)
## [1] 6
which.min(x)
## [1] 2
x[which.max(x)]
## [1] 17
x[which.min(x)]
## [1] 1
```

cut

- makes a range-group(factor) variable

```
mtcars$wt
```

```
##  [1] 2.620 2.875 2.320 3.215 3.440 3.460 3.570 3.190 3.150 3.440 3.440
## [12] 4.070 3.730 3.780 5.250 5.424 5.345 2.200 1.615 1.835 2.465 3.520
## [23] 3.435 3.840 3.845 1.935 2.140 1.513 3.170 2.770 3.570 2.780
```

```
mtcars$wt_grp <- cut(mtcars$wt, breaks = c(0,2,4,6))
```

```
mtcars[, c('wt','wt_grp')]
```

```
##                     wt wt_grp
## Mazda RX4          2.620  (2,4]
## Mazda RX4 Wag      2.875  (2,4]
## Datsun 710         2.320  (2,4]
## Hornet 4 Drive     3.215  (2,4]
## Hornet Sportabout  3.440  (2,4]
## Valiant            3.460  (2,4]
## Duster 360         3.570  (2,4]
## Merc 240D          3.190  (2,4]
## Merc 230           3.150  (2,4]
## Merc 280           3.440  (2,4]
## Merc 280C          3.440  (2,4]
## Merc 450SE         4.070  (4,6]
```

quantile

- to find out percentiles

```
quantile(iris$Sepal.Length)
```

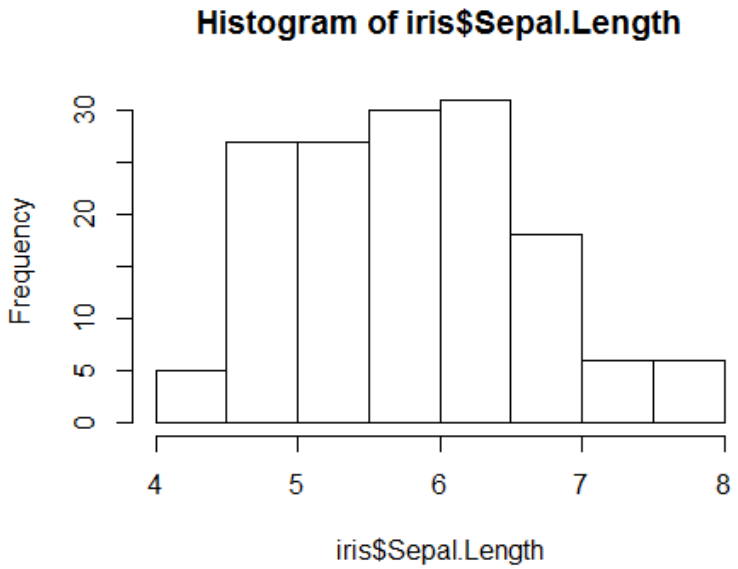
```
  Smallest     Median     Biggest
```

```
##   0%  25%  50%  75% 100%
##  4.3  5.1  5.8  6.4  7.9
```

```
quantile(iris$Sepal.Length, probs = c(0.1,0.5,0.9))
```

```
## 10% 50% 90%
## 4.8 5.8 6.9
```

```
hist(iris$Sepal.Length)
```



Combination of quantile and cut

```
cut_points <- quantile(mtcars$mpg,
c(0,0.25,0.75,1))
mtcars$fuel_efficiency <-
  cut(mtcars$mpg, breaks = cut_points,
include.lowest = T)
head(mtcars[, c('mpg', 'fuel_efficiency')],
10)
```

```
##                   mpg fuel_efficiency
## Mazda RX4         21.0     (15.4,22.8]
## Mazda RX4 Wag     21.0     (15.4,22.8]
## Datsun 710        22.8     (15.4,22.8]
## Hornet 4 Drive    21.4     (15.4,22.8]
## Hornet Sportabout 18.7     (15.4,22.8]
## Valiant           18.1     (15.4,22.8]
## Duster 360        14.3     [10.4,15.4]
## Merc 240D         24.4     (22.8,33.9]
## Merc 230          22.8     (15.4,22.8]
## Merc 280          19.2     (15.4,22.8]
```

```
levels(mtcars$fuel_efficiency) <-
c('low25perc', 'normal', 'high25perc')
head(mtcars[, c('mpg', 'fuel_efficiency')], 10)
```

```
##                   mpg fuel_efficiency
## Mazda RX4         21.0          normal
## Mazda RX4 Wag     21.0          normal
## Datsun 710        22.8          normal
## Hornet 4 Drive    21.4          normal
## Hornet Sportabout 18.7          normal
## Valiant           18.1          normal
## Duster 360        14.3       low25perc
## Merc 240D         24.4      high25perc
## Merc 230          22.8          normal
## Merc 280          19.2          normal
```

quantile

- frequency table

```
table(mtcars$fuel_efficiency)
```

```
##
##  low25perc     normal high25perc
##          8         17          7
```

```
table(mtcars$cyl)
```

```
##
##  4  6  8
## 11  7 14
```

```
table(mtcars$fuel_efficiency,
mtcars$cyl)
```

```
##
##               4 6 8
##   low25perc   0 0 8
##   normal      4 7 6
##   high25perc  7 0 0
```

paste and paste0

- to concatenate several values into one string
- to concatenate element by element from 2 or more vectors
- to smash vector elements into one string

```
paste("one", 1, "test")
```

```
## [1] "one 1 test"
```

```
x <- seq(2, 20, 2)
y <- LETTERS[1:10]
```

```
paste(x,y)
```

```
##  [1] "2 A"  "4 B"  "6 C"  "8 D"  "10 E" "12 F" "14 G" "16 H" "18 I" "20 J"
```

```
paste(x,y, sep =  ':')
```

```
##  [1] "2:A"  "4:B"  "6:C"  "8:D"  "10:E" "12:F" "14:G"
##  [8] "16:H" "18:I" "20:J"
```

https://www.r-bloggers.com/paste-paste0-and-sprintf/

paste and paste0

- need to use ‘sep’ and ‘collapse’ option properly
- useful to generate column names and row names
- paste0 equals to paste(..., sep = ‘’)

```
paste('var', x)
```

```
##  [1] "var 2"  "var 4"  "var 6"  "var 8"  "var 10" "var 12" "var 14"
##  [8] "var 16" "var 18" "var 20"
```

```
paste0('var', x)
```

```
##  [1] "var2"  "var4"  "var6"  "var8"  "var10" "var12" "var14" "var16"
##  [9] "var18" "var20"
```

```
paste('var', x, y, sep = '-')
```

```
##  [1] "var-2-A"  "var-4-B"  "var-6-C"  "var-8-D"  "var-10-E" "var-12-F"
##  [7] "var-14-G" "var-16-H" "var-18-I" "var-20-J"
```

```
paste(x)
```

```
##  [1] "2"  "4"  "6"  "8"  "10" "12" "14" "16" "18" "20"
```

```
paste(x, collapse = ',')
```

```
## [1] "2,4,6,8,10,12,14,16,18,20"
```

```
paste(paste0(x,y), collapse = ',')
```

```
## [1] "2A,4B,6C,8D,10E,12F,14G,16H,18I,20J"
```

https://www.r-bloggers.com/paste-paste0-and-sprintf/

References

- Practical Data Science with R, by Nina Zumel and John Mount