

R packages

# What is a package?

- A collection of useful functions and data for specific tasks
- Add-on packages can be downloaded and installed from the CRAN repository

# Obtaining packages

- Internet access is required
- Installation
  - > `install.packages("packagename")`
- Updating
  - > `update.packages()`
  - > `update.packages("packagename")`
- Uninstallation
  - > `remove.packages("packagename")`
- Check installed packages
  - > `(.packages())` #current packages
  - > `getOption("defaultPackages")` # default packages
  - > `(.packages(all.available=TRUE))` # all packages available
  - > `library()` # all packages and summary of each

# Using packages

- Once a package is installed it should be loaded  
`>library("packagename")` or `library(packagename)`
- To load it automatically when you start R, put the function in the file `".Rprofile"` in your home directory where R is called  
`> Sys.getenv() # to see R directory structure`
- Check what functions and data frames are available from that package  
`> ls("package:pkgname")` e.g. `ls("package:languageR")`  
`> lsf.str("package: pkgname") # in detail`
- useful functions
  - `head()`, `tail()`, `colnames()`, `summary()`, `str()`

# Task

- Install the package “languageR”
- Load the package
- Check what functions are available from that package using `ls()` and `lsf.str()`
- Explore the data frame “verbs” with functions:
  - verbs
  - `head()`, `tail()`, `colnames()`, `summary()`, `str()`

Data.Frame

열(column) : 속성

속성 명, 변수

행(Row) : 관찰대상

ID	Gender	Brand	Coffee	Movie
1	M	Samsung	Americano	Horror
2	F	APPLE	Latte	Comedy
3	M	LG	Americano	Drama
4	M	APPLE	Cappuccino	Comedy
5	F	LG	Americano	Drama
6	M	Samsung	Cappuccino	Comedy
7	F	APPLE	Americano	Action
8	F	APPLE	Latte	Action
9	M	Samsung	Cappuccino	Horror
10	M	APPLE	Americano	Drama

DF_student_info	=	ID	Gender	Brand	Coffee	Movie
		1	M	Samsung	Americano	Horror
		2	F	APPLE	Latte	Comedy
		3	M	LG	Americano	Drama
		4	M	APPLE	Cappuccino	Comedy
		5	F	LG	Americano	Drama
		6	M	Samsung	Cappuccino	Comedy
		7	F	APPLE	Americano	Action
		8	F	APPLE	Latte	Action
		9	M	Samsung	Cappuccino	Horror
		10	M	APPLE	Americano	Drama



# Task

- Make a data frame named DF\_student\_info with the information on page 8
- `DF_student_info = data.frame(ID = c(1, 2, 3...), Gender = c("M", "F", "M"...), Brand = c("Samsung", "APPLE", "LG"...), ...)`

R file input, output

# Data input

- Data input

- `c()`

- `> mydata <- c(1, 2, 3, 4, 5)`

- Standard input

- `> mydata <- scan()`

- `> print(mydata)`

- File input

- .csv, .txt, excel files, etc.

- CSV file

- `> mydf <- read.csv(file = "C:/path/filename.csv", header=T, sep=",")`

# Task

1. Download 'w\_frequency.xlsx' and open this file in MS Excel.
2. Check the file and save it into the CSV format with '.csv' extension.
3. Start R and retrieve the csv file and assign it into the data frame 'wf'
4. `str(wf)`

# Functions for data frame

- `View(DF)` : show the data frame in the script pane
- `dim(DF)` : the dimension of the data frame; how many rows and columns
- `length(DF)` : the length of the data frame
- `head(DF)` : the top 6 values of the data frame
- `tail(DF)` : the bottom 6 values of the data frame
- `summary(DF)` : Summary of basic statistics for the data frame

# Task

- Check the data of “w\_frequency” with the various functions
- Do the same thing with “verbs” and “DF\_student\_info”

# Data output

- CSV file
  - `write.csv(mydf, file="C:/path/filename.csv")`
- Task
  - Save “verbs”, “DF\_student\_info” as csv files

# Access Data Frame



# Data access (column)

- Specific column (variable)

- access one column

- > newdf= mydf\$colname# just “colname” column

- > newdf= mydf[“colname”] # just “colname” column

- > newdf= mydf[index]

- access columns

- > newdf= mydf[, c(“colname1”, “colname3”, “colname7”)]

- > newdf= subset(mydf, select=c(“colname1”, “colname2“, “colname3”))

- > newdf= mydf[n:m]

- > newdf= mydf[c(x1, x2, x3...)]

# Data access (row)

- specific row (using row numbers)

```
> mydf[2, ] # all of row 2
```

```
> mydf[2:4, ] # all of rows 2 through 4
```

```
> mydf[3, 2:3] # row 3, columns 2 through 3
```

```
> mydf[c(3, 5), 2] # rows 3 & 5, column 2
```

- specific row (using subset())

```
> subset(mydf, colname> "num") # specific amount    ex: > subset(mydf, F1 < 400)
```

```
> subset(mydf, colname=="string") # specific string
```

```
    ex: > subset(mydf, vowel == "e" | vowel == "E")
```

```
> subset(mydf, grepl("pattern", colname) # pattern
```

```
    ex: > subset(md, grepl("sm0", speaker))
```

```
    ex: > subset(md, vowel=="e" & grepl("sf0", speaker))
```

```
    ex: > subset(md, dialect=="s" & gender=="f" & vowel=="e")
```

- specific row (using condition)

```
> newdf= mydf[condition] #mydf[mydf$colname > 10, ] or mydf[mydf$colname == "NP", ]
```

# Task

# Load the data “w\_frequency.csv” and assign the data into data frame ‘md’

1. Print only the verb items whose frequency is equal to or greater than 5
2. Print only the verb items whose length is greater than 8 and whose class is animal.
3. Print a word list in alphabetical order.

# Task

- Save a word list as “v\_list.csv” that contains each word in the data ‘verbs’ only once in alphabetical order.
- You may need to use functions: `sort()`, `unique()`

# Task

- Investigate the double-object verb 'give' comparing frequency of its two alternative structure: double object str(NP NP), single object str(NP PP).
- So, you can answer the question: what percent of sentences with 'give' verbs contain the 'PP' structure?  
e.g. He gave me flowers (NP NP) / He gave flowers to me (NP PP)
- We will use the data 'verbs' in the package 'languageR'. Try to proceed step by step following the procedure:
  1. Install and load the package.
  2. Check the data 'verbs' with functions: str(), colnames(), head() and summary().
  3. Select rows whose verb element is 'give'. Save this data as 'v\_give.txt' for future use.
  4. Select all rows where the verb is 'give' and the structure is 'PP'.
  5. Compute percentage of the 'PP' and 'NP' structure, respectively.
  6. Report the result.

# Task

- Analyze double object construction of the verbs: 'cost', 'offer', 'pay', 'sell' and 'send'
  - Check the ratio of 'NP' and 'PP' structure for each verb
  - Write an R code that analyzes and reports your analyses for each verb and overall comparison.