Session_1

YeEunLee

9/8/2022

Working with R

Useful Resources

- YouTube Channel: https://www.youtube.com/c/RProgramming101
- R tutorial website: https://www.tutorialspoint.com/r/index.htm

R and R studio

- R
- R studio:
  - Convert R into more user-friendly style
  - "The RStudio IDE is a set of integrated tools designed to help you be more productive with R and Python. It includes a console, syntax-highlighting editor that supports direct code execution, and a variety of robust tools for plotting, viewing history, debugging and managing your workspace."

Components of R studio - Four windows

- Console
- Script
- Environment
- Results, Path, Packages, Help

Packages

"R packages are a collection of R functions, complied code and sample data. They are stored under a directory called"library" in the R environment." - tutorialspoint

To use packages ...

1. Install packages (`install.packages("name_of_packages")`)
2. Bring them into my workspace (`library(name_of_packages)`)

```
# Installing Packages
install.packages("ISLR2")
# Save the package into your current workspace
library(ISLR2)
```

Importing data

```
# Importing data
Auto <- read.table("Auto.data")

# Auto data is stored in the "ISLR2" package
attach(Auto)
```

This "Auto.data" is included in the package "ISLR" we installed. Thus, we can use the function "attach". However, it is very unlikely that you are going to use this kind of "stored data" but this "attach" function is sometimes very useful. For example, think about the case you bring a data set and you trimmed the data and made subsample of the original dataset and want to work on that subsample. Also, this attach function is useful because you can just use variable name instead of "dataframe.variable".

Data Types

Popular Data Types

- Logical
- Numeric
- Integer
- Character

```
# Logical
test = 1
v = test == 1
print(class(v))

# Numeric
v <- 105
print(class(v))

# Character
```

```r
v <- "True"
print(class(v))
```

```r
# Check data
View(Auto) # view entire data set
head(Auto) # view first few row of the data
dim(Auto) # identifying the dimension

# Generating subsample
Auto_naomit <- na.omit(Auto) # eliminate na data

dim(Auto_naomit)

# Summary function
summary(Auto_naomit)
summary(mpg)

# Check the data type
print(class(mpg))
print(class(cylinders))
```

Basic Commands

```r
# Vectors
students <- c('Maria','Tom',"Harry")

# Matrices
scores <- matrix( c(90,80,70,50,90,80), nrow = 2, ncol = 3, byrow = TRUE)

# Data.Frame
scoredata <-    data.frame(
   s_name = c("Maria", "Tom","Harry"),
   Math = c(90, 80, 70),
   Science = c(50,90, 80)
)

scoredata_1 = rbind(students,scores) # rbind, cbind
scoredata_2 = t(scoredata_1) # t(), transpose

# generating sequence
order = seq(1:3)
```

```r
# length
x = length(mpg)
y = length(students)

# simple calculation
z = x+y

# indexing
```

```r
d1 = Auto[3,4] # specific row and column
d2 = Auto[3,] # an entire row
```

Function

We can make our own functions!

Keep in mind these two things when you make a function... 1. Input 2. Output

```r
mean_score <- function(x, y) {
# function calculating mean score of each student
result <- (x + y)/2
return(result)}

mean_score(scores[1,],scores[2,])
```

```
## [1] 70 85 75
```

Loops

- "for" loop
- "while" loop

Remember the "STRUCTURE"

```r
# For loop
x <- 1:100
n <- length(x)
X2 = 0

for (i in 1:n){
 X2 = X2 + x[i]
}

# While 1

max = 100
init = 0
val1 = 0

while (init<max){
  val1 = val1 + init
  init = init + 1
}

# while 2
init= 0
val2= 0
```

```
while (init<=max){
  val2 = val2 + init
  init = init + 1
}
```