```
###################################################################
### Computational Policy and Project Analysis              ########
### Lecture title: Lecture 02. Programming Basics          ########
### Developed by. KKIM                                     ########
###################################################################

################### R Basics ###########################
?rpart
library(rpart)
install.packages('rpart')
?rpart
library(rpart)
?rpart

### Help
print("Kim")
help(print)
?print
print('ABC')

### Comments
# Welcome to CPPA Lecture.
# Enjoy!

### Operators
1 + 2   ## addition
1 - 3   ## substraction
1 * 3   ## mulitiplication
2 / 3   ## division
1 %% 3  ## remainder

1 > 2     ## inequality
1*2 == 2  ## equality
!(1 > 2)  ## negation

(2 > 1) & (2 > 3)  ## logical AND
(1 > 2) | (1 > 3)  ## logical OR

################### Variable ###########################

### Assign value to variable
myVariable <- 2.4
a <- 123
b <- 234

myVariable
a
b

### Variable Name
name <- 1
name.first <- 1
file23 <- 1

23 <- 1
23jordan <- 1
.3Ace <- 1
.Ace <- 1
_2323jordan <- 1
a2323jordan <- 1

################### Data Type ###########################

### Data Type
typeof(TRUE)
typeof("Hello")
typeof(3.14)
typeof(1)
typeof(1L)
```

```
?numeric
?integer
?double
?character
?factor

### Character
 ‘Yonsei’
'123'
'''Life is too short, You need love'''

Yonsei is my girlfiend's university     ' #
'Yonsei is my girlfiend's university'   ' # (X)
'Yonsei is my girlfiend"s university'     # (0)
'Yonsei is my girlfriend₩' university'     # (0)

### Factor
sex_vector <- c("Male", "Female", "Female",
                "Male", "Male")
typeof(sex_vector)
factor(sex_vector)

factor_sex_vector <- factor(sex_vector)
print(factor_sex_vector)
# factor_sex_vector <- factor(c("Male", "Female", "Female", "Male", "Male"))
# print(factor_sex_vector)

factor_sex_vector
typeof(factor_sex_vector)
str(factor_sex_vector)
levels(factor_sex_vector)

sex_vector
typeof(sex_vector)
str(sex_vector)
levels(sex_vector)

### Factor - Changing Levels
survey_vector <- c("M", "F", "F", "M", "M")
factor_survey_vector <- factor(survey_vector)
factor_survey_vector
# str(factor_survey_vector)

# Specify the levels of factor_survey_vector
levels(factor_survey_vector)
levels(factor_survey_vector) <- c('Female','Male')
levels(factor_survey_vector)
# factor_survey_vector
# str(factor_survey_vector)

### Factor - Summarizing a factor
# Generate summary for survey_vector
summary(survey_vector)
# Generate summary for factor_survey_vector
summary(factor_survey_vector)

### Changing Levels with Orders
tier <- c("Bronze","Iron","Gold","Silver","Silver","Platinum",
          "Diamond","Challenger","Master","GMaster","Iron")
tier_factor <- factor(tier)
tier_factor
str(tier_factor)

tier_factor_ed <- factor(tier,
                         levels=c("Iron","Bronze","Silver","Gold","Platinum",
                                  # "Diamond",
                                  "Master","GMaster","Challenger"))
tier_factor_ed
```

```r
str(tier_factor_ed)

tier_factor_ed
tier_factor_ed[1] < tier_factor_ed[2]

tier_factor_ed_or <-
  factor(tier, ordered=TRUE,
         levels=c("Iron","Bronze","Silver","Gold","Platinum",
                  "Diamond","Master","GMaster","Challenger"))
tier_factor_ed_or
str(tier_factor_ed_or)
tier_factor_ed_or[1] < tier_factor_ed_or[2]

### Boolean
TRUE
FALSE
T
F
True
False

### Data Type Check
is.numeric('123')
is.integer(123.0)
is.double(123)
is.character(123)
is.factor(123)

### Data Type Conversion
as.numeric('123')
class(as.numeric('123'))
as.integer(123.0)
as.double(123)
as.character(123)
as.factor(123)

as.numeric(c("-.1"," 2.7 ","B"))
as.integer(c("-.1"," 2.7 ","B"))

### Quiz
# 1) Create a variable called “MyLottery” with 7 numbers: 8, 11, 15, 16, 17, 37, 36
MyLottery <- c(8, 11, 15, 16, 17, 37, 36)
MyLottery

# 2) Write down a R command that checks the data type of MyLottery
typeof(MyLottery)

# 3) Convert all values in MyLottery to character
as.character(MyLottery)
MyLottery
MyLottery <- as.character(MyLottery)
MyLottery

# 4) What is the result of the following code?
sum(c(TRUE, FALSE, TRUE, TRUE, T, F, T))

################## Data Import ##########################

pools <- read.csv(file="R file/R file_LEC02/pools.csv")
str(pools)
head(pools)


pools_ed <- read.csv(file="R file/R file_LEC02/pools.csv",
                     stringsAsFactors = TRUE)
str(pools_ed)
head(pools_ed)

pools_ed <-
```

```
  read.csv(file="R file/R file_LEC02/pools.csv",
           stringsAsFactors = TRUE,
           header = FALSE)
head(pools_ed)

## Import table
hotdogs <-
  read.table('R file/R file_LEC02/hotdogs.txt',
             sep = '₩t')
head(hotdogs)

hotdogs <-
  read.table('R file/R file_LEC02/hotdogs.txt',
             sep = '₩t',
             col.names = c("type", "calories", "sodium"))
head(hotdogs)

## Export .csv
# hotdogs$cal.type <- ifelse(hotdogs$calories >= 150,
#                             'heavy',
#                             'light')
hotdogs_meat <-
  hotdogs[hotdogs$type=="Meat",]
head(hotdogs_meat)

write.csv(hotdogs_meat,
          file = "R file/R file_LEC03/newhotdog_meat.csv")
write.csv(hotdogs[hotdogs$type!="Meat",],
          file = "R file/R file_LEC03/newhotdog_notmeat.csv",
          row.names = FALSE)

## Export .table
write.table(hotdogs_meat,
            "R file/R file_LEC03/newhotdog_meat.tsv",
            row.names = FALSE,
            sep = '₩t')

################### R Working Environment ###########################

my.var <- 10
my.var2 <- c(1,4,6,22,3)
my.var3 <- c('John', 'Bob', 'Alice')
my.var4 <- data.frame(A = 1:3, B = 9:11)
save(my.var, my.var2, my.var3, my.var4,
     file = "R file/R file_LEC02/myVariables.RData")
load(file = "R file/R file_LEC02/myVariables.RData")

my.var <- 10
my.var2 <- c(1,4,6,22,3)
my.var3 <- c('John', 'Bob', 'Alice')
my.var4 <- data.frame(A = 1:3, B = 9:11)
save(list=ls(),
     file = "R file/R file_LEC03/myVariables.RData")
load(file = "R file/R file_LEC03/myVariables.RData")
```