```
# Introduction

# R is a programming language specialized in statistical computing and graphics.

# Created in 1992 by two statisticians (Ross Ihaka and Robert Gentleman at the
# University of Auckland).

# Free ( <-> matlab, stata) and open source. Open source means that the source
# code is available to the general public for use or modification.

# Statisticians and econometricians are one of main user groups. They provide thousands
# of R codes (so called "R packages") to implement statistical procedures.
# When they develop a new statistcal method, they usually provide R codes from
# implimentation.
# This is very important for us, who are users of those procedures but not
# developers => We can easily use newly developed statistical learning
# methods.

# Interacting with R

## For Windows and Mac, the standard R download comes with an RGui, which we may use
## for simple tasks.
## Rstudio is often used to write and implement R programs easily and efficiently.
## In class, we will use the RStudio.

# Base R and most R packages are available for download from "CRAN" (Comprehensive R
# Archive Network, www.r-project.org)

## Download the R installer from https://cran.cnr.berkeley.edu/
## Install R by opening the installer and follower the steps.

# Installing RStudio

## Verify that you have already installed R and that you can launch the R application.
## Download the RStudio Desktop installer from www.rstudio.com/ide/download.
## Install RStudio Desktop by opening the installer and following the steps.

# Some basics

## We can write R codes in the console or in the script editor.
## If you use the script editor, you can save the code in the working directory.

## getwd(), setwd(), Session-Set Working Directory - Choose Directory

## Anything that R creates can be assigned to a variable. Includes data,
## functions and the results of

## Case sensitive to define a variable.

## Help files for R functions are accessed by typing the function name with ?
## (e.g., ?lm)

## If you execute the command that is not complete, the prompt ">" changes to "+".
## You can escape from this by pressing ESC.

## Some built-in data in R: you can find them with "data()"

################################################################################
# Let's try a few of random commands'

100

n <- 1000                          # Create a variable n and assign 1000

"hello world!"
hello world!

a <- "hello world"

```

```
# If you use the script editor, commands are separated either by ; or by a new line.
# You can add a comment that is not executed after '#'

x <- rbinom(n, 10, 0.5)             # Generate 1000 binomial random numbers (# of success)
                                    # with size 10.
hist(x)                             # ?hist() or help(hist)
                                    # Create a histogram of x.

ls()                                # See variable names that are stored in the workspace.
rm(x)                               # Remove x from the workspace
x

x <- seq(from=1, to=20, by=0.5)     # Make x = (1,1.5,2,...,19,19.5,20) increment is 0.5.
x                                   # Print x (you can also write print(x))

a <- 2
b = 2
e <- rnorm(length(x),0,4)           # The size of e is set to the same as x
y <- a + b*x + e
reg <- lm(y~x)                      # Run a linear regression y on x with a constant and
                                    # store the result in "reg"
summary(reg)                        # Present the summary of regression results.

plot(x,y)                           # Make a scatterplot (x,y)
abline(a,b,lty=3, col="blue",lwd=3)           # Add in the true regression line.
                                    # (what are "lty" and "color" ?par)
abline(reg,lty=1, col="red",lwd=3)            # Add in the estimated regression line.
?abline
```