R Handout 1

Introduction to R

- R is a system for statistical analyses and graphics created by Ross Ihaka and Robert Gentleman.
- It is both a software and a language considered as a dialect of the S language created by the AT & T Bell Laboratories
- R is freely distributed under the terms of the General Public License
- It can be downloaded free from `http://www.r-project.org`
- Another powerful user interface for R is RStudio, which can be downloaded from `http://www.rstudio.com`
- The development and distribution of R are carried out by a group of statisticians known as the R Development Core Team
- R has many inbuilt functions for statistical analyses and graphics. The graphics are visualized immediately in their own window and can be saved in different formats, for instance, jpg, png, bmp, ps, pdf, etc
- The results from a statistical analysis are displayed on the screen and can be saved, written on a file, or used in subsequent analyses
- Unlike other classical softwares which displays immediately the results of an analysis, R stores the results in an "object", and as such, analysis can be carried out with no results displayed. This feature of R is useful in the sense that the user can extract only the part of the result which is of interest
- To get on-line help on how to use R functions, type ?functionname

Numbers, Vectors, and Operations

- To create a vector x with 5 elements:
  x < − c(2.1,3.2,5.6,2.4,5) or x=c(2.1,3.2,5.6,2.4,5) or assign("x",c(2.1,3.2,5.6,2.4,5))
- Try the following: length(x), x[1], x[4], x[length(x)], x[1:3], x[-2], x[-1:-3], xˆ2, x+1, x/10, 1/x, sort(x), sort(x,decreasing=T), sum(x), mean(x), sd(x), var(x), min(x), max(x)

- Create another vector y with the same length as of x:
  y=c(10,20,30,40,50) and try the following: x+y, x-y, x/y, x*y, 2x+y
- Creating a data frame (vectors must be of the same length)
  data=data.frame(x,y), data1=data.frame(expenditure=x,income=y), data1$ expenditure
- Creating a sequence of numbers: seq(1,10,by=1)
- Repeating a number a desired number of times: rep(10,6)
- Logical operators: $<, <=, >, >=, ==, !=$ (will return TRUE or FALSE)
  try out the following: a=5, a<5, a<=5, a>5, a>=5, a! =5, a==5, a> 2 & a< 6, a>2 & a<4, a>2 | a<4

Missing Values

- Missing values in R are denoted by "NA" (not available). A second type of "missingness" is denoted by "NaN" (not a number), for instance, 0/0, (1/0)/(1/0), etc.

Reading Data from Files

- For reading files, R uses the working directory
- To find your working directory, type getwd()
- To set your own working directory, type setwd()
- Reading data set from a text file (.txt) saved on your computer: read.table("filename.txt")
- If the data is available in a spread sheet (.xls or .xlsx), then, you can save the spread sheet as a text file
- To read data from R's package, perform the following steps:
  - Check if the package is already installed using library(packagename)
  - If the package is not installed, perform the following: (1) In R console, click Packages (2) click Install packages(s) which will open the CRAN mirror (3) from CRAN mirror, chose your location and click OK, which will open a list of all R packages (4) from Packages, select the required package and click OK.

    To avoid the above steps for installing a package, simply use the function install.packages("packagename")
  - data(nameofdataset)
  - nameofdataset
- For example, read the data set "cars" (which gives the speed and stopping distances of cars) from the R package "datasets"

Data and Graphical Representation

- five number summary and quantiles: Consider the same example as in Assignment 2, question 1.

  - data=c(123, 116, 122, 110, 175, 126, 125, 111, 118, 117)
  - To calculate the five number summary: fivenum(data)
  - To calculate the quantiles: quantile(data), quantile(data,type=...), quantile(data,type=6). Note: the default type is 7
  - To specify the probabilities: quantile(data,prob=c(0.25,0.5,0.75,0.9),type=6)

- Box plot:

  - boxplot(data, range=1.5)
  - boxplot(data, range=1.5, main="box plot of data", ylab="income", col="red")
  - boxplot(data, range=1.5, main="box plot of data", xlab="income", col="red",horizontal=T)

- Side by side boxplots:

  - data2=c(150,120,115,180,160,142,136,148,123,102,139)
  - boxplot(data, data2, range=1.5, main="box plots of two data sets")
  - boxplot(data, data2, range=1.5, names=c("male","female"), main="box plots of two data sets", col=c("red","blue"))

- Simple scatter plot

  - From the dataset "cars", draw a scatter plot of speed against distance by using the function "plot"

- QQ plot

  - Check if the normality assumption is true for the following data sets by using the function "qqnorm":
    - data1=c(0,-0.3,-0.1,-0.5,-0.4,2.8,2.6,-1.3,0.5,2.6)
    - data2=c(0.4967,0.2028,1.0744,-1.9826,-0.2272,-0.9323,0.0727,0.7041,-1.2972,1.9424)

- Histogram

  - generate a random sample of size 1000 from a normal distribution with mean 50 and sd 5 and draw a histogram using the function "hist" (use the function "rnorm")
  - repeat the above by generating a random sample from a chi-square distribution with 10 degrees of freedom (use the function "rchisq")

Finding percentage points of continuous distributions

- use the functions “qnorm”, “qt”, “qchisq”, and “qf” to find the upper and lower percentage points of normal, t, chi-square, and F distributions, respectively

Hypothesis testing

z-test for population mean(s)

Install the package BSDA

- **single sample:** z.test(x, alternative=c(“two.sided”,“greater”,“less”), mu=. . . , sigma.x=. . . , conf.level=0.95)

  example: data=rnorm(20, mean=0, sd=1)

  z.test(x=data, alternative=“two.sided”, mu=0, sigma.x=1, conf.level=0.95)

- **two sample:** z.test(x, y, alternative=c(“two.sided”,“greater”,“less”), mu=. . . , sigma.x=. . . , sigma.y=. . . , conf.level=0.95)

  example: data1 = c(7.8, 6.6, 6.5, 7.4, 7.3, 7.0, 6.4, 7.1, 6.7, 7.6, 6.8) and data2 =c(4.5, 5.4, 6.1, 6.1, 5.4, 5.0, 4.1, 5.5). Assume $\sigma_x = 0.5, \sigma_y = 0.5$

  z.test(x=data1, y=data2, alternative=“two.sided”, mu=0, sigma.x=0.5, sigma.y=0.5, conf.level=0.95)

t-test for population mean(s)

- **single sample:** t.test(x, alternative=c(“two.sided”,“greater”,“less”), mu=. . . , conf.level=0.95)

  example: data=rnorm(35,mean=5,sd=2)

  t.test(x=data, alternative=“greater”, mu=4, conf.level=0.99)

- **two sample - variance unknown but equal:** t.test(x, y, alternative=c(“two.sided”, “greater”,“less”), mu=. . . , var.equal=TRUE, conf.level=0.95)

  example: treatment=c(2.1, 5.3, 1.4, 4.6, 0.9) and nontreatment=c(1.9, 0.5, 2.8, 3.1)

  t.test(x=treatment, y=nontreatment, alternative=“greater”, mu=0, var.equal=TRUE, conf.level=0.95)

- **two sample - variance unknown but unequal:** t.test(x, y, alternative=c(“two.sided”, “greater”,“less”), mu=. . . , var.equal=FALSE, conf.level=0.95)

- **two sample - paired:** t.test(x, y, alternative=c("two.sided", "greater","less"), mu=. . . , paired=TRUE, conf.level=0.95)

  example: Xray=c(2.0, 2.0, 2.3, 2.1, 2.4) and Chemical=c(2.2, 1.9, 2.5, 2.3, 2.4)

  t.test(x=Xray, y=Chemical, alternative="two.sided", mu=0, paired=TRUE, conf.level=0.95)

F-test for equality of two variances

var.test(x, y, ratio=1, alternative=c("two.sided","greater","less"), conf.level=0.95)

example: data1=rnorm(50, mean=0, sd=2) and data2=rnorm(30, mean=1, sd=1)

var.test(x=data1, y=data2, alternative="two.sided", conf.level=0.95)

testing for proportion(s)

- **single sample**

  prop.test(x, n, p=. . . , alternative=c("two.sided", "greater", "less"), correct=FALSE, conf.level=0.95)

  example (one used in lecture slides): prop.test(x=28, n=90, p=0.25, alternative="greater", correct=FALSE, conf.level=0.95)

- **two sample**

  prop.test(x=c(x1, x2), n=c(n1, n2), p=. . . , alternative=c("two.sided", "greater", "less"), correct=FALSE, conf.level=0.95)

  example (one used in lecture slides): prop.test(x=c(20, 10), n=c(200, 150), alternative="greater", correct=FALSE, conf.level=0.95)

Simple linear regression

- a=lm(y $\sim$ x, data=dataframe)
- summary(a)
- a$fitted.values gives you the predicted values of y
- a$residuals gives you the residuals (observed - predicted)