# Applied Probability and Statistics Lab

For KTU MCA

Dr.Sunil Thomas Thonikuzhiyil[1]

August 26, 2016

[1]CollegeofEngineeringAttingal

Dedicated to Our Supreme Lord DINKAN.


ചിങ്കൻ

# Contents

# List of Figures

# List of Tables

# Preface

## Acknowledgements


-

# 1

# Visualizing Data

## What we will learn

In this experiment we will try to learn the preliminaries of R along with methods of visualizing data in R

Tables, charts and plots. Visualizing Measures of Central Tendency, Variation, and Shape. Box plots, Pareto diagrams. How to find the mean median standard deviation and quantiles of a set of observations.

## 1.1 Introduction to R

To be written

### 1.1.1 Installing R

Platforms Window and linux

### 1.1.2 Installing R studio

Prerequisites

## 1.2 R fundamentals

This is a review of R fundamentals. No details are covered. It is assumed that the participant is familiar with programming.

### 1.2.1 Basic Math operations in R

Type the following on > prompt and see the results.

```
1 + 1
5 - 3
```

```
3 * 3
4 / 5
4 ^ 2  # Exponetiation
5 %% 2  # Modulus
5 %/% 2 # integer division
```

### 1.2.2 Variables and assignment in R

There is no need to declare variables in advance. Rules for variable names are almost similar to those in C programming language. There are two assignment : + and <-. The preferred assignment operator is <- . Try the following.

```
x=20  # assign x=20
y<-3  # assign y =3
x     # Display x
y     # Display y
x=y
x
y<-x*10
```

A variable can be removed using rm() function.

```
rm()  # remove x
x     # try printing x after removal
```

### 1.2.3 R Data Types

R has a wide variety of data types including scalars, vectors (numerical, character, logical), matrices, data frames, and lists. Let us quickly review them.

```
# An integer can be assigned to a variable by suffixing L
x=5L
class(x)  # This prints the data type of x

# numeric
y=5.3
class(y)

# Character string
y=" hello"
class(y)

# date
date=as.Date("2016-10-16")
date
```

### 1.2.4 Logical operators

Similar to C > < <= >= == !=

```
5>6
```

### 1.2.5 Control statements

#### if-else

```
 x <- 5
if(x > 0){
   print("Positive number")
}
```

#### ifelse

```
 a = c(5,7,2,9) # a is a vector. The function c creates vector a
 a   #print a
 ifelse(a %% 2 == 0,"even","odd")
```

#### for loops

```
 x <- c(2,5,3,9,8,11,6) # x is a vactor
count <- 0
for (val in x) {
    if(val %% 2 == 0) count = count+1
}
print(count)
```

#### while loops

```
 i <- 1

while (i < 6) {
   print(i)
   i = i+1
}
```

## 1.3 R data structures

Here we will review and learn basic data structures commonly used in R.

### 1.3.1 Vector

A fundamental R data structure is the vector, which stores an ordered set of values called elements. A vector can contain any number of elements. However, all the elements must be of the same type; for instance, a vector cannot contain both numbers and text.

You can build a vector as below using the c() function. c functions combines several entities of to a vector.

```
state<- c("Kerala","Tamil Nadu"," Maharashtra")
```

A vector can be printed by typing its name at the R prompt.

```
state
```

Let us create several vectors.

```
temperature <- c(98.1, 98.6, 101.4)
capital <- c("tvm","chennai","mumbai")
drought <- c(FALSE, FALSE, TRUE)
```

Let us print them.

```
state
temperature
capital
drought
```

You can access individual elements using square brackets.

```
temperature[2]
```

You can create a vector containing a sequence of numbers using :

```
x=10:50
x
x[3]
y=-2:5
z=10:1
y
z
```

**Vector Operations**

```
x*3

x/4
x+2
sqrt(x)
y=1/x
y
x=1:10
y=10:1
x+y
x-y
x*y
x/y
length(x)
m=c(1,2)
x
x+m
x<5
```

### 1.3.2 Factors

Factor is a data structure used for fields that takes only predefined, finite number of values (categorical data). For example, a data field such as marital status may contain only values from single, married, separated, divorced, or widowed. In such case, we know the possible values beforehand and these predefined, distinct values are called levels

```
status <- factor(c("single","married","married","single"));
status
```

### 1.3.3 Lists

Let us create and print a list.

```
n = c(2, 3, 5)
s = c("aa", "bb", "cc", "dd", "ee")
b = c(TRUE, FALSE, TRUE, FALSE, FALSE)
x = list(n, s, b, 3) # x contains copies of n, s, b

x
```

Display list elements.

```
x[2]
```

```
x[[2]]
x[[2]][1]
```

### 1.3.4 Data Frames

A data frame is used for storing data tables. It is a list of vectors of equal length. For example, the following variable df is a data frame containing three vectors n, s, b.

```
n = c(2, 3, 5)
s = c("aa", "bb", "cc")
b = c(TRUE, FALSE, TRUE)
df = data.frame(n, s, b)

# print the data frame df
df
```

Accessing rows and columns of data frame.

```
df[1]
df[[1]]
df$n
df[2,2]
```

**Examining some built in data frames.**

There are several built in data frames you can use. mtcars is a simple frame about car parameters.

```
mtcars
help(mtcars) # read the documentation.
```

Let us experiment with this data set.

```
mtcars[1, 2]

mtcars["Mazda RX4", "cyl"]
mtcars["Mazda RX4",]
nrow(mtcars)

ncol(mtcars)
head(mtcars)
mtcars[[9]]

mtcars[["am"]]
mtcars$am
mtcars[1]
```

```
mtcars["mpg"]
mtcars[c("mpg", "hp")]
mtcars[24,]

mtcars[c(3, 24),]
mtcars["Camaro Z28",]
mtcars[c("Datsun 710", "Camaro Z28"),]
```

### 1.3.5 Exercises

Examine iris data set.

### 1.3.6 Importing Data

Download the iris data set from . Read the background information from wikipedia.

Data can be stored in a comma separated value format Common extension is .csv. You can read a csv file as below. It is assumed that the file is in your current directory. If you get an error here most probably you are in wrong directory or the data file is missing. Please download the data file and place it in your current directory.

```
mydata = read.csv("iris.csv") # read csv file. Make sure that you
    have the csv file
```

If you get an error, please try out the following commands.

```
getwd() # This command prints your current directory

setwd("/home//sunil") # use this commnad to set the direcotry.


mydata = read.csv("iris.csv") # read csv file

mydata
str(mydata ) # display the structure of mydata variable
```

### 1.3.7 Find out statistics of your data.

```
mean(mtcars$mpg)
median(mtcars$mpg)
sd(mtcars$mpg)
summary(mtcars)
```

```
range(mtcars$mpg)
diff(range(mtcars$mpg))


IQR(mtcars$mpg) # I am not explaining this. Find out from
    documentation.:D
```

Repeat the above for iris data set.

## 1.4 Plotting in R

### 1.4.1 Box Plots

Boxplots are a measure of how well distributed is the data in a data set. It divides the data set into three quartiles. This graph represents the minimum, maximum, median, first quartile and third quartile in the data set. It is also useful in comparing the distribution of data across data sets by drawing boxplots for each of them.

```
boxplot(mpg ~ cyl, data = mtcars, xlab = "Number of Cylinders",
   ylab = "Miles Per Gallon", main = "Mileage Data")
```

### 1.4.2 Histograms

```
 # Create data for the graph.
v <- c(9,13,21,8,36,22,12,41,31,33,19)
hist(v,xlab = "Weight",col = "yellow",border = "blue")


hist(mtcars$mpg, main = "Histogram of miles per gallon",
xlab = "mpg")
```

### 1.4.3 Line graph

```
 v <- c(7,12,28,3,41)
plot(v,type = "o")



# multiple lines
```

```
v <- c(7,12,28,3,41)
t <- c(14,7,6,19,3)
plot(v,type = "o",col = "red", xlab = "Month", ylab = "Rain fall",
   main = "Rain fall chart")
lines(t, type = "o", col = "blue")


\end{lstlisting
\subsection{Scatter plots}

\begin{lstlisting}[language=R]
 plot(x = mtcars$mpg,y = mtcars$cyl)
 help (plot )
 plot(x = mtcars$mpg, y = mtcars$hp,
main = "Scatterplot of mpg vs. hp",
xlab = "mpg",
ylab = "hp")


pairs(~wt+mpg+disp+cyl,data = mtcars,
   main = "Scatterplot Matrix")
```

# 2

# Probability Distributions

## 2.1 Simulating Coin Toss

The probability of getting a Heads or a Tails on a coin toss is both 0.5. We can use R to simulate an experiment of flipping a coin a number of times and compare our results with the theoretical probability.

0 = Tails and 1 = Heads

```
sample(0:1,15,rep=T) #flip 15 times
```

# 3

# Random Samples

# 4

# Binomial Distribution and Central Limit theorem

# 5

# Confidence Intervals

# 6

# Correlation

# 7

# Regression