# MA208: Assignment - R Language

Due on Wednesday, February 10, 2016

*Submitted to Dr. Vishwanath K. P*

**Kishan Desai 13CO212**

10th Feb, 2016

# Contents

# Introduction

R is a software language for carrying out complicated (and simple) statistical analyses. It includes routines for data summary and exploration, graphical presentation and data modelling. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories by John Chambers and colleagues. R can be considered as a different implementation of S.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

One of Rs strengths is the ease with which well-designed publication-quality plots can be produced, including mathematical symbols and formulae where needed. Great care has been taken over the defaults for the minor design choices in graphics, but the user retains full control.

## Setting up R in Windows

For Setting up R in windows- go to url: https://cran.r-project.org/bin/windows/base/. Download the software package for R. Then install the file in windows using install manager. Open R compiler and start screen will look like this:

Figure 1: R window

## R-basics

R stores information and operates on objects. The simplest objects are scalars, vectors and matrices. But there are many others: lists and dataframes for example. In advanced use of R it can also be useful to define new types of object, specific for particular application.

Sample arithmetic can be done as:

```
> 4+5
[1]  9
```

We can assign objects values for subsequent use. For example:

```
> bubba <- c(3,5,7,9)
>
```

At any time we can list the objects which we have created:

```
> ls()
[1] ”x” ”y” ”z”
```

There are many standard functions available in R, and it is also possible to create new ones. Vectors can be created in R in a number of ways. We can describe all of the elements:

```
> vec = c(5,9,1,0)
> vec
[1] 5 9 1 0
```

- There are many inbuilt functions in R for statistical analysis.
- Such as mean(), median() and var() are some of the popular measures.

So for our example:

```
> mean(vec)
[1] 3.75
> median(vec)
[1] 3
> var(vec)
[1] 16.91667
```

We can make different instances of same data type. (# is used for comments)

```
> vec.draft1 = c(5,9,0,1)
> vec.draft2 = typos.draft1      # make a copy
> vec.draft2[1] = 0              # assign the first page 0 typos
```

R makes it easy to translate mathematics in a natural way once your data is read in. We can define user defined functions in R as follow:

```
> arr = c(100, 158, 75, 69, 104, 110, 115, 112)
> max(arr)
[1] 158
> fun = function(x) sqrt(var(x))
> fun(arr)
[1] 27.26556
```

## Plotting Functions

### Bar Charts

A bar chart draws a bar with a a height proportional to the count in the table. The height could be given by the frequency, or the proportion. The graph will look the same, but the scales may be different.
Sample bar chart is given:

```
> var = c(1,2,1,4,1,5,4,1,2,5,4,4,3,1)
> barplot(table(var))
```

So we will get bar chart like this:

Figure 2: Bar chart

## Pie Charts

The same data can be studied with pie charts using the pie function. We use the same data as above:

```
> var = c(1,2,1,4,1,5,4,1,2,5,4,4,3,1)
>  pie(beer.counts,col=c(”purple”,”green2”,”cyan”,”white”))
```

So we will get pie chart like this:

Figure 3: Pie chart

## Numerical data

To describe a distribution we often want to know where is it centered and what is the spread. These are typically measured with mean and variance (or standard deviation), or the median and more generally the five-number summary. The R commands for these are mean, var, sd, median, fivenum and summary.

Summary command gives mean, median, max, mean, 25 and 75 quantile of the data. ( The p quantile, (also known as the 100pin the data where 100p% is less, and 100(1-p)% is larger )

```
> data = c(15,20,26,14,25,39,12)
> summary(data)
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   12.00   14.50   20.00   21.57   25.50   39.00
```

### Histogram

The histogram defines a sequence of breaks and then counts the number of observation in the bins formed by the breaks. It plots these with a bar similar to the bar chart, but the bars are touching. The height can be the frequencies, or the proportions.
Sample program look like:

```
> var =c(2,5,2,1,4,3,3,6,2,1,2,4,4,6,6,6,1,2,4,2,3,5,6)
> hist(var)
```

So we will get histogram like this:

Figure 4: Pie chart

## Bivariate data

Bivariate data is used to combine two variables and it is summarized using table. We can use bivariate data by creating two independent vectors and then combining them using table command.

```
> car = c(”Maruti”,”Hyundai”,”Maruti”,”Maruti”,”Hyundai”,”Maruti”,”Hyundai”,”Maruti”,”Hyun
> amount = c(2,1,1,2,1,3,3,2,3,1)
> table(car,amount)
         amount
car       1 2 3
  Hyundai 2 0 2
  Maruti  2 3 1
> barplot(table(car,amount),col=c(”purple”,”green2”),
+ beside=TRUE)
```

Figure 5: Table chart

## Conclusion

The R programming language is an important tool for development in the numeric analysis and machine learning spaces. With machines becoming more important as data generators, the popularity of the language can only be expected to grow.

The main advantages of R language are:

- R is a powerful scripting language
- Graphics and data visualization
- Integration with document publishing
- Access to powerful, cutting-edge analytics
- No cost

The R programming language is mainly used by data scientists and statisticians to extract or data mine information from a large data set or surveys.