

Introduction to R, Setup, and Tutorial

Part 1

Introduction to R:

What is R?

<https://www.r-project.org/about.html>

"R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

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."

The R environment:

"R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either on-screen or on hardcopy, and
- a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities."

Most importantly: J-O-B-S! <https://www.indeed.com/q-R-Statistics-jobs.html>

Part 2

Requirements (and helper videos):

1. **What is R? What is RStudio? (And, downloads):**
http://qcitr.com/vids/R_Development_Environment_Setup.mp4
2. **Learn to Use R (includes A5 requirements):** http://qcitr.com/vids/R_Introduction.mp4
3. ***Carefully* go through the entire following tutorial (up to and including p. 32):**
LEARN TO USE R Your hands-on guide: <http://core0.staticworld.net/assets/2015/02/17/r4beginners.pdf>

Note: When the above tutorial is successfully completed, the below deliverables will be fulfilled.

Deliverables (see example screenshots below):

1. **R Commands:** save a file of all the R commands included in the tutorial.
2. **R Console:** save a screenshot of some of the R commands executed above (below example).
3. **Graphs:** save at least 2 separate image files displaying graph plots created from the tutorial.
4. **RStudio:** save one screenshot (similar to the one below), displaying the following 4 windows:
 - a. R source code (top-left corner)
 - b. Console (bottom-left corner)
 - c. Environment (or History), (top-right corner)
 - d. Plots (bottom-right corner)

The screenshot displays the RStudio interface with three main panes: Source, Environment, and Console.

Source Pane: Contains R code for data manipulation and plotting. The code includes loading the `mtcars` dataset, creating a bar chart of `mpg` by `hp`, and filtering the data to show only cars with `mpg > 20`. The final plot is titled "Test scores descending".

```
# data()
# barChart(AAPL)
# barChart(AAPL, subset="last 14 days")
# chartSeries(AAPL, subset="last 14 days")
# chartSeries(AAPL, subset="last 14 days")
# chartSeries(AAPL, subset="last 14 days")
# barChart(AAPL[, 2013-01-1:2013-04-12])
# rm(x)
# str(titanic)
# colnames(titanic)
# rownames(titanic)
# cor(solar.test)
# choose(15,4)
# mypop <- c("Sub", "Joane", "Sally", "Fin", "Neal")
# combn(mypop, 2)
# mtcars$mpg
# mtcars[, 2:5]
# mtcars[, c(2,4)]
# mtcars$mpg-20
# mtcars[mtcars$mpg>20,]
# mtcars[mtcars$mpg>20, c(1,4)]
# mtcars[mtcars$mpg>20, c("mpg", "hp")]
# mpg20 <- mtcars$mpg > 20
# cols <- c("mpg", "hp")
# mtcars[mpg20, cols]
# attach(mtcars)
# mpg <- mpg > 20
# detach()
# subset(mtcars, mpg>20, c("mpg", "hp"))
# subset(mtcars, mpg=max(mpg))
# subset(mtcars, mpg, hp)
# subset(mtcars, ~ c("mpg", "hp"))
# subset(mtcars, select=c("mpg", "hp"))
# install.packages("dplyr", dependencies = TRUE)
# update.packages(ask = FALSE, checkBuilt = TRUE)
# install.packages()
# library(dplyr)
# filter(mtcars, mpg>20)
# select(mtcars, mpg, hp)
# NOTE: does "NOT" work! Incorrect pipe syntax!
# mtcars %>% filter(mpg > 20) %>%
# select(mpg, hp)
# se .... [TRUNCATED]
# barChart(AAPL)
# barChart(AAPL, subset="last 14 days")
# chartSeries(AAPL, subset="last 14 days")
# chartSeries(AAPL, subset="last 14 days")
# ... [TRUNCATED]
# filter(mtcars, mpg>20)
# select(mtcars, mpg, hp)
# NOTE: does "NOT" work! Incorrect pipe syntax!
# mtcars %>% filter(mpg > 20) %>%
# se .... [TRUNCATED]
# mpg, hp
# 1 21.0 110
# 2 21.0 110
# 3 22.8 93
# 4 21.4 110
# 5 24.4 62
# 6 22.8 95
# 7 32.4 66
# 8 30.4 52
# 9 31.9 65
# 10 21.5 97
# 11 27.3 66
# 12 26.0 91
# 13 30.4 111
# 14 21.4 109
# mtcars %>%
# filter(mpg > 20) %>%
# select(mpg, hp) %>%
# arrange(desc(mpg))
# mpg, hp
# 1 37.9 65
```

Environment Pane: Shows the global environment with variables `mtcars`, `mpg20`, `cols`, and `mpg_hp`.

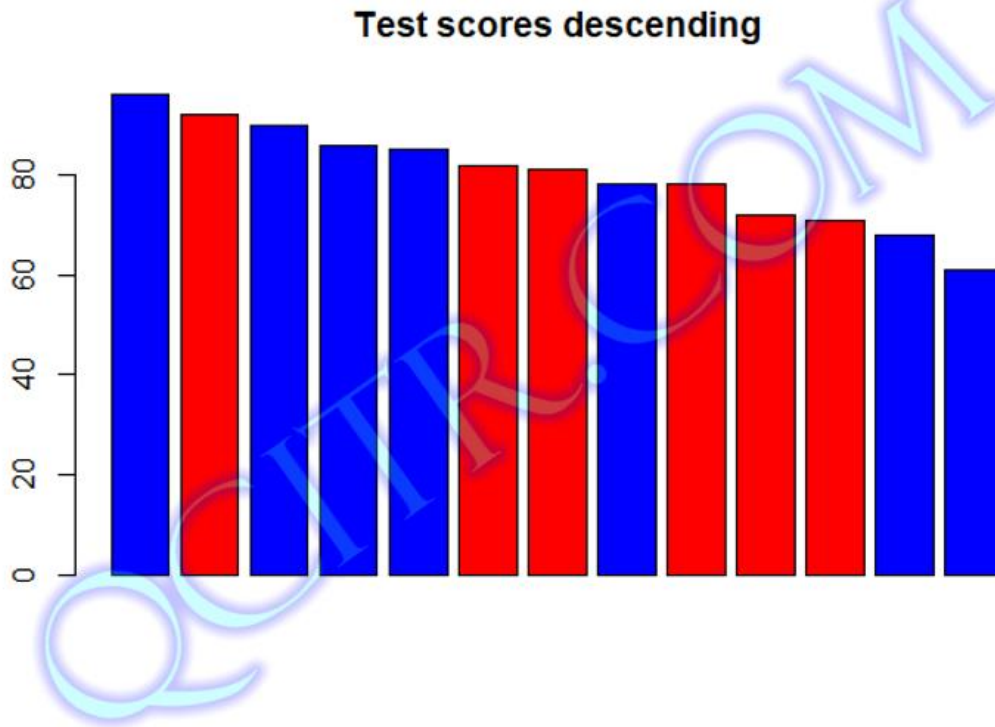
Console Pane: Displays the output of the R code, including the execution of `barChart`, `chartSeries`, and `filter` functions, and the resulting data for the bar chart.

Bar Chart: A bar chart titled "Test scores descending" showing the distribution of `mpg` values for cars with `mpg > 20`. The x-axis represents the car index (1 to 14), and the y-axis represents the `mpg` value (0 to 80). The bars are colored blue and red, alternating in pairs.

Index	mpg
1	37.9
2	37.9
3	30.4
4	30.4
5	27.3
6	27.3
7	26.0
8	26.0
9	24.4
10	24.4
11	22.8
12	22.8
13	21.5
14	21.5

factor(cyl)	Frequency
4	12
6	7
8	14

Learn to Use R (Graph 2)



R and Python Comparisons

Source: Python for R Users: A Data Science Approach

	R	Python (using pandas package*)
Getting the names of rows and columns of data frame "df"	<code>rownames(df)</code>	<code>df.index</code>
	<i>returns the name of the rows</i>	<i>returns the name of the rows</i>
	<code>colnames(df)</code>	<code>df.columns</code>
	<i>returns the name of the columns</i>	<i>returns the name of the columns</i>
Seeing the top and bottom "x" rows of the data frame "df"	<code>head(df,x)</code>	<code>df.head(x)</code>
	<i>returns top x rows of data frame</i>	<i>returns top x rows of data frame</i>
	<code>tail(df,x)</code>	<code>df.tail(x)</code>
	<i>returns bottom x rows of data frame</i>	<i>returns bottom x rows of data frame</i>
Getting dimensions of data frame "df"	<code>dim(df)</code>	<code>df.shape</code>
	<i>returns in this format: rows, columns</i>	<i>returns in this format: (rows, columns)</i>
Length of data frame "df"	<code>length(df)</code>	<code>len(df)</code>
	<i>returns no. of columns in data frames</i>	<i>returns no. of columns in data frames</i>