Data Programming with R

Karl Ho

9/21/2022

Table of contents

Preface

This course requires no prior experience in programming. Yet, if you have some programming experience (e.g. SPSS, Stata, HTML), it will be helpful. R is an interpreted languages. In other words, the programs do not need compilation but will run in an environment to get the outputs. In this course, that is RStudio.

All packages and accounts are free and supported by open sources. It is recommended students bring their own computers (not mobile device) running MacOS, Linux or Windows operating systems.

Recommended software and IDE's:

1. R version 4.2.1 or later (https://cran.r-project.org)
2. RStudio version 1.2.x (https://www.rstudio.com)
3. Text editor of own choice (e.g. Atom, Sublime Text, Ultraedit)

Recommended websites/accounts:

1. GitHub (https://github.com)

2. RStudio Cloud

1 Introduction

This chapter introduces the general principles for data programming or coding involving data. Data programming is a practice that works and evolves with data. Unlike the point-and-click approach, programming allows the user to manage most closely the data and process data in more effective manner. Programs are designed to be replicable, by user and collaborators. A data program can be developed and updated iteratively and incrementally. In other words, it is building on the culminated works without repeating the steps. It takes debugging, which is the process of identifying problems (bugs) but, in fact, updating the program in different situations or with different inputs when used in different contexts, including the programmer himself or herself working in future times.

1.1 Principles of Programming

Social scientists Gentzkow and Shapiro (2014) list out some principles for data programming.

1. Automation
   - For replicability (future-proof, for the future you)
2. Version Control
   - Allow evolution and updated edition
   - Use Git and GitHub
3. Directories/Modularity
   - Organize by functions and data chunks
4. Keys
   - Index variable (relational)
5. Abstraction
   - KISS (Keep in short and simple)
6. Documentation
   - Comments for communicating to later users

7. Management
    - Collaboration ready

1.2 Functionalities of Data Programs

A data program can provide or perform :

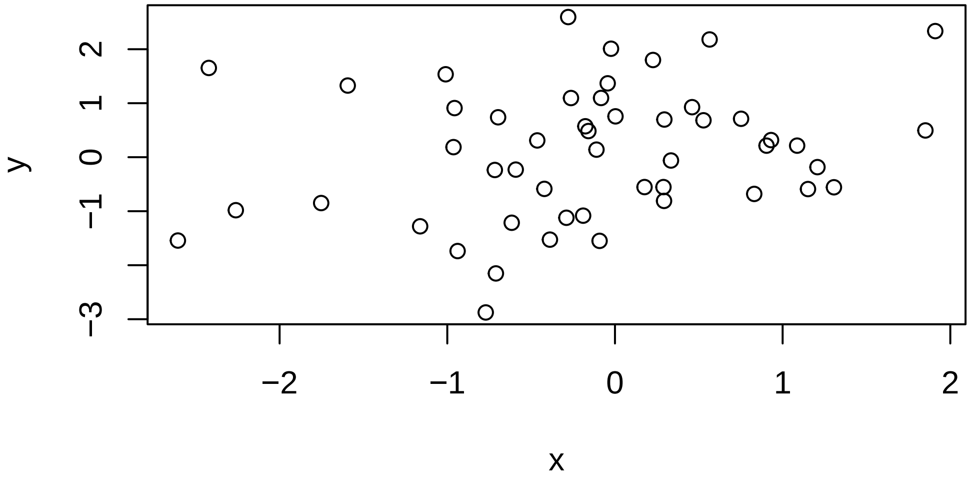
1. Data source
2. Documentation of data
3. Importing and exporting data
4. Management of data
5. Visualization of data
6. Data models

Sample R Programs:

R basics

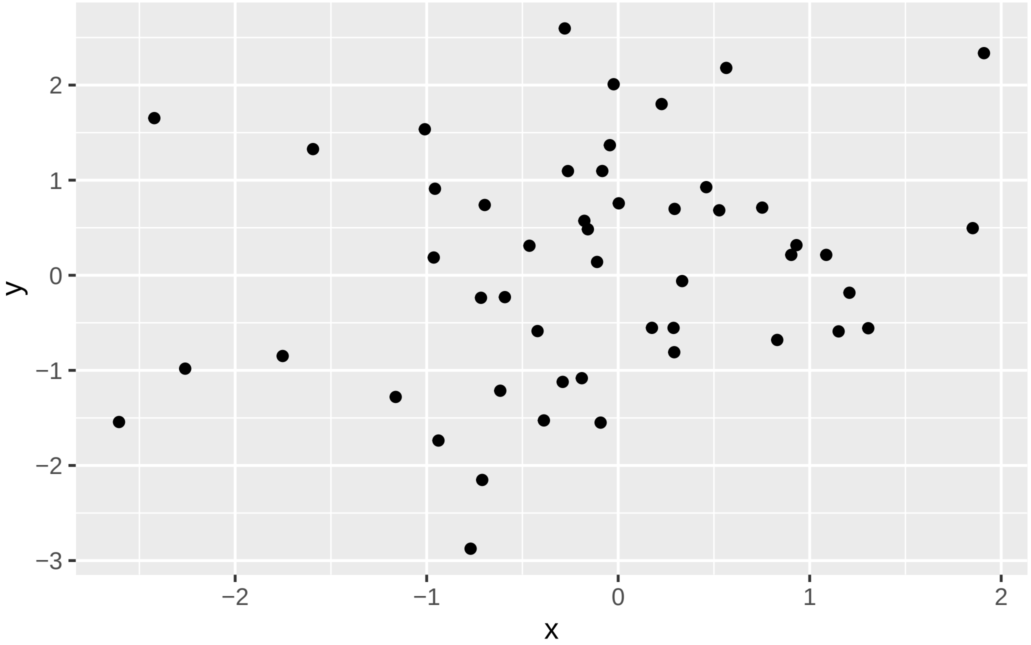
```
# Create variables composed of random numbers
x <-rnorm(50)
y = rnorm(x)

# Plot the points in the plane
plot(x, y)
```



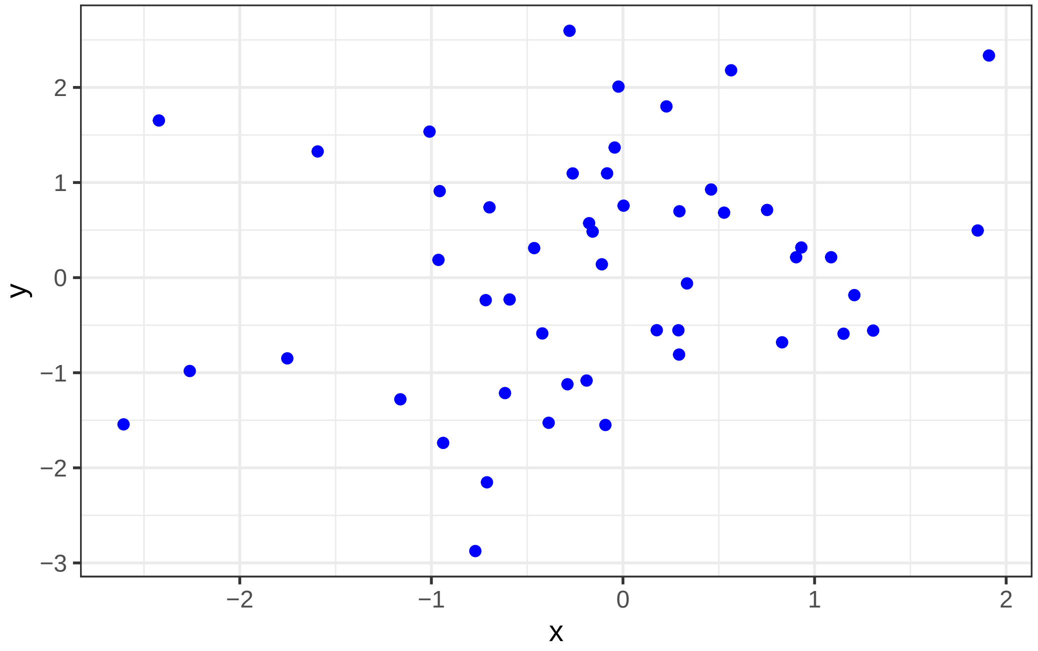
Using R packages

```
# Plot better, using the ggplot2 package
## Prerequisite: install and load the ggplot2 package
## install.packages("ggplot2")
library(ggplot2)
qplot(x,y)
```



More R Data Visualization

```
# Plot better better with ggplot2
ggplot(,aes(x,y)) + theme_bw() + geom_point(col="blue")
```


2
1
0
−1
−2
−3
y
−2
−1
0
1
2
x

2 Summary

This book is designed for the short course titled **Data Programming with R**, offered in the University of Texas at Dallas. It provides training for aspiring data scientists to program surrounding data using R. The primary goal is to build a comprehensive program preparing students in four major elements in Data Science:

1. Data collection
2. Data visualization
3. Data management
4. Data modeling

The major platform is R with the IDE (Integrated Development Environment) is Rstudio. However, the principles and applications can be used for other languages and platforms such as Python and Julia.

References