

# qtraining repository

Date of documentation update: 20161221

Documentation updated by: Veronica

## Introduction

**qtraining** is a private Github repository containing Quantide training courses material.

The courses included are:

- **R for Beginners** (two days course)
- **R for Beginners** (one day course)
- **Statistical Models with R**
- **Base R programming**
- **Data Manipulation with R**
- **Data Visualization with R** in english and in italian (under development)
- **Data Mining with R**
- **Big Data** (under development)

## Clone qtraining repository on your pc

**qtraining** repository is a private Quantide repository on Github, so to work on it you have to belong to Quantide team on Github.

If you want to work on **qtraining** repository, you have to clone it on your pc. Github projects must be developed on *dev* folder inside your “/home/user-name”. If you don’t have this folder you have to create it.

Follow these steps:

1. Open your browser
2. Go on <https://github.com/quantide/qtraining>
3. Click on “Clone or Download” green button
4. Choose “Clone with ssh” and copy the address (press “copy to clipboard” button)
5. Open a shell
6. Place yourself inside *dev* folder
7. Enter `git clone <address>`

A folder, named *qtraining* will be created into *dev* folder. It will include **qtraining** repository.

## qtraining Structure

**qtraining** repository contains both courses development material and courses manuals.

The structure is organized in the following folders:

- *00-qdata*: containing courses data organized as an R package, named **qdata**
- *010-rbase-two-days*: containing “R for Beginners” (two days) course
- *011-rbase-one-days*: containing “R for Beginners” (one day) course
- *020-models*: containing “Statistical Models with R” course
- *030-rprogramming-base*: containing “Base R programming” course
- *050-dplyr-datamanage*: containing “Data Manipulation with R” course

- *060-ggplot*: containing “Data Visualization with R” course in english
- *061-ggplot-ita*: containing “Data Visualization with R” course in italian (under development)
- *080-data-mining*: containing “Data Mining with R” course
- *100-bigdata-spark*: contains “Big Data” course
- *courses-index*: containing courses index
- *include*: containing files for output structure building

All previous folders includes courses material, apart from *00-qdata*, *courses-index* and *include* folders.

## An overview on courses material folders

Each course material folder name must be identified by a **number** and by the **course name**, in the following format: “number-course name”.

The **number** to be assigned to the course name must consist of three digits and the **course name** must describe the course name in short. If the **course name** consists in more than one word, the words must be separated by -.

The courses material folders have the same base structure:

1. *input* folder, which contains source files of course material
2. *output* folder, which contains built course material files
3. *exercises* folder, which contains source files of exercises
4. *docs* folder, which contains documents useful for course development
5. *Makefile*, which is a text file containing code written in Linux kernel. It is a simple way to organize code compilation. It contains the instructions for building the content of **output** folder
6. RStudio project file, named **course name.Rproj**

In some courses material folder is included also *data* folder, which contains data that can't be included in *qdata* package.

### input folder

It includes:

- *Markdown and R Markdown scripts with course content*. Course content is organized in chapters. Each chapter is identified by a script. It is recommended to add a number to the script names for ordering them (e.g. 01-first.Rmd). The course material is usually written in english.
- *images* folder, which contains figures to be included in the course content
- *TOC* file, which contains course index.

### TOC file

TOC files includes the structure of the course manual.

The md or Rmd files must be ordered, in column, from the first to the last.

Each file must be associated with a title. In particular, each row identifies a chapter: you have to write firstly the title, then “|” and finally the md and Rmd name related to that chapter. You can also divide the rmd into sections, writing only the title of the section before the belonging Rmds.

Here an example:

First Section

First Rmd | 1-first.Rmd

Second Rmd | 2-second.Rmd

## output folder

It includes:

- *html* folder, which contains htmls files with the full content of the course and a zip folder containing all html files (the folder name is “course name.zip”)
- *purl* folder, which contains R script with R code extracted from the Rmds (one R script for each Rmd), and a zip folder containing all R scripts (the folder name is “R.zip”)
- *pdf* folder, which contains pdf file/s of courses exercises (this folder is not always present)

## exercises folder

It includes:

- *Markdown and R Markdown scripts with course exercises.*
- *images* folder, which contains figures to be included in exercises content

This folder is not populated in all material courses folders. It is populated in: *010-rbase*, *020-models*, *050-dplyr-datamanage*.

In *010-rbase* and *050-dplyr-datamanage*, exercises are organized in chapters. Each chapter is identified by a script. The pdf file/s of courses exercises are built by the *Makefile* and included in *output/pdf* folder.

In *020-models* script exercises and built pdf and html files are included in *exercises* folder. They are not built by the *Makefile*, but manually clicking “Knit PDF” from RStudio toolbar, because the exercises are not yet completed.

## Guide for building course material

1. Open RStudio
2. Double-click on the project file
3. Click on “Build All” from RStudio “Build” tab or click “Ctrl+Shift+B” on the keyboard

The course building follows the instructions provided by *Makefile*. For more details see the *Makefile* of the course of interest.

## 00-qdata folder

This folder contains courses data organized as an R package, named **qdata**.

It is structured in the following way:

- *data* folder, which contains data in .RData format included in **qdata** package
- *R* folder, which contains two R script: **qdata.R** and **qdata-data.R** with package and data documentation (written with roxygen2)
- *pkgs* folder, which contains **qdata** package versions realized
- *doc* folder, which contains an R script **raw-data.R**. This script includes some commands of operations done on data before including them in package.
- *rowdata* folder, which contains original data files
- *DESCRIPTION* file, which contains library description
- *NAMESPACE* file, which contains informations about imported and exported functions (automatically created by library ‘build’)

- R project file, named `qdata.Rproj`

Once built the package, *man* folder is created. It contains documentation files. I recommend you to not add to commit and push this folder.

On Decembre 20 2016, the version of `qdata` is 0.27.

## Guide for building `qdata`

### Build `qdata`

`qdata` is an R package so its building works as well as any R package building.

1. Open RStudio
2. Double-click on the project file
3. Click “Build & Reload” on RStudio “Build” tab or click “Ctrl+Shift+B” on the keyboard

*Note:* When one ore more data file/s are added or when data documentation is modified, the package version MUST be updated. To update the package version modify *Version:* tag in *DESCRIPTION* file going forward of one digit (e.g. 027 -> 0.28)

### Create source package

When you edit a new version of `qdata`, you MUST build a Source package of the new version:

Follow these steps:

1. Move the folders: *pkgs* and *rowdata* from *00-qdata* folder into another location (cut and paste), in order to not include them in the source package
2. Open RStudio
3. Open “More” window on RStudio “Build” Tab
4. Click on “Build Source package”
5. Reinsert the folders: *pkgs* and *rowdata* into *00-qdata* folder
6. Move the file *qdata\_x.xx.tar.gz* from “~dev/qtraining/00-qdata” into “~dev/qtraining/00-qdata/pkgs”

## Guide for installing and loading `qdata`

1. Open RStudio
2. Install the package typing the following lines on the R console:

```
install.packages("~/dev/qtraining/00-qdata/pkgs/qdata_0.27.tar.gz", repo = NULL )
```

3. Loading `qdata` package on the workspace, typing:

```
require(qdata)
```

4. Loading data, included in `qdata` package, on the workspace. For example, if we want to load “bank” data we have to type:

```
data("bank")
```

Otherwise, you can install `qdata` from Rstudio “Packages” tab:

1. Open RStudio
2. Click on “Install” into Rstudio “Packages” tab
3. Set “Install from” field to “Package Archive (.tar.gz)”
4. Choose “qdata” for “Packages” field

## qdata locations

The materials included into *00-qdata* folder is included also in another github repository: <https://github.com/quantide/qdata>.

*qdata* is a public Quantide repository containing the data necessary for public courses available on Quantide website. So also this version has to be updated. When you update the version of **qdata** on *qtraining* repository you have to update also that in *qdata* repository.

*Advice to update the version:* in *qdata* repository, add the new .Rdata files into *data* folder, and replace the scripts included in *R* folder with that modified of *qtraining* version.

*Future Developments:* develop *qdata* package into a single location.

## include folder

This folder contains files with the instructions on output files structure building for each course. It means that the *Makefile* of each course refers to these files for building the structure of output files.

It is structured in these folders:

- *html*
- *libs*
- *r*
- *tex*

## courses-index folder

The content of this folder is used for building an html index of courses.

It includes:

- *course-index.html*: html file of courses index
- *images* folder: which includes figures to be used in *course-index.html* file

## Modify courses index

1. Open *course-index.html* with a text editor
2. Modify the file. The code is written in HTML
3. Save the file

## Github

You are working on a team, so you need to remain aligned with your colleagues.

In particular, you have to add your work to the repository and to remain aligned with the work of the others.

You can do these operations on the terminal or from “Git” tab on Rstudio. We will see in details both options.

## **Add new files to the repository**

Now we should add some new files. Create new files or copy your project files inside this folder. Then FOR EACH FILE we tell git to add them to the list of file to add or update to the repository. Usually, a Github repository includes only the source code, but in this case we will make an exception. Indeed, it is useful also to add also compiled files (html and purl) of a stable version of the course material.

### **First option: terminal**

1. Open the terminal, position inside repository folder and type this command:

```
$ git add example-file.R
```

2. Check the list of added file typing (not mandatory):

```
$ git status
```

3. Finally we add these files with:

```
$ git commit -m "comment of this version"
```

### **Second option: Rstudio "Git" tab**

Rstudio "Git" tab includes all files added or modified in the repository.

1. To add the one or more file/s, select it/them into Rstudio "Git" tab.
2. Then, click on "Commit". A Window will appear. That window has a space in top left where you have to write the "comment of this version".
3. Done this, click on "Commit".

## **Push changes on the remote repository (github.com)**

We want propagate changes to the remote repository.

### **First option: terminal**

Git retains the remote repository link so it is enough to type, always in the terminal:

```
$ git push
```

### **Second option: Rstudio "Git" tab**

Once you have done a commit, you can push it by click on green upward arrow on Rstudio "Git" tab.

## **Pull changes committed by others**

When your colleague commit and push its changes, you need to update your local files.

### **First option: terminal**

In order to update them, type on the terminal:

```
$ git pull
```

### **Second option: Rstudio "Git" tab**

In order to update them, click on light blue downward arrow on Rstudio "Git" tab.

## Note

Although you are working on a single project to build a course, Rstudio “Git” tab or `git status` on the terminal, shows the files to be added to commit and pushed of all folders of `qtraining` repository. The same when you do the pull, you will pull the committed changes of all folders of `qtraining` repository and not that of the single project (course folder) you are working to.

## Bookdown

`bookdown` is an R package, built on top of R Markdown, and inherits the simplicity of the Markdown syntax, as well as the possibility of multiple types of output formats (PDF/HTML/Word/...).

To use it, you need a version of RStudio higher than 1.0.0.

First of all you have to install and load `bookdown`.

To build a book with bookdown, you have to refer to a particular .Rproj file, `bookdown-demo.Rproj`, which is available on <https://github.com/rstudio/bookdown-demo>. I recommend you to download it as a Zip file and to use it as a structure for your book.

In `qtraining`, I used bookdown-demo into *manual-pdf* and *exercises* folders of `011-rbase-one-day`. These folders represents my first trial to use `bookdown`, so I advice you to look at the structure of that folders and to read the Bookdown web book.

## My idea of structure

1. Generate into bookdown-demo folder two subfolders, *input* and *output* and includes all the content of bookdown-demo into input.
2. Generate your Rmd containing the course content in *input* folder. Each Rmd represents a chapter of your book, so you have to put a number at the beginning of the Rmd name. The number have to correspond to the Rmd order in the book.
3. *input* folder must contain `index.Rmd` file, which will be the first chapter of your book.
4. Open `_bookdown.yml` file and set these paramters:
  - `book_filename`: name of the file name created
  - `chapter_name`: "Chapter ": name to be added at the beginning of each chapter
  - `new_session`: `yes`: set a new session for each rmd run
  - `output_dir`: `"../output"`, in this way the output will be created into *output* folder
5. Open `_output.yml` and set `bookdown::pdf_book`: if you want to create a pdf book, `bookdown::gitbook`: if you want to create a web book, ... . Here, you have to specify the related .css or .tex files. `preamble.tex` is a style .tex file that contains the instructions to create the cover of my book. You can start from here of you can create your own style .tex file.

## Build book

1. Open RStudio
2. Open `bookdown-demo.Rproj`
3. Click on “Build Book” on “Build” tab

Before building the book I advice you to click on “More” on “Build tab” and to choose “Clean All”, in order to delete all old results.