

GGPlot2 Essentials

Great Data Visualization in R

Alboukadel KASSAMBARA

Edition 1
Datanovia.com

GGPlot2 Essentials for Great Data Visualization in R

Alboukadel KASSAMBARA

Copyright ©2017 by Alboukadel Kassambara. All rights reserved.

Published by STHDA (<http://www.sthda.com>), Alboukadel Kassambara

Contact: Alboukadel Kassambara <alboukadel.kassambara@gmail.com>

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, without the prior written permission of the Publisher. Requests to the Publisher for permission should be addressed to STHDA (<http://www.sthda.com>).

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials.

Neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

For general information contact Alboukadel Kassambara <alboukadel.kassambara@gmail.com>.

Contents

0.1	What you will learn	vi
0.2	Key features of this book	vi
0.3	Book website	vii
0.4	Executing the R codes from the PDF	vii
0.5	Colophon	viii
About the author		ix
1	Introduction to R	1
1.1	Install R and RStudio	1
1.2	Install and load required R packages	2
1.3	Data format	2
1.4	Import your data in R	3
1.5	Demo data sets	3
1.6	Data manipulation	3
1.7	Close your R/RStudio session	4
2	Introduction to GGPlot2	5
2.1	What is ggplot2	5
2.2	Key functions	5
2.3	Example of plots	6
2.4	Legend position	8
2.5	Titles and axis labels	8
2.6	Facet: Plot with multiple pnels	9
2.7	GGPlot theme	10
2.8	Further customizations of a ggplot	11
2.9	Save ggplots	11
2.10	Conclusion	12
3	Scatter Plot	13
3.1	Introduction	13
3.2	Data preparation	13
3.3	Loading required R package	14
3.4	Basic scatter plots	14
3.5	Scatter plots with multiple groups	15
3.6	Add regression lines	16
3.7	Add marginal rugs to a scatter plot	17
3.8	Jitter points to reduce overplotting	18
3.9	Add point text labels	18
3.10	Bubble chart	20
3.11	Color by a continuous variable	21

4	Boxplot	22
4.1	Introduction	22
4.2	Key R functions	22
4.3	Data preparation	22
4.4	Loading required R package	23
4.5	Basic boxplots	23
4.6	Change boxplot colors by groups:	24
4.7	Create a boxplot with multiple groups	25
4.8	Multiple panel boxplots	25
4.9	Conclusion	26
5	Violin Plot	27
5.1	Introduction	27
5.2	Key R functions	27
5.3	Data preparation	27
5.4	Loading required R package	28
5.5	Basic violin plots	28
5.6	Create a Violin Plot with multiple groups	29
5.7	Conclusion	29
6	Dot Plot	30
6.1	Introduction	30
6.2	Key R functions	30
6.3	Data preparation	30
6.4	Loading required R package	31
6.5	Basic Dot Plots	31
6.6	Create a Dot Plot with multiple groups	32
6.7	Conclusion	32
7	Stripcharts	33
7.1	Introduction	33
7.2	Key R functions	33
7.3	Data preparation	33
7.4	Loading required R package	33
7.5	Basic stripcharts	34
7.6	Combine with box plots and violin plots	34
7.7	Create a stripchart with multiple groups	35
7.8	Conclusion	36
8	Line Plot	37
8.1	Introduction	37
8.2	Key R functions	37
8.3	Data preparation	37
8.4	Loading required R package	38
8.5	Basic line plots	38
8.6	Line plot with multiple groups	39
8.7	Line plot with a numeric x-axis	39
8.8	Line plot with dates on x-axis: Time series	40
8.9	Conclusion	42
9	Barplot	43
9.1	Key R functions	43

9.2	Data preparation	43
9.3	Loading required R package	44
9.4	Basic barplots	44
9.5	Change barplot colors by groups	45
9.6	Barplot with multiple groups	45
9.7	Conclusion	48
10	Error Bars	49
10.1	Introduction	49
10.2	Loading required R package	49
10.3	Data preparation	49
10.4	Key R functions and error plot types	50
10.5	Basic error bars	51
10.6	Grouped error bars	54
10.7	Conclusion	57
11	Density Plot	58
11.1	Key R functions	58
11.2	Data preparation	58
11.3	Loading required R package	59
11.4	Basic density plots	59
11.5	Change color by groups	60
12	Histogram Plot	61
12.1	Key R functions	61
12.2	Data preparation	61
12.3	Loading required R package	62
12.4	Basic histogram plots	62
12.5	Change color by groups	63
12.6	Combine histogram and density plots	64
12.7	Conclusion	65
13	QQPlot	66
13.1	Key R functions	66
13.2	Data preparation	66
13.3	Loading required R package	66
13.4	Create qqplots	67
13.5	Conclusion	68
14	ECDF Plot	69
14.1	Data preparation	69
14.2	Loading required R package	69
14.3	Create ECDF plots	69
14.4	Conclusion	70
15	Multiple GGPlots into a Figure	71
15.1	Introduction	71
15.2	Loading required R packages	71
15.3	Basic ggplot	71
15.4	Multiple panels figure using ggplot facet	72
15.5	Combine multiple ggplots using ggarrange()	74
15.6	Conclusion	78

Preface

0.1 What you will learn

GGPlot2 is a powerful and a flexible R package, implemented by Hadley Wickham, for producing elegant graphics piece by piece.

ggplot2 has become a popular package for data visualization. The official documentation of the package is available at: <https://ggplot2.tidyverse.org/reference/>. However, going through this comprehensive documentation can “drive you crazy”!

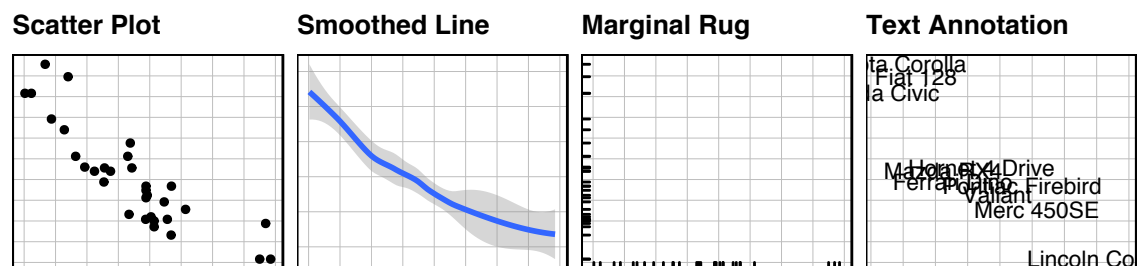
This book presents the essentials of ggplot2 to easily create beautiful graphics in R.

0.2 Key features of this book

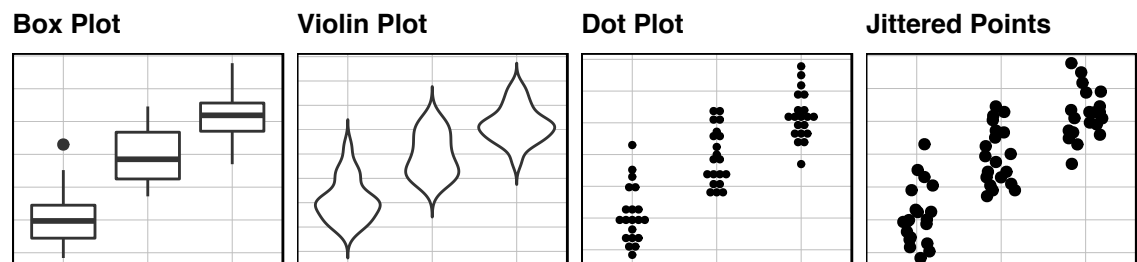
- Covers the most important graphic functions
- Short, self-contained chapters with practical examples.

Some examples of graphs, described in this book, are shown below.

- Create **Scatter plots** to display the relationship between two continuous variables x and y

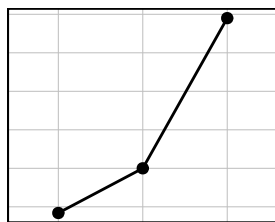


- Using Box plots and alternatives to visualize data grouped by the levels of a categorical variable

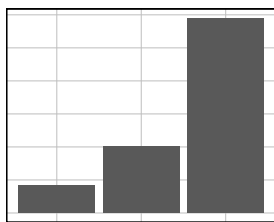


- Bar and Line Plots

Line Plot

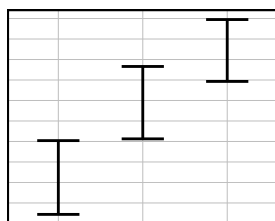


Bar Plot

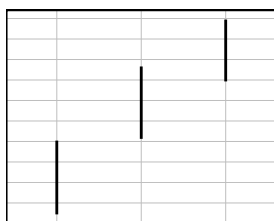


- Visualizing error bars

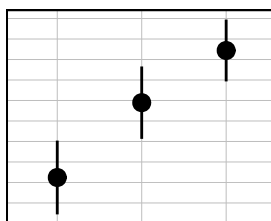
Error Bars



Line Range

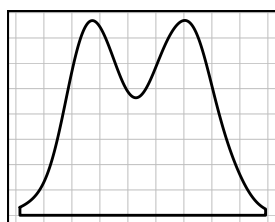


Point Range

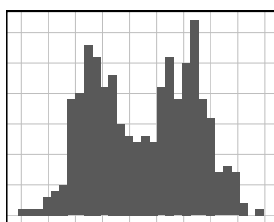


- Inspecting the distribution of a **continuous variable** using **density plots**, **histograms** and alternatives

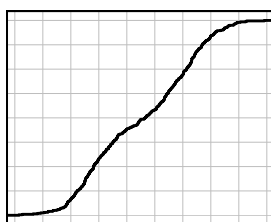
Density Plot



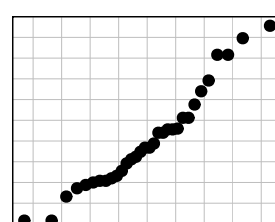
Histogram Plot



ECDF Plot



QQ Plot



You will also learn how to combine multiple ggplots into one figure.

0.3 Book website

The website for this book is located at : <https://www.datanovia.com/english/>. It contains number of resources.

0.4 Executing the R codes from the PDF

For a single line R code, you can just copy the code from the PDF to the R console.

For a multiple-line R codes, an error is generated, sometimes, when you copy and paste directly the R code from the PDF to the R console. If this happens, a solution is to:

- Paste firstly the code in your R code editor or in your text editor
- Copy the code from your text/code editor to the R console

0.5 Colophon

This book was built with R 3.3.2 and the following packages :

##	name	version	source
## 1	bookdown	0.7	CRAN
## 2	cowplot	0.9.2	CRAN
## 3	dplyr	0.7.7	CRAN
## 4	dplyr	0.7.7	CRAN
## 5	ggplot2	3.1.0.9000	Github:tidyverse/ggplot2
## 6	ggpubr	0.2	local:kassambara/ggpubr
## 7	ggrepel	0.8.0	CRAN
## 8	readr	1.3.0	CRAN

About the author

Alboukadel Kassambara is a PhD in Bioinformatics and Cancer Biology. He works since many years on genomic data analysis and visualization (read more: <http://www.alboukadel.com/>).

He has work experiences in statistical and computational methods to identify prognostic and predictive biomarker signatures through integrative analysis of large-scale genomic and clinical data sets.

He is the author of:

- 1) the bioinformatics tool named **GenomicScape** (www.genomicscape.com), an easy-to-use web tool for gene expression data analysis and visualization.
- 2) the **Datanovia** (<https://www.datanovia.com/en/>) and **STHDA** (<http://www.sthda.com/english/>) websites, which contains many courses and **tutorials** on data data mining and statistics for decision supports.
- 3) many popular **R packages** for multivariate data analysis, survival analysis, correlation matrix visualization and basic data visualization (<https://rpkgs.datanovia.com/>).
- 4) many **books** on data analysis, visualization and machine learning (<https://www.datanovia.com/en/shop/>)

Chapter 1

Introduction to R

R is a free and powerful statistical software for **analyzing** and **visualizing** data. If you want to learn easily the essential of R programming, visit our series of tutorials available on STHDA: <http://www.sthda.com/english/wiki/r-basics-quick-and-easy>.

In this chapter, we provide a very brief introduction to **R**, for installing R/RStudio as well as importing your data into R and installing required libraries.

1.1 Install R and RStudio

R and RStudio can be installed on Windows, MAC OSX and Linux platforms. RStudio is an integrated development environment for R that makes using R easier. It includes a console, code editor and tools for plotting.

1. R can be downloaded and installed from the Comprehensive R Archive Network (CRAN) webpage (<http://cran.r-project.org/>)
2. After installing R software, install also the RStudio software available at: <http://www.rstudio.com/products/RStudio/>.
3. Launch RStudio and start use R inside R studio.

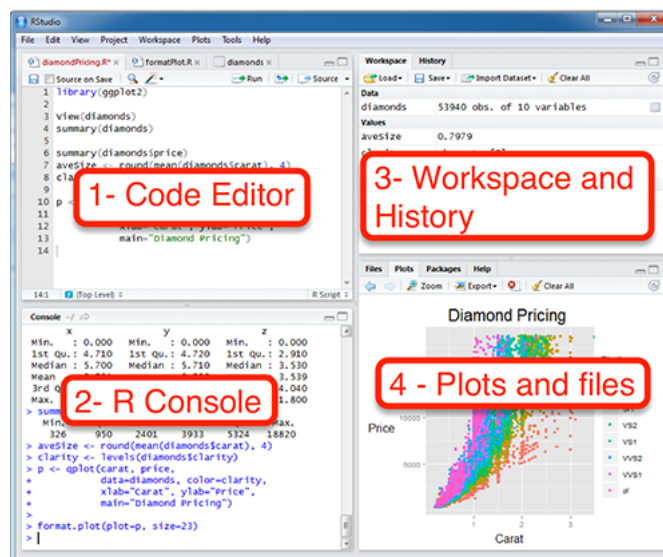


Figure 1.1: Rstudio interface

1.2 Install and load required R packages

An R package is a collection of functionalities that extends the capabilities of base R. For example, to use the R code provide in this book, you should install the following R packages:

- **tidyverse** packages, which are a collection of R packages that share the same programming philosophy. These packages include:
 - `readr`: for importing data into R
 - `dplyr`: for data manipulation
 - `ggplot2`: for data visualization.
- `ggpubr` package, which makes it easy, for beginner, to create publication ready plots.

1. **Install the tidyverse package.** Installing tidyverse will install automatically `readr`, `dplyr`, `ggplot2` and more. Type the following code in the R console:

```
install.packages("tidyverse")
```

2. **Install the ggpubr package.**

```
install.packages("ggpubr")
```

3. **Load required packages.** After installation, you must first load the package for using the functions in the package. The function `library()` is used for this task. An alternative function is `require()`. For example, to load `ggplot2` and `ggpubr` packages, type this:

```
library("ggplot2")
library("ggpubr")
```

Now, we can use R functions, such as `ggscatter()` [in the `ggpubr` package] for creating a scatter plot.

If you want a help about a given function, say `ggscatter()`, type this in R console: `?ggscatter`.

1.3 Data format

Your data should be in rectangular format, where columns are variables and rows are observations (individuals or samples).

- Column names should be compatible with R naming conventions. Avoid column with blank space and special characters. Good column names: `long_jump` or `long.jump`. Bad column name: `long jump`.
- Avoid beginning column names with a number. Use letter instead. Good column names: `sport_100m` or `x100m`. Bad column name: `100m`.
- Replace missing values by `NA` (for not available)

For example, your data should look like this:

	manufacturer	model	displ	year	cyl	trans	drv
1	audi	a4	1.8	1999	4	auto(l5)	f
2	audi	a4	1.8	1999	4	manual(m5)	f
3	audi	a4	2.0	2008	4	manual(m6)	f
4	audi	a4	2.0	2008	4	auto(av)	f

Read more at: [Best Practices in Preparing Data Files for Importing into R¹](#)

1.4 Import your data in R

First, save your data into txt or csv file formats and import it as follow (you will be asked to choose the file):

```
library("readr")

# Reads tab delimited files (.txt tab)
my_data <- read_tsv(file.choose())

# Reads comma (,) delimited files (.csv)
my_data <- read_csv(file.choose())

# Reads semicolon(;) separated files(.csv)
my_data <- read_csv2(file.choose())
```

Read more about how to import data into R at this link: <http://www.sthda.com/english/wiki/importing-data-into-r>

1.5 Demo data sets

R comes with several demo data sets for playing with R functions. The most used R demo data sets include: **USArrests**, **iris** and **mtcars**. To load a demo data set, use the function **data()** as follow. The function **head()** is used to inspect the data.

```
data("iris") # Loading
head(iris, n = 3) # Print the first n = 3 rows
```

##	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
## 1	5.1	3.5	1.4	0.2	setosa
## 2	4.9	3.0	1.4	0.2	setosa
## 3	4.7	3.2	1.3	0.2	setosa

To learn more about iris data sets, type this:

```
?iris
```

After typing the above R code, you will see the description of **iris** data set: this iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica.

1.6 Data manipulation

After importing your data in R, you can easily manipulate it using the **dplyr** package (?), which can be installed using the R code: `install.packages("dplyr")`.

¹<http://www.sthda.com/english/wiki/best-practices-in-preparing-data-files-for-importing-into-r>

After loading `dplyr`, you can use the following R functions:

- `filter()`: Pick rows (observations/samples) based on their values.
- `distinct()`: Remove duplicate rows.
- `arrange()`: Reorder the rows.
- `select()`: Select columns (variables) by their names.
- `rename()`: Rename columns.
- `mutate()`: Add/create new variables.
- `summarise()`: Compute statistical summaries (e.g., computing the mean or the sum)
- `group_by()`: Operate on subsets of the data set.

Note that, `dplyr` package allows to use the forward-pipe chaining operator (`%>%`) for combining multiple operations. For example, `x %>% f` is equivalent to `f(x)`. Using the pipe (`%>%`), the output of each operation is passed to the next operation. This makes R programming easy.

Read more about Data Manipulation at this link: <https://www.datanovia.com/en/courses/data-manipulation-in-r/>

1.7 Close your R/RStudio session

Each time you close R/RStudio, you will be asked whether you want to save the data from your R session. If you decide to save, the data will be available in future R sessions.

Chapter 2

Introduction to GGPlot2

2.1 What is ggplot2

GGPlot2 is a powerful and a flexible R package, implemented by Hadley Wickham, for producing elegant graphics piece by piece (Wickham et al., 2017).

The **gg** in ggplot2 means *Grammar of Graphics*, a graphic concept which describes plots by using a “grammar”. According to the ggplot2 concept, a plot can be divided into different fundamental parts: **Plot = data + Aesthetics + Geometry**

- **data**: a data frame
- **aesthetics**: used to indicate the **x** and **y** variables. It can be also used to control the **color**, the **size** and the **shape** of points, etc....
- **geometry**: corresponds to the type of graphics (histogram, box plot, line plot, ...)

The ggplot2 syntax might seem opaque for beginners, but once you understand the basics, you can create and customize any kind of plots you want.

Note that, to reduce this opacity, we recently created an R package, named **ggpubr** (ggplot2 Based Publication Ready Plots), for making ggplot simpler for students and researchers with non-advanced programming backgrounds.

2.2 Key functions

After installing and loading the ggplot2 package, you can use the following key functions:

Plot types	GGPlot2 functions
Initialize a ggplot	ggplot()
Scatter plot	geom_point()
Box plot	geom_boxplot()
Violin plot	geom_violin()
strip chart	geom_jitter()
Dot plot	geom_dotplot()
Bar chart	geom_bar() or geom_col()
Line plot	geom_line()
Histogram	geom_histogram()
Density plot	geom_density()

Plot types	GGPlot2 functions
Error bars	<code>geom_errorbar()</code>
QQ plot	<code>stat_qq()</code>
ECDF plot	<code>stat_ecdf()</code>
Title and axis labels	<code>labs()</code>

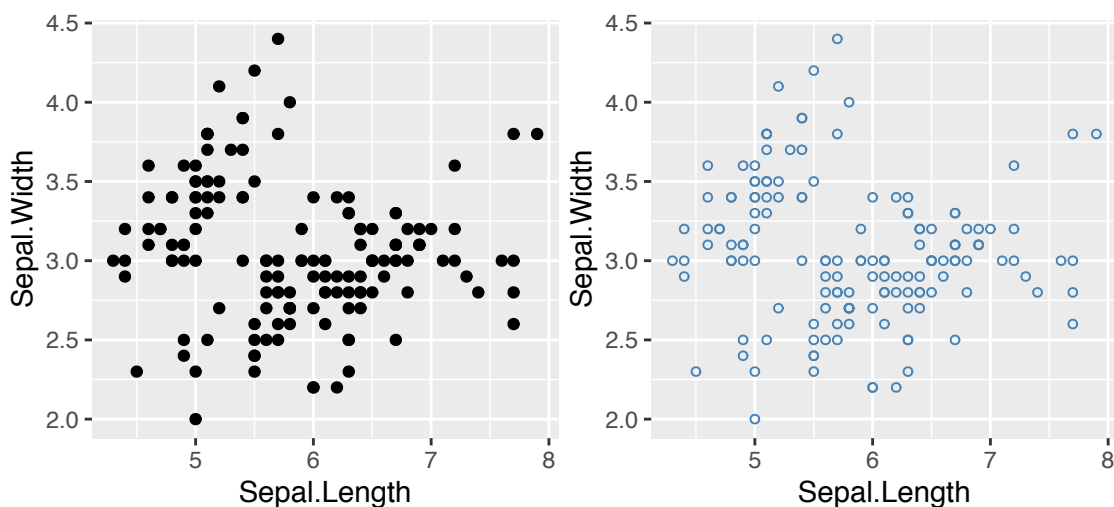
2.3 Example of plots

The main function in the `ggplot2` package is `ggplot()`, which can be used to initialize the plotting system with data and x/y variables.

For example, the following R code takes the `iris` data set to initialize the `ggplot` and then a layer (`geom_point()`) is added onto the `ggplot` to create a scatter plot of `x = Sepal.Length` by `y = Sepal.Width`:

```
library(ggplot2)
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width))+
  geom_point()

# Change point size, color and shape
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width))+
  geom_point(size = 1.2, color = "steelblue", shape = 21)
```



Note that, in the code above, the shape of points is specified as number. The different point shape available in R, include: