R for life sciences. Chapter 6, ggplot: A new way of thinking about graphics

2020-09-13

[**R for life sciences. Chapter 6, ggplot: A new way of thinking about graphics**](#_8e5vk97g8al1) **1**

[Alfonso Garmendia](#_1lu31vo84s33) 2

[2020-09-13](#_yvgrzj7oren4) 2

[Table of contents](#_saxo8h7fyhys) 2

[**A new way of thinking about graphics**](#_v333s8fqwc2i) **2**

[What is ggplot](#_vtlc3p3eb7e9) 3

[Installation](#_jyel4zk9by62) 3

[Cheatsheet](#_aveh7mw1rvrc) 3

[Usage](#_ne4p9sewud28) 3

[Other layers of information](#_vfhg40pqvdl7) 7

[To save your plots](#_6vzo9nagez5m) 8

[More features](#_d9ksesu3xilx) 8

[References](#_w03lapmx3w9w) 9

[**Exercises**](#_yvhphl4jyldz) **9**

[**About this tutorial**](#_qi82hhs9ht65) **10**

# R for life sciences. Chapter 6, ggplot: A new way of thinking about graphics

#### *Alfonso Garmendia*

Instituto Agroforestal Mediterráneo. Departamento de Ecosistemas Agroforestales. Universitat Politècnica de València.  
[algarsal@upv.es](mailto:algarsal@upv.es) <https://garmendia.blogs.upv.es/>

#### *2020-09-13*

## Table of contents

* [A new way of thinking about graphics](#2et92p0)
  + [What is ggplot](#tyjcwt)
  + [Installation](#3dy6vkm)
  + [Cheatsheet](#1t3h5sf)
  + [Usage](#4d34og8)
  + [Other layers of information](#2s8eyo1)
  + [To save your plots](#17dp8vu)
  + [More features](#3rdcrjn)
  + [References](#26in1rg)
* [Exercises](#lnxbz9)
* [About this tutorial](#35nkun2)

# A new way of thinking about graphics

## What is ggplot

ggplot2 is a system for declaratively creating graphics, based on The [Grammar of Graphics](https://codewords.recurse.com/issues/six/telling-stories-with-data-using-the-grammar-of-graphics).

You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

## Installation

It is in cran, so it is easy:

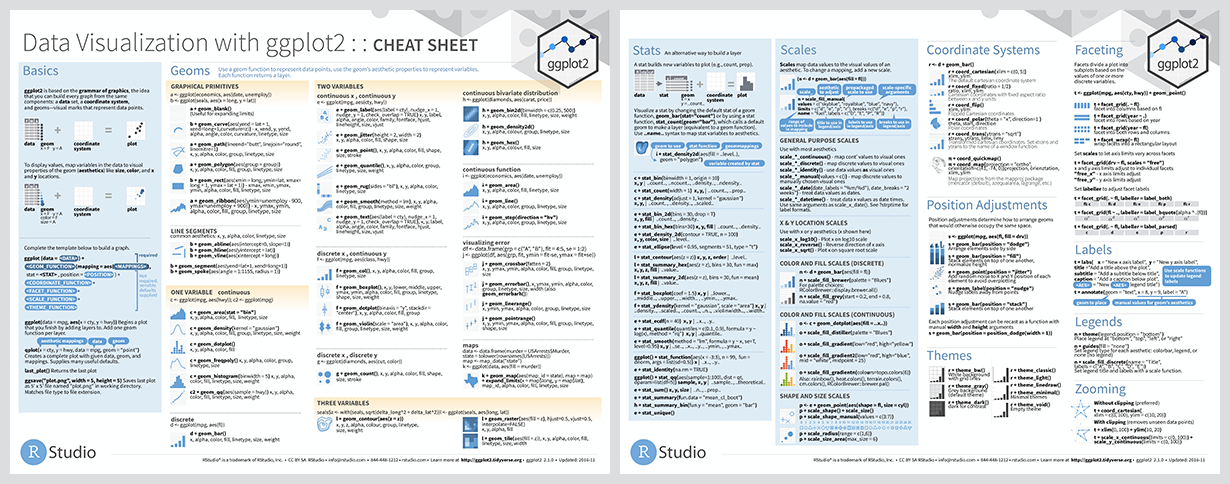
### Install ggplot2 package --------------------  
install.packages("ggplot2", dependencies = TRUE)

Remember than in your script the install.packages() command should allways be commented and at the beginning of the script.

### Packages  
*# install.packages("ggplot2", dependencies = TRUE)*  
  
library(ggplot2)

## Cheatsheet

There is an official cheatsheet here: <https://github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.pdf>

[](https://github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.pdf)

## Usage

All ggplot2 plots begin with a call to ggplot(), supplying default data and aesthethic mappings, specified by aes(). You then add layers, scales, coords and facets with **+**. To save a plot to disk, use ggsave().

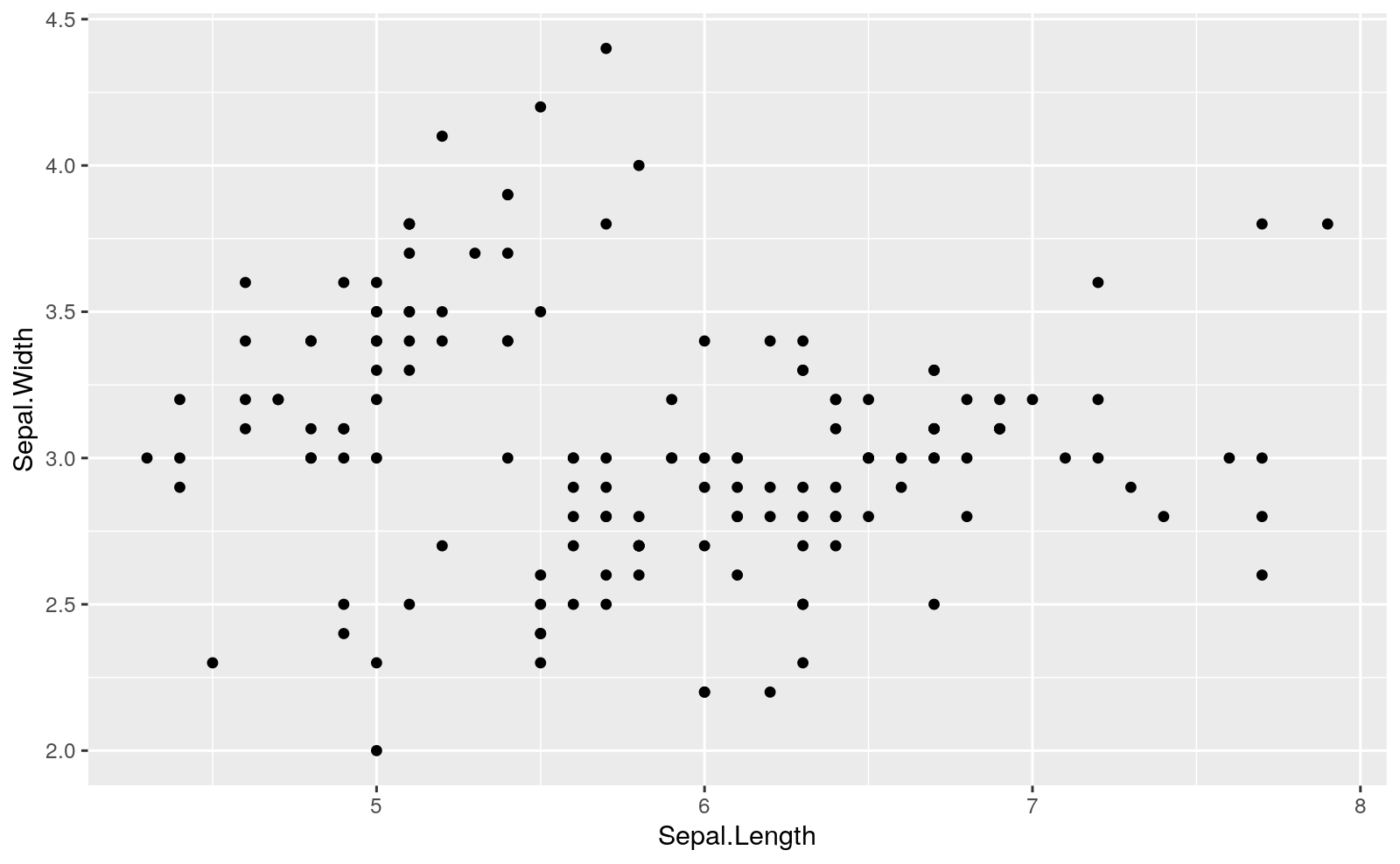
For plotting a ggplot2 plot it is necessary to have the data into a dataframe in long format.

**Producing a plot with ggplot2, we must give three things**:

1. A data frame containing our data.
2. How the columns of the data frame can be translated into positions, colors, sizes, and shapes of graphical elements (“aesthetics”).
3. The actual graphical elements to display (“geometric objects”).

Let’s make our first ggplot.

### Plot relationship between sepal length and width ----------  
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +  
 geom\_point()



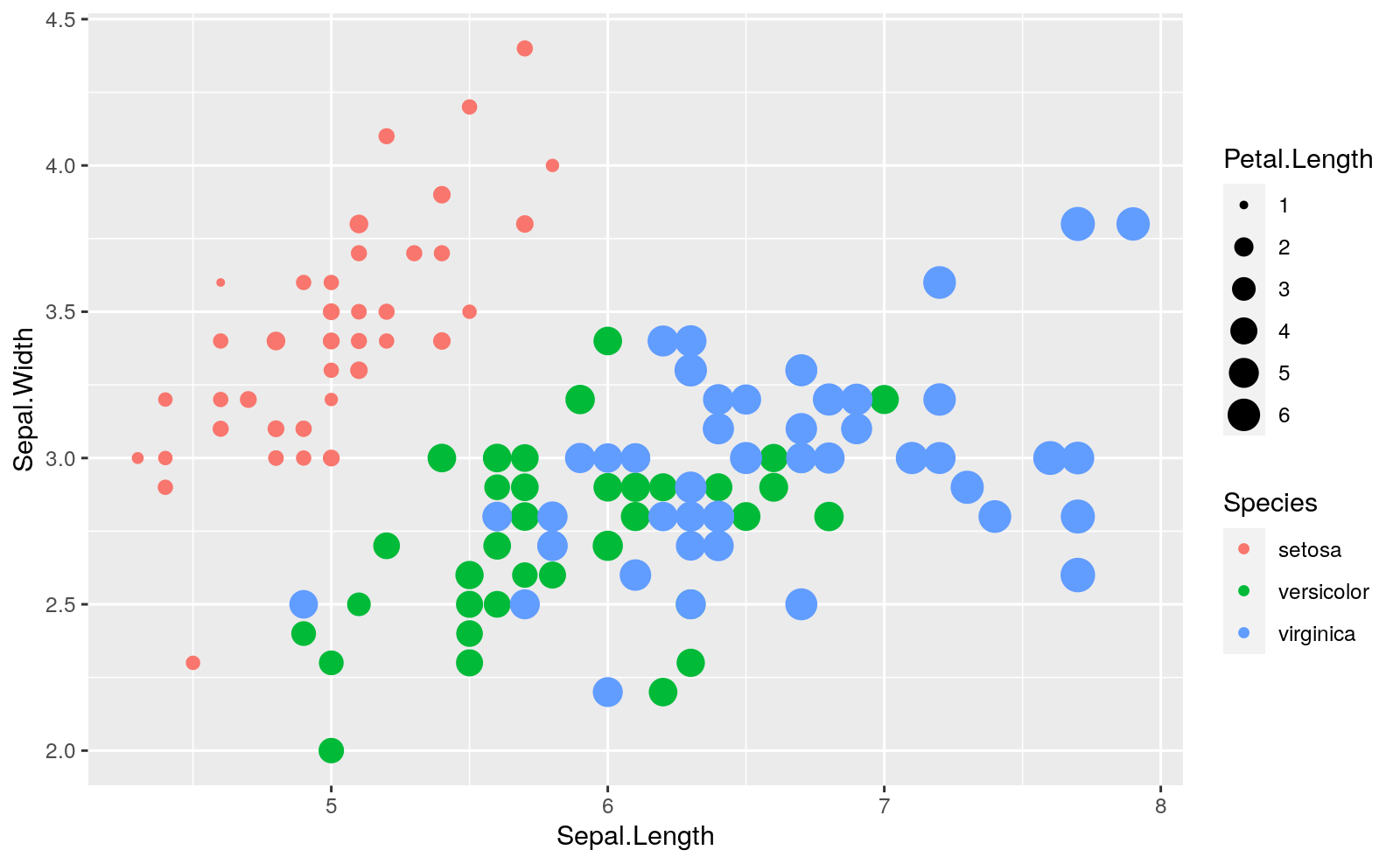
The call to ggplot() and aes() sets up the basics of the data and variables forthe data frame.

aes() defines the “aesthetics”, which is how columns of the data frame map to graphical attributes such as x and y position, color, size, etc. Arguments to aes() may refer to columns of the data frame directly.

We then literally add layers of graphics (“geoms”) to this.

Further aesthetics can be used. Any aesthetic can be either numeric or categorical, an appropriate scale will be used.

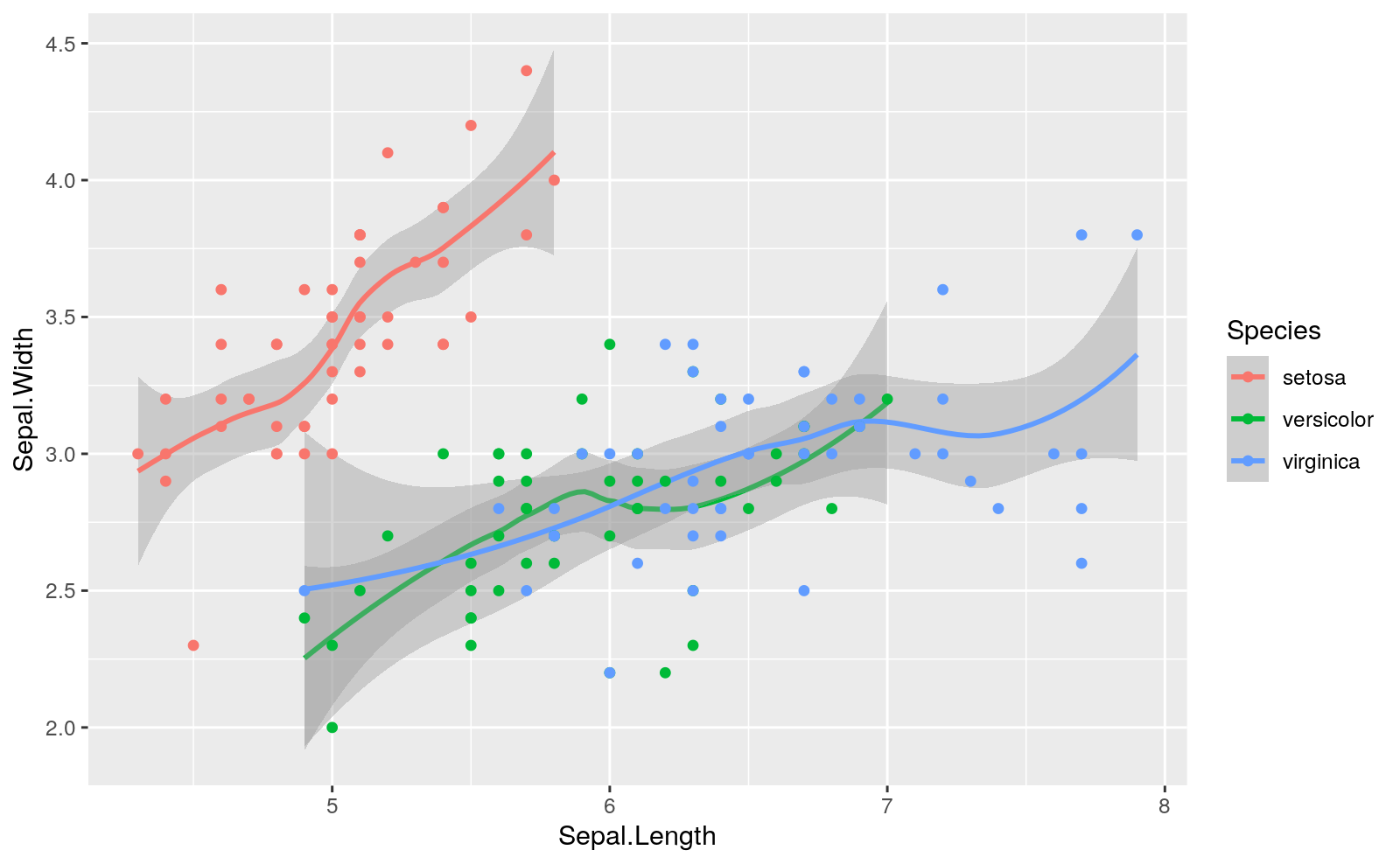
### Plot relationship between sepal length and width  
### related to Species and Petal.Length -----------------------  
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width,   
 color = Species, size = Petal.Length)) +  
 geom\_point()



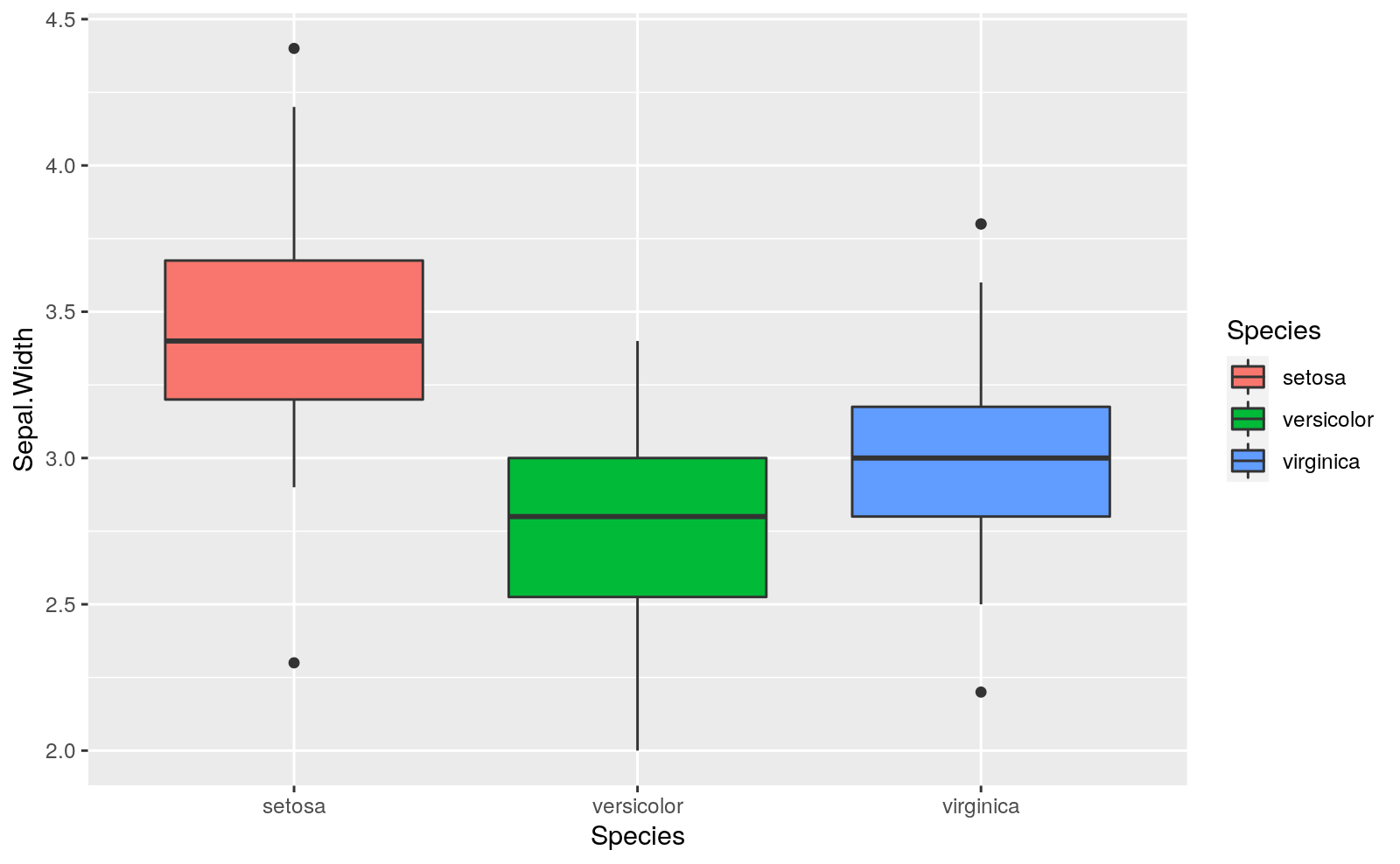
We can also use different geometries.

### Plot relationship between sepal length and width  
### related to Species ----------------------------  
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width,   
 color = Species)) +  
 geom\_smooth() +  
 geom\_point()

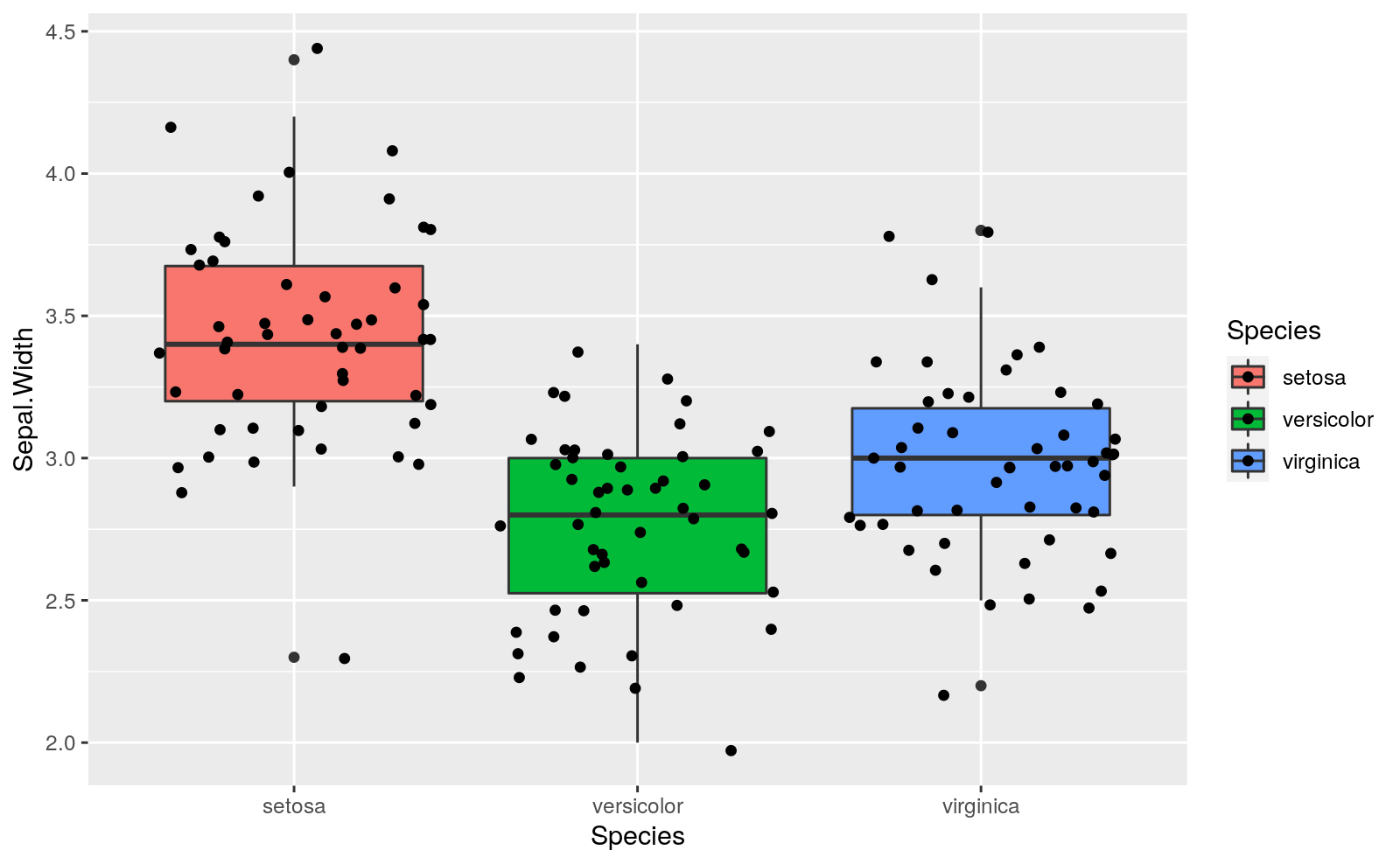
## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



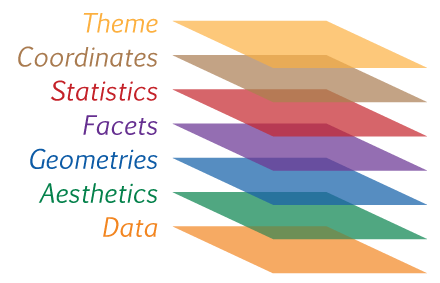
### boxplot of sepal width for each species ----------------------  
g <- ggplot(iris, aes(x = Species, y = Sepal.Width,   
 fill = Species))  
  
g + geom\_boxplot()



g + geom\_boxplot() +  
 geom\_jitter()



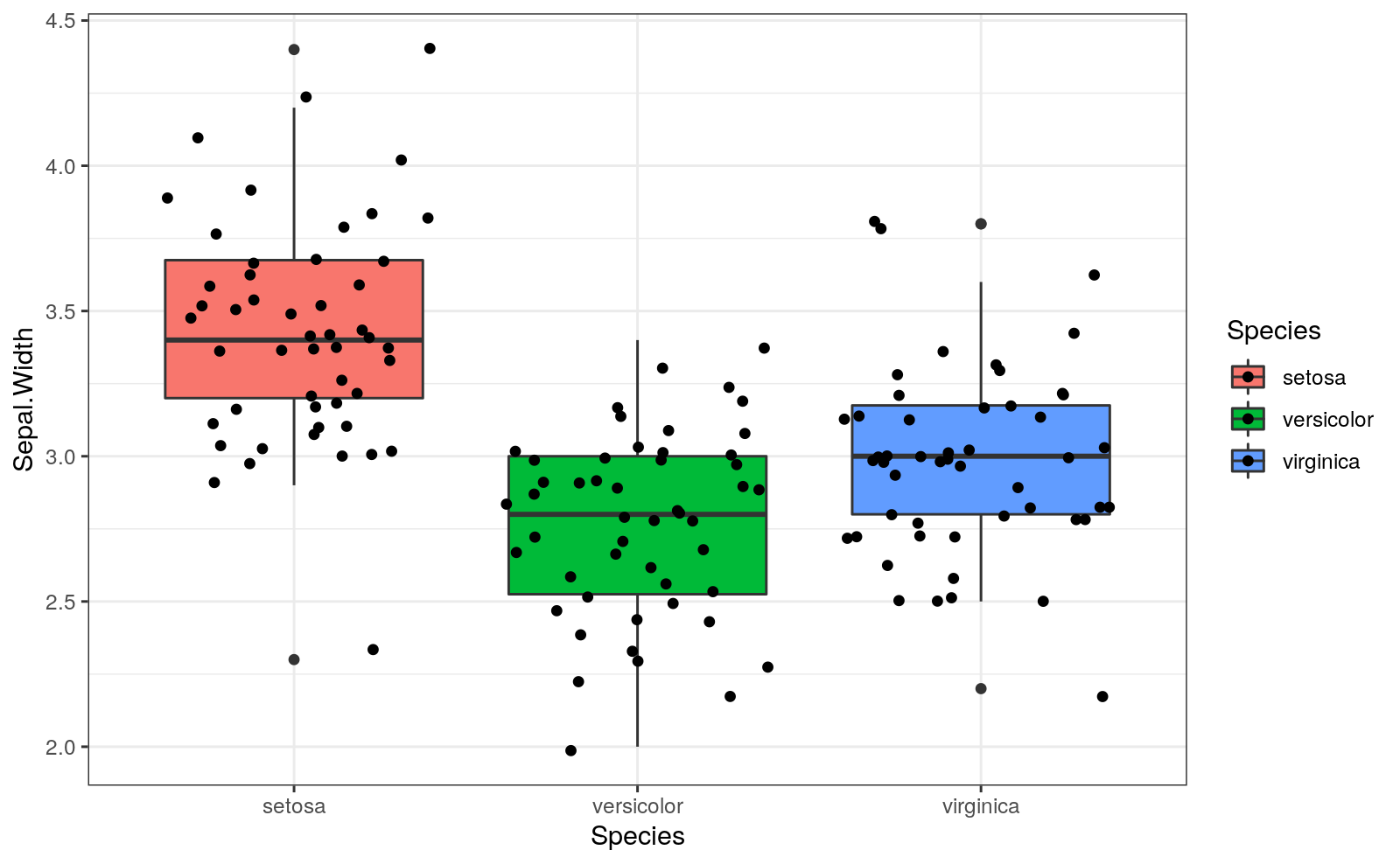
## Other layers of information



Different layers of information in ggplot

Besides data, aesthetics and geometries you can add different layers of information. For example, the previous plot can be draw with a different theme.

### boxplot of sepal width for each species --------------------  
g +  
 geom\_boxplot() +  
 geom\_jitter() +  
 theme\_bw()



## To save your plots

To save a particular plot from a R script you can use the command [ggsave()](https://ggplot2.tidyverse.org/reference/ggsave.html).

But for a RMarkdown document it is better if in the first setup R chunk you include a fig.path parameter with the direction where to save all the plots from the document.

knitr::opts\_chunk$set(fig.path = 'results/Fig\_')

## More features

The best way to see more features is to look for them in google when needed. Some interesting ones might be:

* The ggplot2 reference web, with all the posibilities. Please, click on the different posibilities to get to the help page with examples. <https://ggplot2.tidyverse.org/reference/>
* The R Graph Gallery is a very complete gallery of basic graphs. Each one of them is explained and with code examples. <https://www.r-graph-gallery.com/>
* Statistical tools for high-throughput data analysis (STHDA). A web with good introduccions to different graph types in R. <http://www.sthda.com/english/wiki/data-visualization>
* Other packages for particular plots. <http://www.ggplot2-exts.org/gallery/>
* The book “[R for Data Science](https://r4ds.had.co.nz/)” chapters for:
  1. data visualization. <https://r4ds.had.co.nz/data-visualisation.html>
  2. graphics for comunication. <https://r4ds.had.co.nz/graphics-for-communication.html>
* The web “*From Data to Viz*” <https://www.data-to-viz.com/> is a very good place to find the right graph for each type of data. Do not miss the **Story** link for each data type. It has also a collection of caveats to have in mind to avoid the most common mistakes when choosing the right graph for representign your data <https://www.data-to-viz.com/caveats.html>.

## References

* <https://ggplot2.tidyverse.org/>
* <https://ggplot2.tidyverse.org/reference/>
* <http://bloggotype.blogspot.com/2016/08/holiday-notes2-grammar-of-graphics.html>
* <https://monashdatafluency.github.io/r-intro-2/plotting-with-ggplot2.html>

# Exercises

In a new Rmarkdown document, copy and answer the following questions:

1. create a ggplot with the trees data using

* Girth as x
* Height as y
* Volume as size and color

1. Try different geometries

* geom\_point()
* geom\_line()
* geom\_smooth()
* geom\_smooth(method = "lm")
* Other geometries from <https://ggplot2.tidyverse.org/reference/>

¿Which is the best combination to represent the data?

1. Look at the R Graph Gallery <https://www.r-graph-gallery.com/> and choose a fancy plot you like. Copy paste the code from the example and do some modifications. Change the data, the theme or any other part of it.

# About this tutorial

Cite as: Alfonso Garmendia (2020) R for life sciences. Chapter 6, ggplot: A new way of thinking about graphics. <http://personales.upv.es/algarsal/R-tutorials/06_Tutorial-6_R-ggplot.html>

Available also in other formats (pdf, docx, …): <https://drive.google.com/drive/folders/19w914WCg8BVTVBE_zpgShmg2vpjguV1e?usp=sharing>.

Other simmilar tutorials: <https://garmendia.blogs.upv.es/r-lecture-notes/>

Originals are in bitbucket repository: <https://bitbucket.org/alfonsogar/tea_daa_tutorials>.

Document written in [Rmarkdown](http://rmarkdown.rstudio.com/), using [Rstudio](https://www.rstudio.com/).

* System: R version 4.0.2 (2020-06-22) x86\_64-pc-linux-gnu (64-bit) Ubuntu 18.04.5 LTS.
* Base packages: stats 4.0.2, graphics 4.0.2, grDevices 4.0.2, utils 4.0.2, datasets 4.0.2, methods 4.0.2, base 4.0.2.
* Other loaded packages: tidyr 1.1.0, dplyr 1.0.0, ggplot2 3.3.2, knitr 1.29, rmarkdown 2.3, googledrive 1.0.1.



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](http://creativecommons.org/licenses/by-nc-sa/4.0/).