**- COURSE 07 -**

**- MODULE 4 : WORKING WITH DATA IN R -**

The R programming language was designed to work with data at all stages of the data analysis process. In this part of the course, you’ll examine how R can help you structure, organize, and clean your data using functions and other processes. You’ll learn about data frames and how to work with them in R. You’ll also revisit the issue of data bias and how R can help.

### Learning Objectives

* Discuss how R functions may be used to address issues of bias and relationship between data variables
* Describe R functions that may be used to clean and organize data
* Describe functions used to work with data frames including read\_csv(), data(), and datapasta()
* Discuss the difference between tibbles and tribbles
* Compare and contrast data cleaning with different tools
* Create and work with data in R

CREATE DATA VISUALIZATIONS IN R

[Visualizations in R](https://www.coursera.org/learn/data-analysis-r/lecture/sbUO0/visualization-basics-in-r-and-tidyverse)

I've really enjoyed introducing you to R. By now, you know your way around R and R studio and you're familiar with some basic programming concepts. You've also learned how to clean and organize data in R. You've taken on the challenge of learning programming and developed new skills for Data Analysis. Great job. Now you'll learn how to use code to visualize your data. Data visualization is one of the most important parts of data analysis. Powerful visuals show stakeholders what your data means in a clear and compelling way and highlighting key insights. Visuals helped bring the story of your data to life and make that story easier to understand. You might remember the sneak peak I gave you of R's data visualization powers. I created those visuals with ggplot2, one of the core packages of the tidy verse. ggplot2 is R's most popular visualization package and for good reason, it's a powerful and user friendly, data viz tool. Up next, you'll learn how to write and execute all the code we previewed earlier. You'll learn how to use ggplot2 to create a variety of plots, organize and represent different variables in your data set and customize the look and feel of your visuals. Working with ggplot2 can help you get the most out of your data. Your new data viz skills will also make it easier to learn other parts of R. Going forward, you'll be better able to visualize the results of any change you make to your data. Plus you get an immediate result for all your hard work. Which is one of my favorite parts of creating plots in ggplot2, just enter some code, run it, and out comes a cool looking visual that helps you and others understand your data. Visualization is a key part of a data analyst's workflow. R lets you move back and forth between analysis and visualization quickly and easily. I'm looking forward to showing you what ggplot2 can do.

[Visualization basics in R and tidyverse](https://www.coursera.org/learn/data-analysis-r/lecture/sbUO0/visualization-basics-in-r-and-tidyverse)

Hi again, in this video we'll focus on ggplot2. We'll learn about its main features and functions and how it can help you visualize your data. First, let's talk about some different visualization packages you can use with R.

Base R has its own package and there are other useful packages you can add. They'll help you do almost anything you want with your data from making simple pie charts, to creating more complex visuals like interactive graphs and maps.

General-purpose packages like **Plotly** let you do a wide range of visualization functions. Others like RGL, focus on specific solutions like 3D visuals.

Some of the most popular include ggplot2, Plotly, Lattice, RGL, Dygraphs, Leaflet, Highcharter, Patchwork, gganimate and ggridges.

Personally, ggplot2 is my favorite for data analysis. It's both powerful and flexible. With a little bit of code, you can create all kinds of different plots. You can use ggplot2 on its own or extend its powers with other packages. Plus it's the most popular visualization package in R. A lot of data analysts prefer to use ggplot2 which is why we're using ggplot2 here.

Ggplot2 was originally created by the statistician and developer Hadley Wickham in 2005. Wickham's inspiration for creating ggplot2 came from the **1999 book The Grammar of Graphics**, a scholarly study of data visualization by computer scientist Leland Wilkinson.

The first two letters of ggplot2 actually stand for grammar of graphics. And in the same way the grammar of a human language gives us rules to build any kind of sentence, the grammar of graphics gives us rules to build any kind of visual. So ggplot2 has some basic building blocks that you can use to create plots.

In other words, when you learn the basic steps for creating a plot in ggplot2, you can reuse these steps to create lots of different kinds of plots. Plus, you can add or remove layers of detail to your plot without changing its basic structure or the underlying data. This makes ggplot2 really powerful.

Ggplot2 has lots of other benefits too.

You can create all different types of plots including scatter plots, bar charts, line diagrams and tons more.

You can change the colors, layout and dimensions of your plots and add text elements like titles, captions and labels. With just a little bit of code you can create high-quality visuals.

Plus ggplot2 lets you combine data manipulation and visualization using the pipe operator.

Ggplot2 also has tons of functions that cover all your data viz needs. To give you an idea, check out the ggplot2 cheat sheet, which is a popular reference guide. It's not important to learn all these functions right away or even know what they are. Over time as you get into more advanced data analysis, you can learn about new functions as you need them.

Just know that if you need to find a function for something, ggplot2 probably has it. We'll focus on some core concepts in ggplot2: aesthetics, geoms, facets, labels and annotations. These might be new concepts to you and that's okay.

In ggplot2 an aesthetic is a visual property of an object in your plot. For example, in a scatter plot aesthetics include things like the size, shape or color of your data points. Think of an aesthetic as a connection or mapping between a visual feature in your plot and a variable in your data.

A geom refers to the geometric object used to represent your data. For example, you can use points to create a scatter plot, bars to create a bar chart, or lines to create a line diagram. You can choose a geom to fit the type of data you have. Points show the relationship between two quantitative variables. Bars show one quantitative variable varies across different categories.

Facets let you display smaller groups or subsets of your data. With facets, you can create separate plots for all the variables in your dataset.

Finally, the label and annotate functions let you customize your plot. You can add text like titles, subtitles and captions to communicate the purpose of your plot or highlight important data.

[Hands-On Activity: Visualizing data with ggplot2](https://www.coursera.org/learn/data-analysis-r/quiz/fmyHH/hands-on-activity-visualizing-data-with-ggplot2)



## **Activity overview**

****

Earlier in this course, you encountered ggplot2, an R package for data visualization. In this activity, you’ll learn about the basic logic of data visualization in ggplot2 and how to create a plot using R code.

By the time you complete this activity, you’ll be able to write R functions that create data visualizations. This will enable you to create basic visualizations to demonstrate and share findings with your data and code.

## **The basics of ggplot2**

****

The ggplot2 package lets you make high quality, customizable plots of your data. As a refresher, ggplot2 is based on the grammar of graphics, which is a system for describing and building data visualizations. The essential idea behind the grammar of graphics is that you can build any plot from the same basic components, like building blocks.

These building blocks include:

* A dataset
* A set of geoms: A geom refers to the geometric object used to represent your data. For example, you can use points to create a scatter plot, bars to create a bar chart, lines to create a line diagram, etc.
* A set of aesthetic attributes: An aesthetic is a visual property of an object in your plot. You can think of an aesthetic as a connection, or mapping, between a visual feature in your plot and a variable in your data. For example, in a scatterplot, aesthetics include things like the size, shape, color, or location (x-axis, y-axis) of your data points.

To create a plot with ggplot2, you first choose a dataset. Then, you determine how to visually organize your data on a coordinate system by choosing a geom to represent your data points and aesthetics to map your variables.

## **Prepare your data**

****

The ggplot2 package lets you use R code to specify the dataset, geom, and aesthetics of your plot.

To do this, first choose a dataset to work with. For this activity, you will use the Palmer Penguins data that you’re already familiar with from earlier videos. However, you can also use another dataset instead.

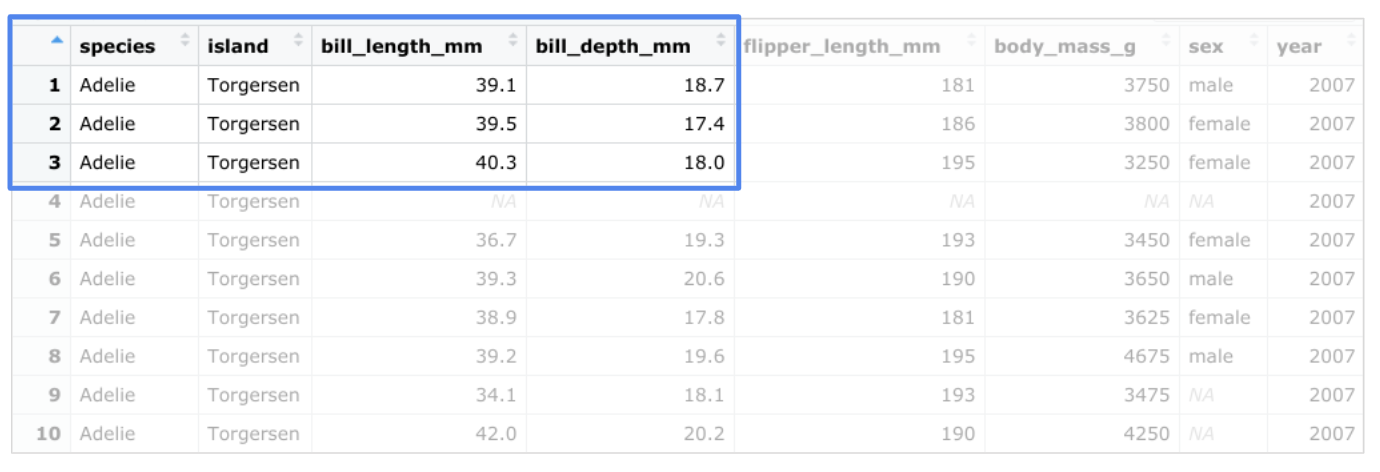
Once you decide on your dataset, open RStudio and follow these steps:

1. If you have not done so before, use the install.packages() function to install both ggplot2 and the Palmer Penguins data set. Type install.packages(“ggplot2”) and install.packages(“palmerpenguins”), then click Run.

2. Load ggplot2 and the dataset using the library() function. Type library(ggplot2) and library(palmerpenguins).

3. Now, examine the data frame for the penguins data. To do this, use the data() and View() functions. Use a capital “V” for the View() function since functions in R are case sensitive. Type data(penguins) and View(penguins), then click Run.

The first 10 rows of the data frame should appear like this:



The penguins dataset contains size measurements for three penguin species (Adelie, Chinstrap, and Gentoo) that live on the Palmer Archipelago in Antarctica. The columns include information such as body mass, flipper length, and bill length.

## **Create a plot in ggplot2**

****

Suppose you want to plot the relationship between body mass and flipper length in the three penguin species. You can choose a specific geom that fits the type of data you have. Points show the relationship between two quantitative variables. A scatterplot of points would be an effective way to display the relationship between the two variables. You can put flipper length on the x-axis and body mass on the y-axis.

Type the following code to create the plot. But before you run it, review the code piece by piece:

ggplot(data = penguins) + geom\_point(mapping = aes(x = flipper\_length\_mm, y = body\_mass\_g))

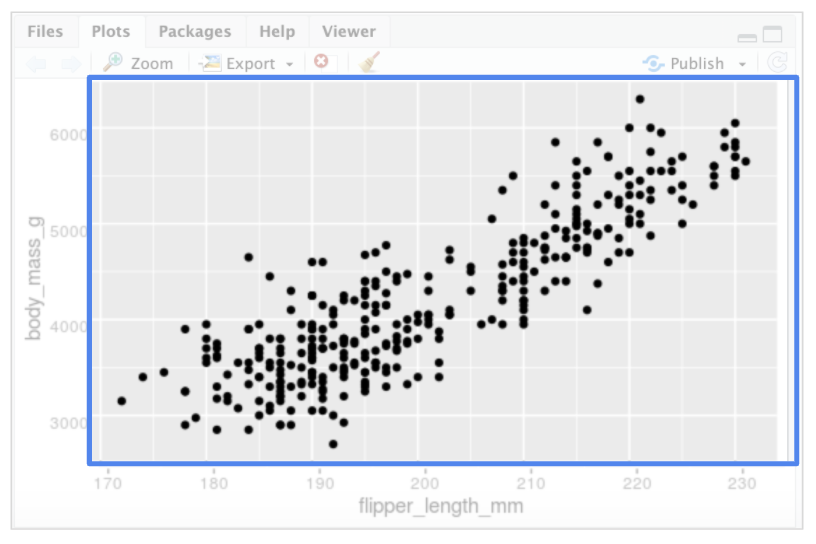
ggplot(data = penguins): In ggplot2, you begin a plot with the ggplot() function. The ggplot() function creates a coordinate system that you can add layers to. The first argument of the ggplot() function is the dataset to use in the plot. In this case, it’s “penguins.”

+: Then, you add a “+” symbol to add a new layer to your plot. You complete your plot by adding one or more layers to ggplot().

geom\_point(): Next, you choose a geom by adding a geom function. The geom\_point() function uses points to create scatterplots, the geom\_bar function uses bars to create bar charts, and so on. In this case, choose the geom\_point function to create a scatter plot of points. The ggplot2 package comes with many different geom functions. You’ll learn more about geoms later in this course.

(mapping = aes(x = flipper\_length\_mm, y = body\_mass\_g)): Each geom function in ggplot2 takes a mapping argument. This defines how variables in your dataset are mapped to visual properties. The mapping argument is always paired with the aes() function. The x and y arguments of the aes() function specify which variables to map to the x-axis and the y-axis of the coordinate system. In this case, you want to map the variable “flipper\_length\_mm” to the x-axis, and the variable “body\_mass\_g” to the y-axis.

Now go ahead and run the code. When you do, you get the following plot:



The plot shows a positive relationship between the two variables. In other words, the larger the penguin, the longer the flipper.

## **Create your own plot**

****

To create your own plot using code, follow these three steps:

1. Start with the ggplot() function and choose a dataset to work with.

2. Add a geom\_ function to display your data.

3. Map the variables you want to plot in the arguments of the aes() function.

Try plotting with different datasets using different geoms and mapping arguments. Coming up in this course, you’ll learn even more about the process of creating a plot. You’ll also get a chance to work with the Penguins dataset to create lots of different plots in ggplot2.

Pro-Tip: You can write the same section of code above using a different syntax with the mapping argument inside the ggplot() call: ggplot(data = penguins, mapping = aes(x = flipper\_length\_mm, y = body\_mass\_g)) + geom\_point()

## **The ggplot2 cheat sheet**

****

This is just the beginning of what you can do with ggplot2. If you want to find out more about ggplot2, RStudio has a useful reference guide called the “Data Visualization with ggplot2 Cheat Sheet.” You can use the Cheat Sheet as a quick reference while you work to learn about the main functions and features of ggplot2.

Click the link to check it out: [Cheat Sheet](https://ggplot2.tidyverse.org/)

[Getting started with ggplot()](https://www.coursera.org/learn/data-analysis-r/lecture/tziSv/getting-started-with-ggplot)

[Common problems when visualizing in R](https://www.coursera.org/learn/data-analysis-r/supplement/cwdaL/common-problems-when-visualizing-in-r)

[Hands-On Activity: Using ggplot](https://www.coursera.org/learn/data-analysis-r/quiz/pV7we/hands-on-activity-using-ggplot)

[Visualizations in Tableau versus R](https://www.coursera.org/learn/data-analysis-r/discussionPrompt/UlRwR/visualizations-in-tableau-versus-r)

[Joseph: Career path to people analytics](https://www.coursera.org/learn/data-analysis-r/lecture/lMGsU/joseph-career-path-to-people-analytics)

[Test your knowledge on data visualizations in R](https://www.coursera.org/learn/data-analysis-r/quiz/DvivC/test-your-knowledge-on-data-visualizations-in-r)

EXPLORE AESTHETICS IN ANALYSIS

[Enhancing visualizations in R](https://www.coursera.org/learn/data-analysis-r/lecture/A7ESc/enhancing-visualizations-in-r)

[Aesthetic attributes](https://www.coursera.org/learn/data-analysis-r/supplement/jvaYe/aesthetic-attributes)

[Doing more with ggplot](https://www.coursera.org/learn/data-analysis-r/lecture/qoIua/doing-more-with-ggplot)

[Smoothing](https://www.coursera.org/learn/data-analysis-r/supplement/06qO1/smoothing)

[Aesthetics and facets](https://www.coursera.org/learn/data-analysis-r/lecture/XXkuD/aesthetics-and-facets)

[Hands-On Activity: Aesthetics and visualizations](https://www.coursera.org/learn/data-analysis-r/quiz/g1z2B/hands-on-activity-aesthetics-and-visualizations)

[Filtering and plots](https://www.coursera.org/learn/data-analysis-r/supplement/M8obB/filtering-and-plots)

[Hands-On Activity: Filters and plots](https://www.coursera.org/learn/data-analysis-r/quiz/8Myvu/hands-on-activity-filters-and-plots)

[Elements of ggplot](https://www.coursera.org/learn/data-analysis-r/ungradedWidget/oSPAH/elements-of-ggplot)

[Test your knowledge on aesthetics in analysis](https://www.coursera.org/learn/data-analysis-r/quiz/jN8vO/test-your-knowledge-on-aesthetics-in-analysis)

ANNOTATE AND SAVE VISUALIZATIONS

[Annotation layer](https://www.coursera.org/learn/data-analysis-r/lecture/6RujH/annotation-layer)

[Adding annotations in R](https://www.coursera.org/learn/data-analysis-r/supplement/R8VGv/adding-annotations-in-r)

[Saving your visualizations](https://www.coursera.org/learn/data-analysis-r/lecture/3UHgo/saving-your-visualizations)

[Saving images without ggsave()](https://www.coursera.org/learn/data-analysis-r/supplement/zXYZ9/saving-images-without-ggsave)

[Hands-On Activity: Annotating and saving visualizations](https://www.coursera.org/learn/data-analysis-r/quiz/NEmsh/hands-on-activity-annotating-and-saving-visualizations)

[Test your knowledge on annotating and saving visualizations](https://www.coursera.org/learn/data-analysis-r/quiz/VNkC8/test-your-knowledge-on-annotating-and-saving-visualizations)

M4 CHALLENGE

[Glossary: Terms and definitions](https://www.coursera.org/learn/data-analysis-r/supplement/dmmFf/glossary-terms-and-definitions)