Notes 04 - Data Visualization with ggplot2

STS 2300 (Fall 2024)

Updated: 2024-09-15

Table of Contents

[Reading for Notes 04 1](#_Toc177306018)

[Learning Goals for Notes 04 2](#_Toc177306019)

[The ggplot2 package 2](#_Toc177306020)

[Types of graphs 3](#_Toc177306021)

[Scatter plots 4](#_Toc177306022)

[Histograms 4](#_Toc177306023)

[(Side-by-side) Boxplots 5](#_Toc177306024)

[Bar graphs 6](#_Toc177306025)

[geom\_bar() for raw data 7](#_Toc177306026)

[geom\_col() for summarized data 7](#_Toc177306027)

[Adding additional variables to our graphs 8](#_Toc177306028)

[Revisiting the Learning Goals for Notes 04 9](#_Toc177306029)

# Reading for Notes 04

Read [Chapter 2 of ModernDive](https://moderndive.com/2-viz.html) to learn about creating graphs with the ggplot2 package. (Note: Some topics in this chapter, like Section 2.6 will be covered in Notes 05.)

The [ggplot2 cheat sheet](https://rstudio.github.io/cheatsheets/data-visualization.pdf) will also be helpful as a reference.

Additional optional resources include [Chapter 10 of R for Data Science](https://r4ds.hadley.nz/layers) and [Chapter 3 of Modern Data Science with R](https://mdsr-book.github.io/mdsr3e/03-datavizII.html).

For a **much** more in depth look, see the [R Graphics Cookbook](https://r-graphics.org/)

# Learning Goals for Notes 04

* Understand the ggplot2 grammar of graphics including how to use the aes() function.
* Be able to use the ggplot2 package to create histograms, boxplots, bar graphs, and scatterplots.
* Be able to display information about additional variables in these graphs using color and other features.

# The ggplot2 package



ggplot2 Cartoon (from <https://github.com/allisonhorst/stats-illustrations>)

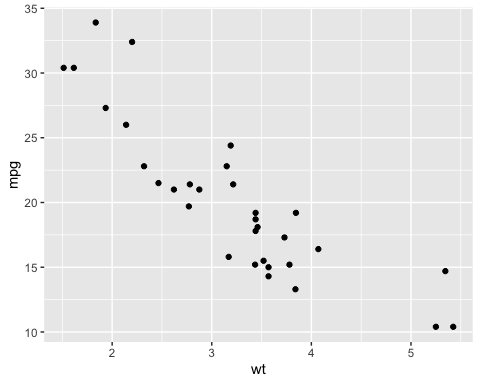
ggplot2 is a package that lets you make nice looking graphs quickly while using consistent syntax. This is in contrast with “base graphics”, which are graphing functions pre-installed in R. The letters “gg” stand for “grammar of graphics”.

To make graphs using the grammar of graphics for this package, we will

1. start with a data frame
2. map variables from the data frame to aesthetic features of our graph (like the x-axis, the y-axis, the color of the points, etc.)
3. and add geometric objects (called geoms) to our graph (like bars on a bar graph or points in a scatterplot).

Below is an example of how we can make a scatterplot:

library(ggplot2)  
  
ggplot(data = mtcars) +  
 geom\_point(aes(x = wt,   
 y = mpg))



So how do I understand the code above?

* First, the ggplot() function is creating an object for graphing, and I’m specifying that I want to use the mtcars data frame.
* Then I’m adding a “points” geometric object using the geom\_point() function.
* Lastly, I’m using the aes() function inside geom\_point() to *map* variables in my data frame to specific *aesthetic* elements of my scatterplot. In this case, I’m mapping the weight of the cars (variable wt) to the x axis and the mileage per gallon of the cars (variable mpg) to the y axis.

**Practice Questions**

1. Take the code above and run it on your own to get this graph (make sure you’ve installed and loaded the ggplot2 package).
2. Try changing the code to make a scatterplot for horsepower (the variable is called hp) on the x axis vs. miles per gallon on the y axis.
3. (Bonus) Try using geom\_bar() (instead of geom\_point()) to create a bar graph of transmission types. The transmission variable is called am (with 0 = automatic and 1 = manual transmission). You will only have *one* variable to map to an aesthetic here. We will cover bar graphs in more detail below, but it will be helpful to think through the process of adapting this code before that.

# Types of graphs

There are many different kinds of graphs we can make with the ggplot2 (and even more variations on those graphs). For example, the [R Graph Gallery](https://r-graph-gallery.com/) includes lots of examples. In many cases, there will be more than one way to write code to create these graphs. For now, we will just scratch the surface of what ggplot2 can do, but we will expand on this in Notes 05, and I encourage you to explore ways to create graphs of interest to you.

## Scatter plots

Scatter plots are typically used to display the relationship between two numerical variables. We saw above that scatter plots are created by adding a point geom to an object created by ggplot(). In general, our code will look like this:

ggplot(data = dataset\_name) +  
 geom\_point(aes(x = explanatory\_var,   
 y = response\_var))

There are lots of other ways we can customize our scatterplots within geom\_point(). Below are some examples of other arguments:

* color controls the color of the points (default is “black”)
* size controls how big the points are (default is 1)
* shape controls the shape of the points (see [here](http://www.sthda.com/sthda/RDoc/images/points-symbols.png) for more options)
* alpha controls the transparency of the points (0 is completely see through and 1 is completely solid)

We will put these arguments **outside** of the aes() function unless we want to map another variable in our dataset to one of these aesthetic elements (more on this later).

**Practice:** Take the scatterplot we made above and see if you can make the points purple, bigger than they were before, slightly transparent, and triangles instead of circles.

## Histograms

Unlike scatterplots, histograms only show the distribution of a single numerical variable. Below is some generic code for creating histograms. You can replace the template code with a data frame and the quantitative variable you’re interested in.

ggplot(data = dataset\_name) +  
 geom\_histogram(aes(x = quant\_var))

Notice that R prints a message telling you that the default number of bins was used and that this may not be ideal. You can fix this by specifying either of the following two arguments within geom\_histogram() (but outside of aes()):

1. bins = \_\_ argument to specify how many bins your histogram should have or
2. binwidth = \_\_ argument to specify how wide each bin should be.

You may also want to use the boundary = \_\_ argument to say where a bin should start (e.g., start at 0). You only need to specify this for one bin because all of the rest will follow from that.

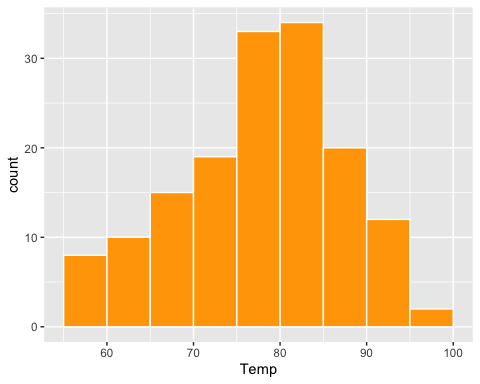
If you want to add color to your graph, you can use

1. fill = \_\_ to change the color inside the bars or
2. color = \_\_ to change the color of the lines around the bars.

Replace \_\_ with a color name in quotes (like “violet”). Many common color names will work or you can see a complete list of options [here](http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf).

(Note: Like with scatterplots, the alpha argument can be used to make the bars transparent.)

**Practice**: See if you can use what you’ve learned to recreate the graph below. It uses the airquality data frame that is built into R. Instead of trying to get the graph perfect right away, pick one element at a time to work on.



## (Side-by-side) Boxplots

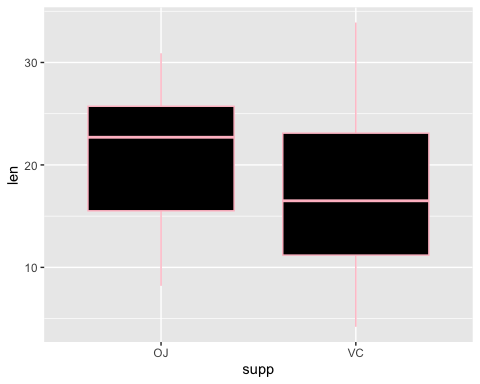
While it’s possible to make a single boxplot, these graphs are most often used to compare the distributions of a numerical variable across multiple groups. Below is some generic code for creating side-by-side boxplots. You can replace the template code with a data frame and your quantitative and categorical variables to make your own.

ggplot(data = dataset\_name) +  
 geom\_boxplot(aes(x = cat\_var,   
 y = quant\_var))

(**Note**: Want horizontal boxplots instead of vertical ones? Just switch x and y!)

Like with histograms, the fill and color arguments can be used to change the inside of the boxes and the lines around the boxes respectively (and alpha can be used to make the boxes transparent).

**Practice**: See if you can use what you’ve learned to create the following graph that uses the ToothGrowth dataset.



Boxplots represent the distribution of a quantitative variable by splitting the data into quarters via what we call the **five number summary** of variable, which consists of the following values:

1. The minimum (the bottom of the line)
2. The first quartile, or Q1 (the bottom line of the box)
3. The median (the thick line inside the box)
4. The third quartile, or Q3 (the top line of the box)
5. The maximum (the top of the line)

These numbers are used to split our data into fourths. In other words, 25% of the data is between the min and Q1, 25% of the data is between Q1 and the median, 25% of the data is between the median and Q3, and the last 25% of the data is between Q3 and the max.

## Bar graphs

There are two different ways to create bar graphs using the ggplot2 package, and which method we use depends on what our data looks like.

### geom\_bar() for raw data

The first option is for “raw data” where each row in the data represents a person or object. In this situation, we use the geom\_bar() function to add a bar graph geom to our ggplot() object.

Below is some generic code for creating bar graphs using this method. You can replace the template code with a data frame and your categorical variable to make your own.

# Option 1 - Raw data  
  
ggplot(data = dataset\_name) +  
 geom\_bar(aes(x = cat\_var))

### geom\_col() for summarized data

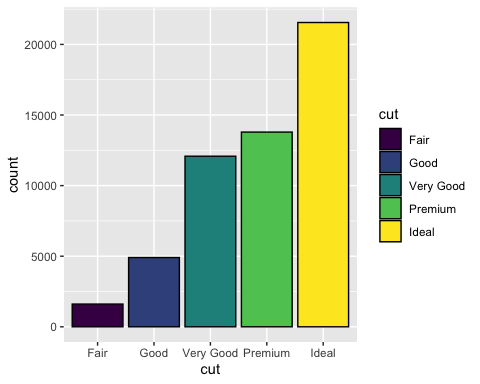
The second option is for summarized data in which you have one variable for your categories and one corresponding to how many people / objects are in that category. This means each row will represent many people/objects instead of a single observation like the “raw data” case. In this situation, we use the geom\_col() function to add a bar graph geom to our ggplot() object.

Below is some generic code for creating bar graphs using this method. You can replace the template code with a data frame and your categorical variable to make your own. You will also have a variable that contains the counts of how many people/objects are in each category.

# Option 2 - Summarized data  
  
ggplot(data = dataset\_name) +  
 geom\_col(aes(x = cat\_var,   
 y = count\_var))

As with the two previous types of graphs, you can use fill and color arguments to make the graphs look nicer.

**Practice**: See if you can create the following graph that uses the diamonds dataset in the ggplot2 package. You may want to look at the help file for the data set to decide if you need to use geom\_bar() or geom\_col(). (**Note**: Rather than make each bar correspond to the same color, I mapped a variable to the bar color. How do you think I could do that in my code?)



# Adding additional variables to our graphs

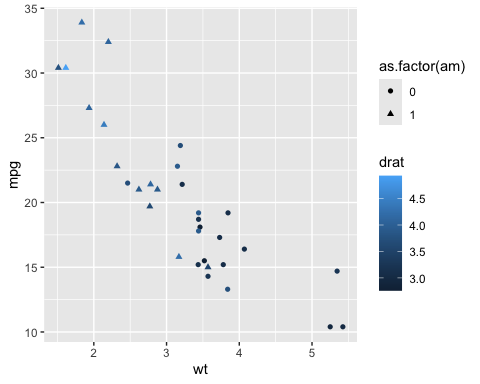
In Activity 04, we will see some examples of how additional variables can be added to specific types of graphs. However, for most arguments that we can specify outside aes(), we can also choose to specify them inside aes() to map a variable to that element of the graph.

For example, I may want to make the bars of a bar graph have colors corresponding to their category instead of all having the same color. To do that, I assign my x variable to the fill argument too (inside of aes()).

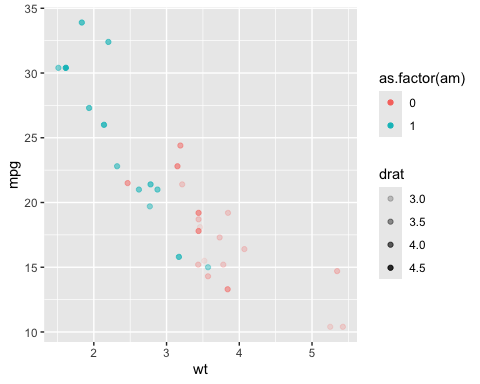
ggplot(data = diamonds) +  
 geom\_bar(aes(x = cut,   
 fill = cut),   
 color = "black")

In a scatterplot, I may want the size, color, shape, or transparency of my points to correspond with a different variable in my data. If that variable is treated as categorical, I will have different values for each category. If that variable is treated as numeric, I will get a continuum of values. Below are two examples.

# Continuum of colors but two separate shapes  
ggplot(mtcars) +  
 geom\_point(aes(x = wt,   
 y = mpg,   
 color = drat,   
 shape = as.factor(am)))



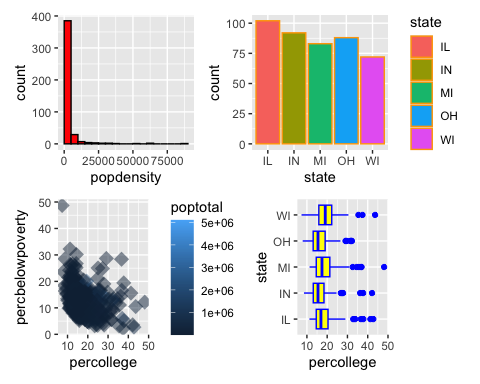
# Two separate colors but continuum of transparencies  
ggplot(mtcars) +  
 geom\_point(aes(x = wt,   
 y = mpg,   
 color = as.factor(am),   
 alpha = drat))



We will continue to explore this in Activity 04. In Notes 05, we will learn how to improve our graphs by editing the theme, labels, color values, and more.

# Revisiting the Learning Goals for Notes 04

* Understand the ggplot2 grammar of graphics including how to use the aes() function.
  + Which function do we start each graph with?
  + How do we know whether an argument like color should go inside or outside of the aes() function?
* Be able to use the ggplot2 package to create histograms, boxplots, bar graphs, and scatterplots.
  + Attempt to recreate each of the graphs below (that uses the midwest data frame from the ggplot2 package) to the best of your ability.



* Be able to display information about additional variables in these graphs using color and other features.
  + See examples above