Making graphs in R - GGPLOT2

Juan Mejia
jmejiare@chicagobooth.edu

Agenda

1. What is Tidyverse?
2. What is GGPLOT2?
3. GGPLOT Usage Basics
4. GGPLOT Graphic Template
5. GGPLOT Components Explained
6. Proceed to R Markdown for some examples

What is Tidyverse?

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Basically it makes R behave a little bit more like SQL!

What is GGPLOT2?

ggplot2 is a system for declaratively creating graphics, based on 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

```r
# The easiest way to get ggplot2 is to install the whole tidyverse:
install.packages("tidyverse")


# Alternatively, install just ggplot2:
install.packages("ggplot2")


# Or the the development version from GitHub:
# install.packages("devtools")
devtools::install_github("tidyverse/ggplot2")
```

GGPLOT Usage Basics

In most cases you start with ggplot(), **(TOOL)**

Supply a dataset **(DATA)**

Specify an aesthetic (at the very least the x and y variables) **(HOW DO I WANT MY DATA TO LOOK)**

And then you add on layers such as:

- Geometric Objects **(TYPE OF CHART)**
- Stat **(MATHEMATICAL OPERATOR)**
- Position **(HOW IS MY DATA PRESENTED - E.G STACKING)**
- Coordinate Function **(WHAT IS THE ORIENTATION OF MY DATA)**
- Facet function **(DATA DIVISIONS)**

GGPLOT GRAPHING TEMPLATE

```
 ggplot(data = <DATA>) +

 <GEOM_FUNCTION>( mapping = aes(<MAPPINGS>), stat = <STAT>,

  position = <POSITION>) +

<COORDINATE_FUNCTION> +

<FACET_FUNCTION>
```

```
ggplot(data = mpg) +
geom_point(mapping = aes(x = displ, y = hwy, color = class, position = “jitter”)) +
 labs( x = "engine size", y = "miles per gallon",
title ="Car Efficiency",
subtitle = "Subdivided by class",
caption = "Analytics Group Booth") +
facet_wrap (~year) +
coord_flip()
```

GGPLOT COMPONENTS EXPLAINED

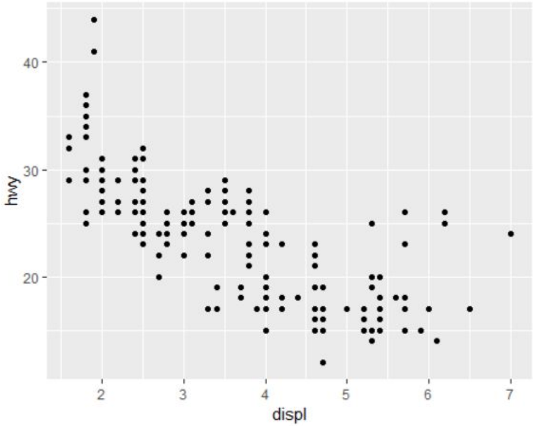
1. Geometric Objects
2. Aesthetic Mappings
3. Facets
4. Statistical Transformations (Stat)
5. Position Adjustments
6. Coordinate Systems

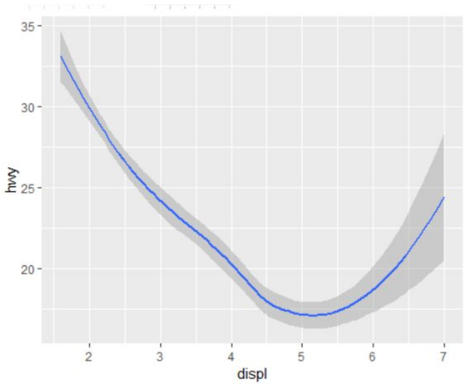
Geometric Objects (GEOMS)

A **geom** is the geometrical object that a plot uses to represent data.

For example:

- Bar charts use **bar geoms**
- Line charts use **line geoms**
- Boxplots use **boxplot geoms**
- Scatterplots break the trend; they use the **point geom**





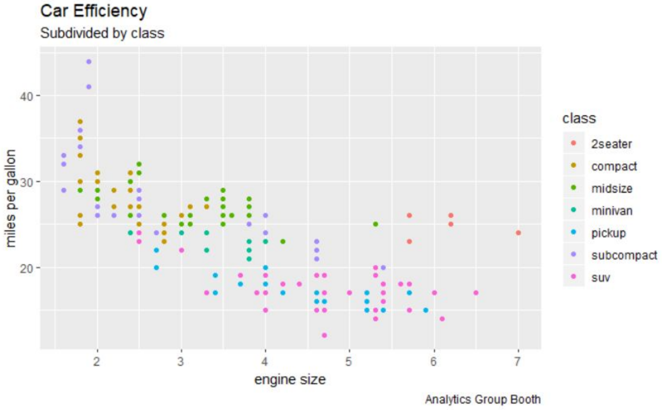
https://ggplot2.tidyverse.org/reference/

Aesthetic Mappings (aes)

An **aesthetic** is a **visual property** of the objects in your plot.

Aesthetics include things like the **size, the shape, or the color of your points.**

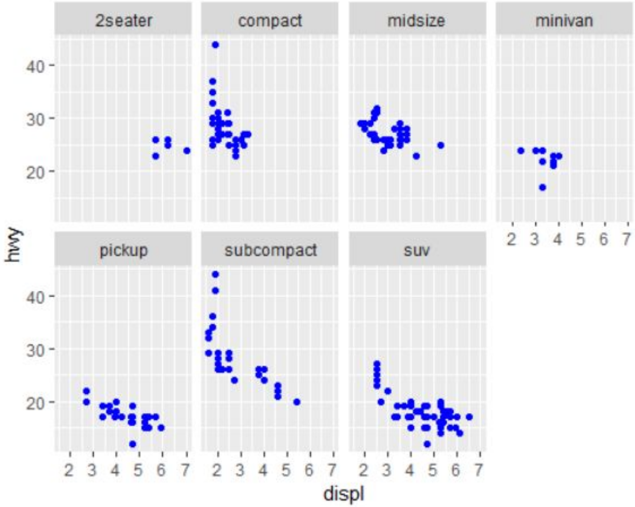
geom_point(mapping = aes(x = displ, y = hwy),color = "blue")



Facets

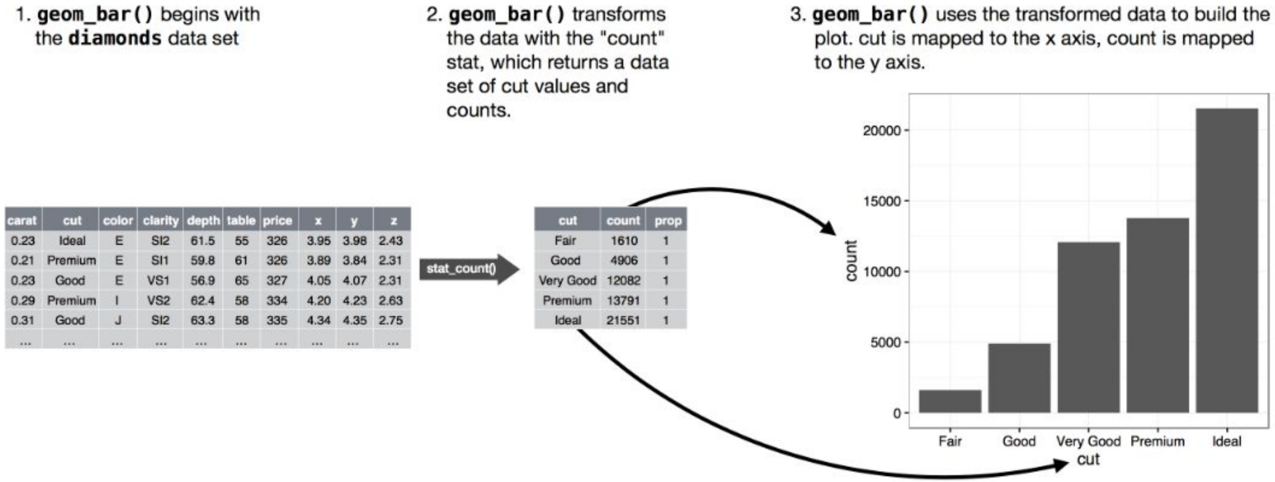
Facets are a way to add additional variables to your data, by splitting it into subplots! (This is amazing!)

facet_wrap(~ class, nrow = 2)



Statistical Transformations (Stats)

Stats are statistical transformations performed by certain GEOM objects such as bar charts! (Confusing, right?)



1. **geom_bar()** begins with the **diamonds** data set

2. **geom_bar()** transforms the data with the "count" stat, which returns a data set of cut values and counts.

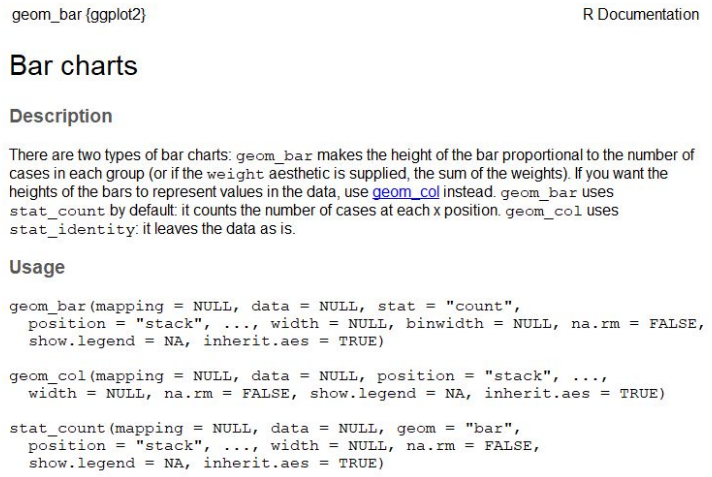
3. **geom_bar()** uses the transformed data to build the plot. cut is mapped to the x axis, count is mapped to the y axis.

carat	cut	color	clarity	depth	table	price	x	y	z
0.23	Ideal	E	SI2	61.5	55	326	3.95	3.98	2.43
0.21	Premium	E	SI1	59.8	61	326	3.89	3.84	2.31
0.23	Good	E	VS1	56.9	65	327	4.05	4.07	2.31
0.29	Premium	I	VS2	62.4	58	334	4.20	4.23	2.63
0.31	Good	J	SI2	63.3	58	335	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...

cut	count	prop
Fair	1610	1
Good	4906	1
Very Good	12082	1
Premium	13791	1
Ideal	21551	1

Statistical Transformations (Stats) - Continued

You can learn which stat a geom uses by inspecting the default value for the `stat` argument. For example, `?geom_bar` shows that the default value for `stat` is “count”, which means that `geom_bar()` uses `stat_count()`



geom_bar {ggplot2}   R Documentation

Bar charts

Description

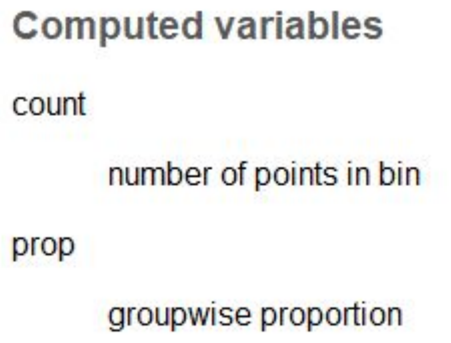
There are two types of bar charts: `geom_bar` makes the height of the bar proportional to the number of cases in each group (or if the `weight` aesthetic is supplied, the sum of the weights). If you want the heights of the bars to represent values in the data, use geom_col instead. `geom_bar` uses `stat_count` by default: it counts the number of cases at each x position. `geom_col` uses `stat_identity`: it leaves the data as is.

Usage

```
geom_bar(mapping = NULL, data = NULL, stat = "count",
  position = "stack", ..., width = NULL, binwidth = NULL, na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE)

geom_col(mapping = NULL, data = NULL, position = "stack", ...,
  width = NULL, na.rm = FALSE, show.legend = NA, inherit.aes = TRUE)

stat_count(mapping = NULL, data = NULL, geom = "bar",
  position = "stack", ..., width = NULL, na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE)
```



Computed variables

count

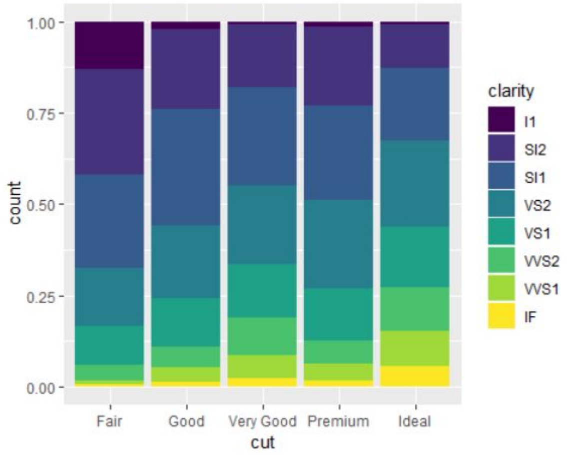
number of points in bin

prop

groupwise proportion

Position Adjustments

Position Adjustments help you view your data more clearly by helping you stack chart, removing or adding noise to a dataset, overplacing objects beside each other, etc

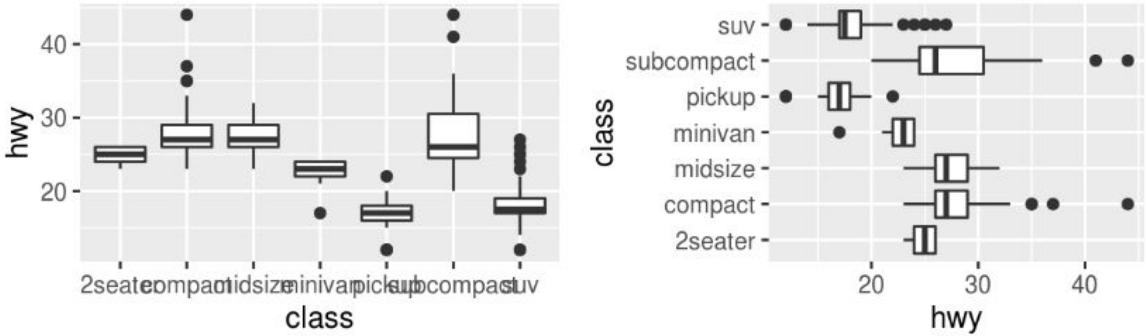


Coordinate Systems

Coordinate systems are probably the most complicated part of ggplot2. The default coordinate system is the Cartesian coordinate system where the x and y positions act independently to determine the location of each point.

I DON’T KNOW MUCH ABOUT THEM ---> BUT…

One coordinate system command is helpful….. COORD_FLIP() switches the x and y axes



Now we will proceed to the RMarkdown!

Important:

- Take a look at the mpg and diamonds datasets
- A lot of info, we might not have time to go through everything!
- If needed we will do one more Visualization workshop (intermediate)