



BIMM 143
Hands-on Lab Session
Class 05
Barry Grant
UC San Diego
<http://thegrantlab.org/bimm143>

How do we make informative and compelling figures?



ggplot2
www.rstudio.com



ggplot2

Currently the premier plotting library on the planet!

Key Insight: All visualizations map data into quantifiable aesthetic features of the resulting graphic

Key Insight: All visualizations map data into quantifiable aesthetics features of the resulting graphic

data → aesthetics



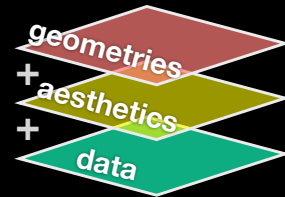
data + aesthetics + geometrys

Three main "layers"
that are in every ggplot



data + aesthetics + geometrys

Three main "layers" that are in every ggplot



data + aesthetics + geometrys

```
ggplot(data=mpg) +
  aes(x=displ, y=hwy, color=class) +
  geom_point()
```

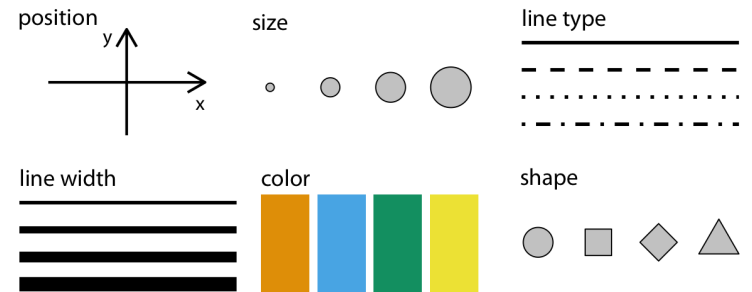


data + aesthetics + geometrys

```
ggplot(data=mpg) +
  aes(x=displ, y=hwy, color=class) +
  geom_point()
```



Common aesthetics include



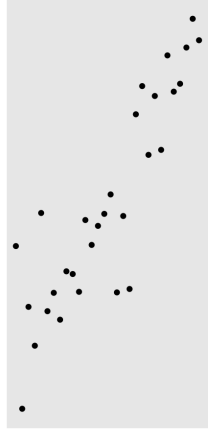
Modified from: Wilke (2019)



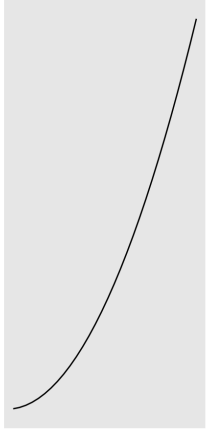
data + aesthetics + geometrys

Three main "layers" that are in every ggplot

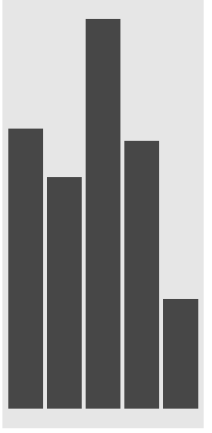
geom_point()



geom_line()



geom_col()



Data Visualization with ggplot2 :: CHEAT SHEET

Cheat sheet for ggplot2 with sections for Basics, Geoms, and various functions like geom_point, geom_line, geom_col, etc.

Learn more about core geom_FUNCTIONS()

There are > 40 core "geom" functions. See cheat-sheet link on class website!



Screenshot of the R Studio interface showing the console, environment, and help windows.

Follow Along!

R Studio

Follow Along!

```

R version 3.6.0 (2019-04-26) -- "Planting a Tree"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(ggplot2)
> ggplot(mpg, aes(displ, hwy, color=class)) +
  geom_point()
  
```

Environment is empty

Class:
 - suv
 - subcompact
 - pickup
 - minivan
 - midsize
 - compact
 - 2seater

RStudio >

Create a new **Project** and open a new **R Script**
 (N.B. make a **report** with notes and plots)

In addition to your **PDF lab report** answer the **inbuilt questions**

Question Counter

Questions

1. Overview
 2. Background
 3. Getting Organized
 4. Common Plot Types
 5. Creating Scatter Plots

Introduction to scatter plots
 Specifying a dataset with `ggplot()`
 Specifying aesthetic mappings with `aes()`
 Specifying a geom layer with `geom_point()`
 Adding more plot aesthetics through `aes()`

6. OPTIONAL: Going Further
 7. OPTIONAL: Bar Charts
 About this document

5. Creating Scatter Plots

In this section we will focus on:

- Defining a dataset for your plot using the main `ggplot()` function.
- Specifying how your data maps to plot aesthetics with the `aes()` function.
- Adding geometric layers using the `geom_point()` function.
- Combining the above function calls with `+` operator to make your plot

Q. Which plot types are typically NOT used to compare distributions of numeric variables?
 Density plots
 Network graphs
 Histograms
 Violin plots
 Box plots

Making a HTML Lab Report

Lab Report

- Save your **R script** (make sure it has some plots and comments)
- Can you **source** this **R script** file to re-generate all your plots without error?

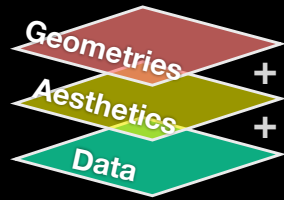
- If so you can now generate a nice **PDF report** of your work for upload to **GradeScope...**

[Optional Sections get you bonus points!]

data + aesthetics + geometrys

- Summary: ggplot takes an input *data.frame*, a mapping of columns to *aesthetics* and one or more *geom layers* (e.g. `geom_point()`, `geom_line()`, ...)

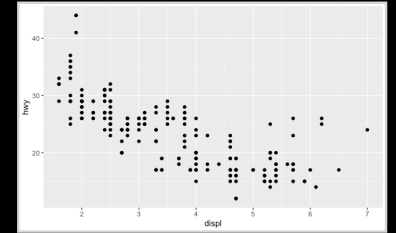
```
ggplot(data=mpg) +  
  aes(x=displ, y=hwy) +  
  geom_point()
```



data + aesthetics + geometrys

- Summary: ggplot takes an input *data.frame*, a mapping of columns to *aesthetics* and one or more *geom layers* (e.g. `geom_point()`, `geom_line()`, ...)

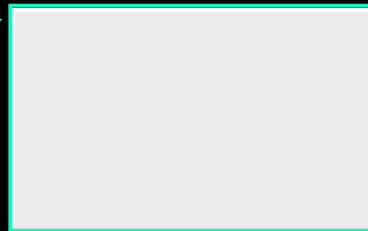
```
ggplot(data=mpg) +  
  aes(x=displ, y=hwy) +  
  geom_point()
```



data + aesthetics + geometrys

- Summary: ggplot takes an input *data.frame*, a mapping of columns to *aesthetics* and one or more *geom layers* (e.g. `geom_point()`, `geom_line()`, ...)

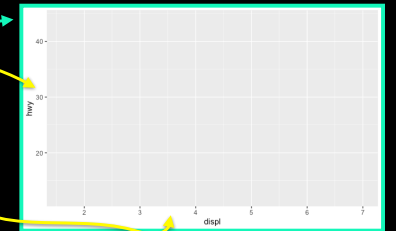
```
ggplot(data=mpg)
```



data + aesthetics + geometrys

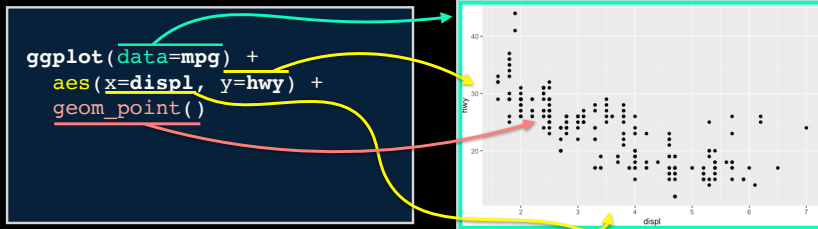
- Summary: ggplot takes an input *data.frame*, a mapping of columns to *aesthetics* and one or more *geom layers* (e.g. `geom_point()`, `geom_line()`, ...)

```
ggplot(data=mpg) +  
  aes(x=displ, y=hwy)
```



data + aesthetics + geometrys

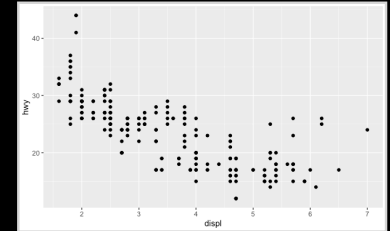
- Summary: ggplot takes an input *data.frame*, a mapping of columns to *aesthetics* and one or more *geom layers* (e.g. `geom_point()`, `geom_line()`, ...)



data + aesthetics + geometrys

- We can keep building more complicated plots by adding more *layers*

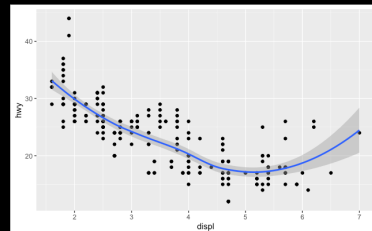
```
ggplot(data=mpg) +  
  aes(x=displ, y=hwy) +  
  geom_point()
```



data + aesthetics + geometrys

- We can keep building more complicated plots by adding more *layers*
- For example lets add another *geom*, in this case a smooth line fitted to the data...

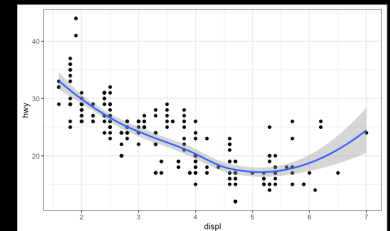
```
ggplot(data=mpg) +  
  aes(x=displ, y=hwy) +  
  geom_point() +  
  geom_smooth()
```



data + aesthetics + geometrys

- We can also add other customizations like *themes*...

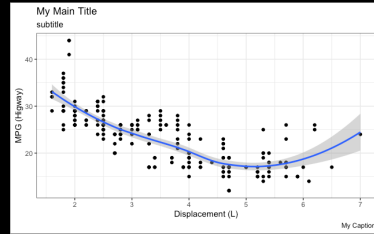
```
ggplot(data=mpg) +  
  aes(x=displ, y=hwy) +  
  geom_point() +  
  geom_smooth() +  
  theme_bw()
```



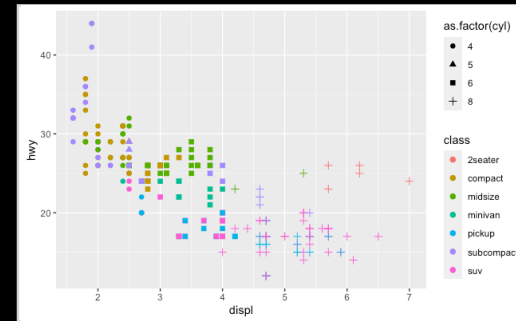
data + aesthetics + geometrys

- And various custom annotation labels...

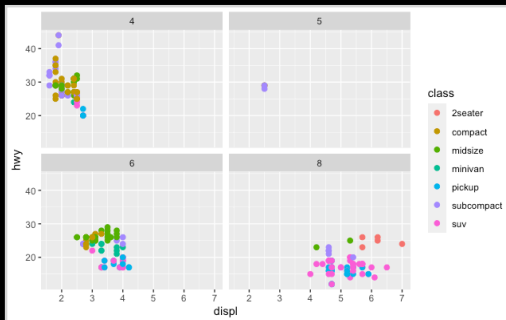
```
ggplot(data=mpg) +
  aes(x=displ, y=hwy) +
  geom_point() +
  geom_smooth() +
  theme_bw() +
  labs(title="My Main Title",
       subtitle="subtitle",
       caption="My Caption",
       x="Displacement (L)",
       y="MPG (Higway)")
```



```
ggplot(data=mpg) +
  aes(x=displ, y=hwy, color=class,
      shape=factor(cyl)) +
  geom_point()
```



```
ggplot(data=mpg) +
  aes(x=displ, y=hwy, color=class) +
  geom_point() +
  facet_wrap(~cyl)
```



Data Visualization with ggplot2 :: CHEAT SHEET

Basics

ggplot() is based on the grammar of graphics. This idea says you build every graph from 3 main components: **data**, **aesthetics**, and **geoms**—visual marks that represent data points.

To describe values, map variables in the data to visual properties of the geom (aesthetics) like size, color, and alpha properties.

Graphical Primitives

- `geom_point()`: scatter plot
- `geom_smooth()`: smoothed trend line
- `geom_line()`: line plot
- `geom_rect()`: rectangle
- `geom_text()`: text annotation
- `geom_label()`: text with background
- `geom_vline()`: vertical line
- `geom_hline()`: horizontal line
- `geom_fgeom()`: free-form geometry

Faceting

- `facet_wrap()`: wrap plots around a variable
- `facet_grid()`: grid of plots

Annotations

- `labs()`: titles and subtitles
- `annotate()`: add text or geom to a plot
- `geom_text()`: text annotation
- `geom_label()`: text with background
- `geom_vline()`: vertical line
- `geom_hline()`: horizontal line
- `geom_fgeom()`: free-form geometry

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables.

Graphical Primitives

- `geom_point()`: scatter plot
- `geom_smooth()`: smoothed trend line
- `geom_line()`: line plot
- `geom_rect()`: rectangle
- `geom_text()`: text annotation
- `geom_label()`: text with background
- `geom_vline()`: vertical line
- `geom_hline()`: horizontal line
- `geom_fgeom()`: free-form geometry

Faceting

- `facet_wrap()`: wrap plots around a variable
- `facet_grid()`: grid of plots

Annotations

- `labs()`: titles and subtitles
- `annotate()`: add text or geom to a plot
- `geom_text()`: text annotation
- `geom_label()`: text with background
- `geom_vline()`: vertical line
- `geom_hline()`: horizontal line
- `geom_fgeom()`: free-form geometry

Continuous Variables

- `geom_smooth()`: smoothed trend line
- `geom_line()`: line plot
- `geom_rect()`: rectangle
- `geom_text()`: text annotation
- `geom_label()`: text with background
- `geom_vline()`: vertical line
- `geom_hline()`: horizontal line
- `geom_fgeom()`: free-form geometry

Discrete Variables

- `geom_point()`: scatter plot
- `geom_smooth()`: smoothed trend line
- `geom_line()`: line plot
- `geom_rect()`: rectangle
- `geom_text()`: text annotation
- `geom_label()`: text with background
- `geom_vline()`: vertical line
- `geom_hline()`: horizontal line
- `geom_fgeom()`: free-form geometry

Three Variables

- `geom_smooth()`: smoothed trend line
- `geom_line()`: line plot
- `geom_rect()`: rectangle
- `geom_text()`: text annotation
- `geom_label()`: text with background
- `geom_vline()`: vertical line
- `geom_hline()`: horizontal line
- `geom_fgeom()`: free-form geometry

© 2019 by D. Iain Flocke. All rights reserved. Learn more at <http://ggplot2.tidyverse.org>

Learn more about core `geom_FUNCTIONS()`

DataCamp course!



BIMM 143

Hands-on Lab Session

Class 05

Barry Grant

UC San Diego

<http://thegrantlab.org/bimm143>