Introduction to data visualization

Scatterplots with smooths, line plots

Daniel Anderson
Week 2, Class 1

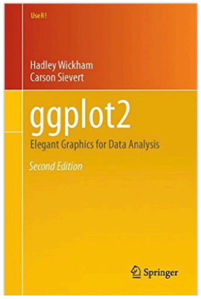
UNIVERSITY OF OREGON

Agenda

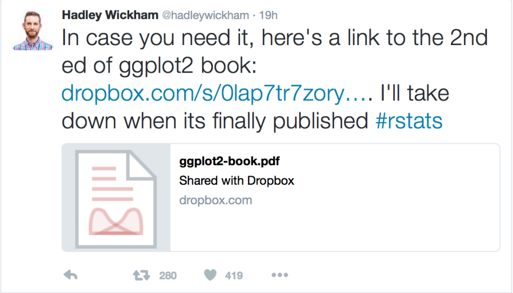
- Introduce ggplot2
- Discuss scatterplots and smooths
- Discuss line plots
- Lab

The *ggplot2* package

Today, we'll primarily be covering the basics of the *ggplot2* package.



Part of the many reasons Hadley is a good human



Hadley Wickham @hadleywickham · 19h

In case you need it, here's a link to the 2nd ed of ggplot2 book: dropbox.com/s/0lap7tr7zory…. I'll take down when its finally published #rstats

ggplot2-book.pdf
Shared with Dropbox
dropbox.com

(It's no longer there, but if you want access to it let me know)

Other resources

The *ggplot2* package is one of the most popular R packages. There are a plethora of resources to learn the syntax.

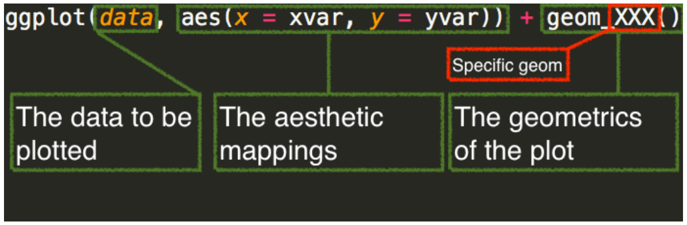
- Perhaps the most definitive, and indexes all the capabilities of ggplot2, along with multiple examples
    - http://docs.ggplot2.org/current/index.html#
- RStudio cheat sheet can also be helpful
    - https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf
- R Graphics Cookbook
    - http://www.cookbook-r.com/Graphs/

Components

Every *ggplot* plot has three components

1. data
    - The data used to produce the plot
2. aesthetic mappings
    - between variables and visual properties
3. layer(s)
    - usually through the `geom_*` function to produce geometric shape to be rendered

Basic syntax



```
ggplot(data, aes(x = xvar, y = yvar)) + geom_XXX()
```

Data for today

From ggplot: `mpg`

- Very similar to the *mtcars* data, but with more cases and a few more interesting variables

```
library(ggplot2)
head(mpg)
```

```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto… f        18    29 p     comp…
## 2 audi         a4      1.8  1999     4 manu… f        21    29 p     comp…
## 3 audi         a4      2    2008     4 manu… f        20    31 p     comp…
## 4 audi         a4      2    2008     4 auto… f        21    30 p     comp…
## 5 audi         a4      2.8  1999     6 auto… f        16    26 p     comp…
## 6 audi         a4      2.8  1999     6 manu… f        18    26 p     comp…
```

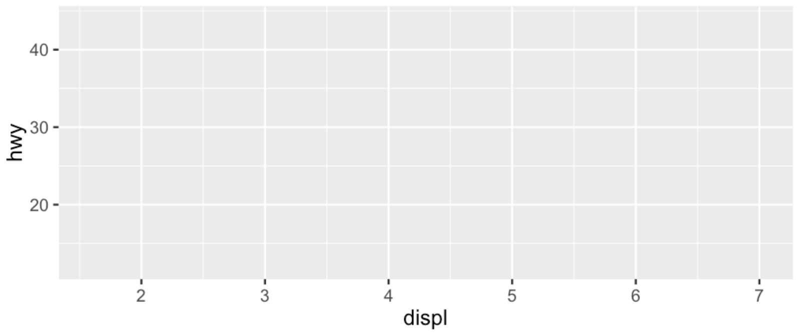
Setting up a plot

- Run the following. What do you see?

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

Plot setup

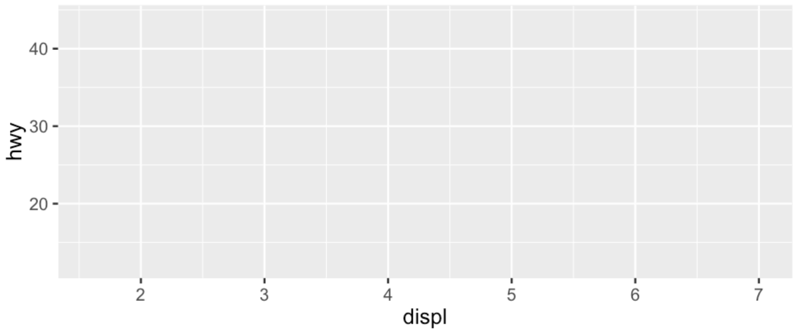
```
ggplot(mpg, aes(x = displ, y = hwy))
```



- It's ready for you to add some layers... what do you want to add?

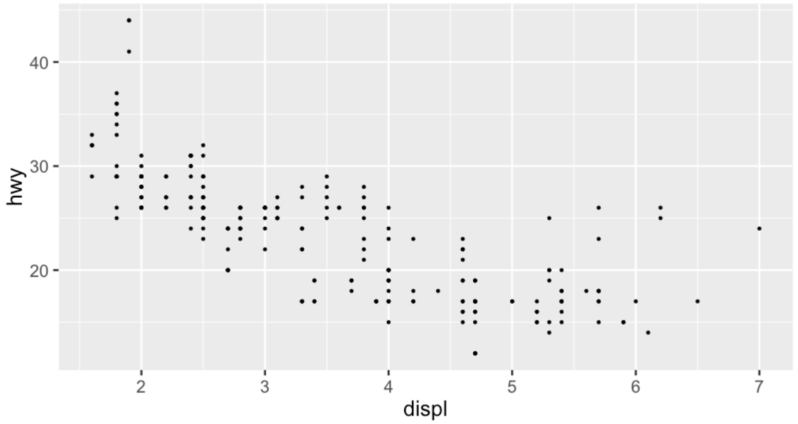
Plot setup

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



- It's ready for you to add some layers... what do you want to add?
  How about points!

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



Adding layers

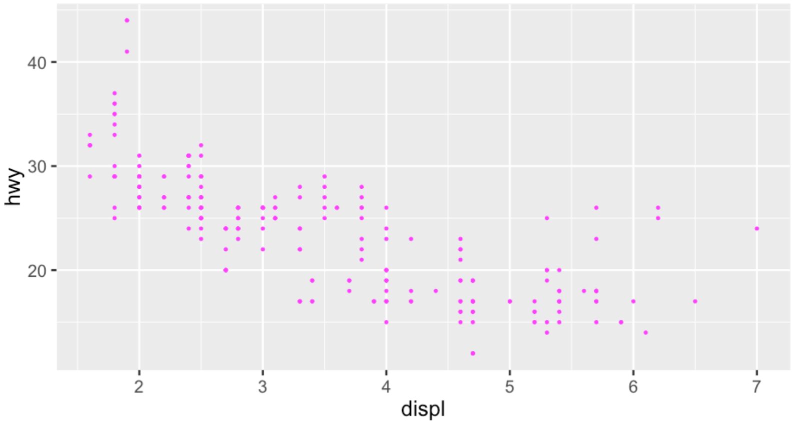
- In the previous slide, we added a layer of points
- The `geom_point` layer is a function, complete with it's own arguments
- How do you think we might change the color of the points?

Adding layers

- In the previous slide, we added a layer of points
- The `geom_point` layer is a function, complete with it's own arguments
- How do you think we might change the color of the points?

```r
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(color = "magenta")
```

```r
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(color = "magenta")
```



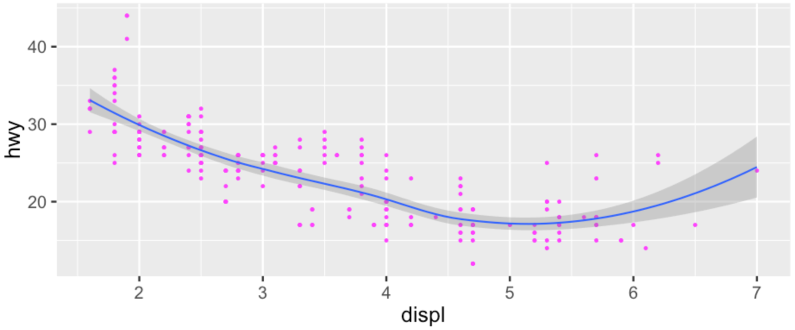
Add another layer

- Let's add a smooth with `geom_smooth()`

Add another layer

- Let's add a smooth with `geom_smooth()`

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(color = "magenta") +
  geom_smooth()
```



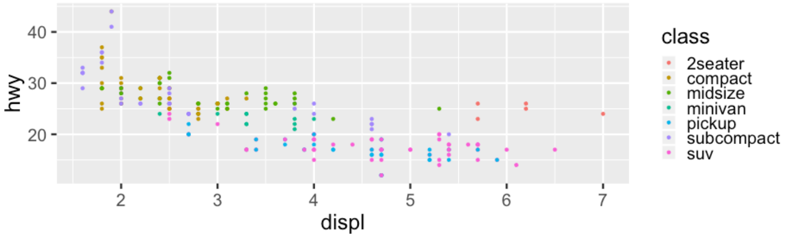
Global versus conditional coloring

- Prior examples changed colors globally
- Use `aes()` to access variables, and color **by** the specific variable

Global versus conditional coloring

- Prior examples changed colors globally
- Use `aes()` to access variables, and color **by** the specific variable

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



Conditional flow through layers

- If we use something like `color = x` in the main aesthetic, it will bleed through to all other layers.

Conditional flow through layers

- If we use something like `color = x` in the main aesthetic, it will bleed through to all other layers.
- These two lines of code are the same

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

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

Conditional flow through layers

- If we use something like `color = x` in the main aesthetic, it will bleed through to all other layers.
- These two lines of code are the same

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

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

- But these are not... why?

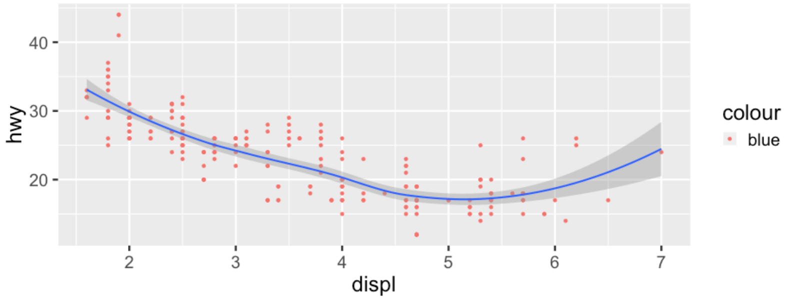
```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = class)) +
  geom_smooth()

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

Be careful with `aes()`

Using `aes` when you don't need it

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(color = "blue")) +
  geom_smooth()
```



Be careful with `aes()`

Not using `aes` when you need it

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

```
## Error in rep(value[[k]], length.out = n): attempt to replicate an object of typ
```

Challenge time

1. Start a new R project
2. Create a new script, save it as "lastname-lab2.R"
3. Load the *tidyverse*
4. Print the `msleep` dataset to see it's structure - it's within *ggplot2*.

For each of the following, produce a separate plot

1. Plot the relation between `sleep_total` and `brainwt` (with brainwt as the DV).
2. Overlay a smooth on the prior plot
3. Color the points by `vore`, but fit a single smooth
4. Fit separate smooths by `vore`, but with all points being gray
5. Omit the standard error of the smooths
6. Use `ylim` as an additional layer to restrict the y-axis to range from 0 to 5

Let's talk themes

- The default is `theme_gray`.
  - I don't like it
- Check out th *ggthemes* package for a lot of alternative
- *ggplot2* also comes with some built in alternatives
  - `theme_minimal` is my favorite
- Check out the `ggthemeassist` add-in

[demo `ggthemeassist`]

Other themes worth checking out

- The hrbrthemes are nice (and the developer is not only great, but a very nice human)
- Consider building your own theme
- When in doubt, google around a bit. For example, this one looks fairly decent that I found with about 7 seconds of searching

Other themes worth checking out

- The hrbrthemes are nice (and the developer is not only great, but a very nice human)
- Consider building your own theme
- When in doubt, google around a bit. For example, this one looks fairly decent that I found with about 7 seconds of searching

Set themes globally

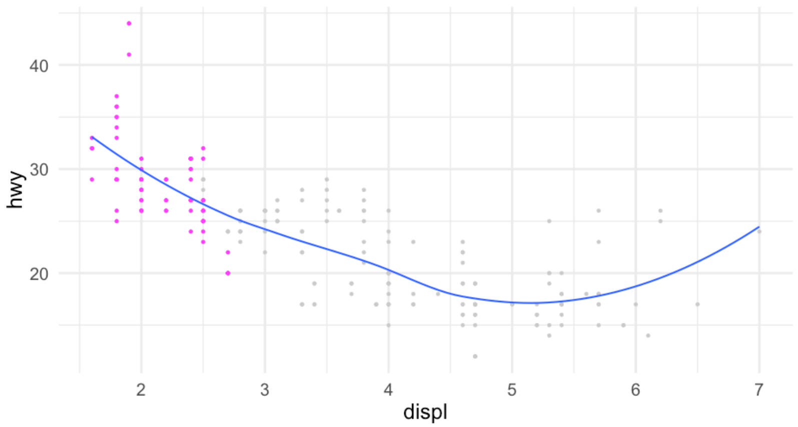
One of the first lines in many of my scripts is

```
theme_set(theme_minimal())
```

Get a little fancy

- You can use `geom_point` for more than one layer
- You can also use a different data source on a later
- Use these two properties to highlight points
    - Like maybe the 4 cylinder cars?

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(color = "gray80") +
  geom_point(data = dplyr::filter(mpg, cyl == 4),
             color = "magenta") +
  geom_smooth(se = FALSE)
```



Line plots

Discussion first

- When should you use line plots instead of smooths?
- What are some good candidate data for line plots?

Line plots

Discussion first

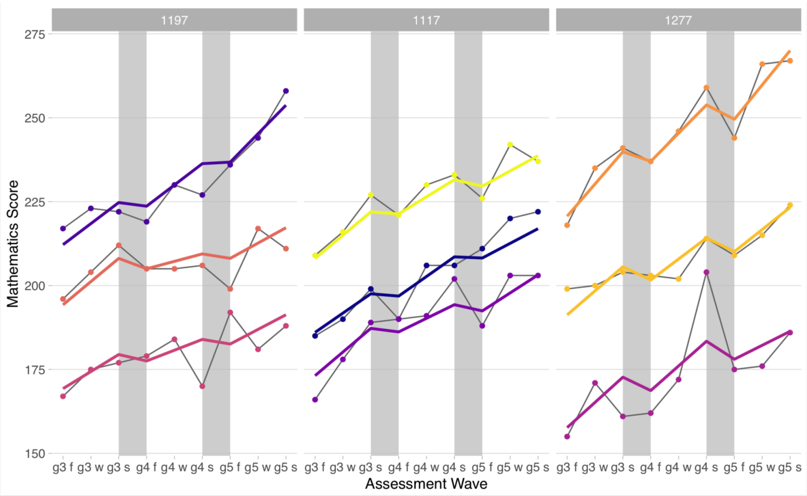
- When should you use line plots instead of smooths?
- What are some good candidate data for line plots?
- Usually when time is involved

Line plots

Discussion first

- When should you use line plots instead of smooths?
- What are some good candidate data for line plots?
- Usually when time is involved
- One of my favorites - observed versus model-implied

Example



Classical example

- Time series plot w/the economics dataset

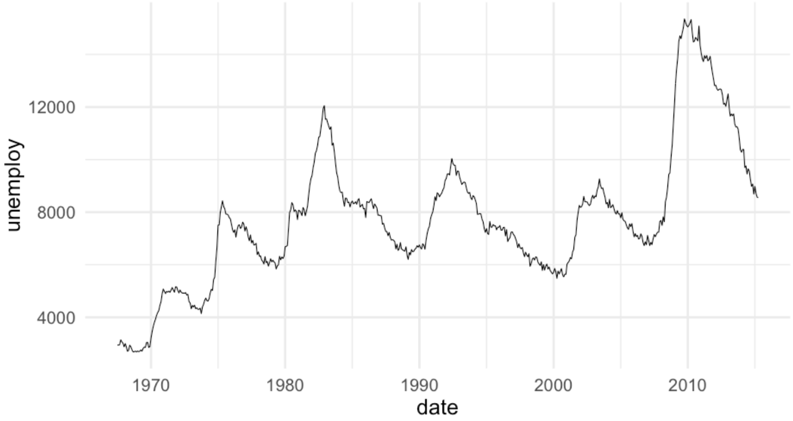
```
economics
```

```
## # A tibble: 574 x 6
##    date         pce    pop psavert uempmed unemploy
##    <date>     <dbl>  <int>   <dbl>   <dbl>    <int>
##  1 1967-07-01  507. 198712    12.5     4.5     2944
##  2 1967-08-01  510. 198911    12.5     4.7     2945
##  3 1967-09-01  516. 199113    11.7     4.6     2958
##  4 1967-10-01  513. 199311    12.5     4.9     3143
##  5 1967-11-01  518. 199498    12.5     4.7     3066
##  6 1967-12-01  526. 199657    12.1     4.8     3018
##  7 1968-01-01  532. 199808    11.7     5.1     2878
##  8 1968-02-01  534. 199920    12.2     4.5     3001
##  9 1968-03-01  545. 200056    11.6     4.1     2877
## 10 1968-04-01  545. 200208    12.2     4.6     2709
## # ... with 564 more rows
```

- How do you expect we'd fit a line plot to these data, showing the unemployment rate over time?

  Try it out!

```
ggplot(economics, aes(date, unemploy)) +
  geom_line()
```



Short challenge

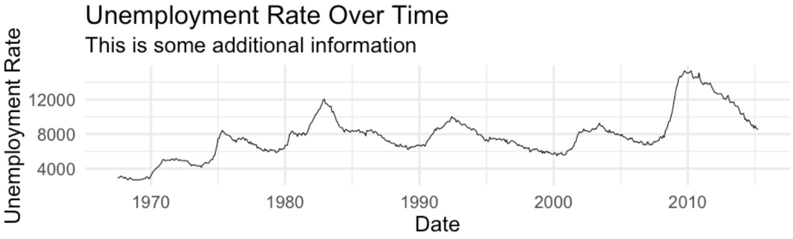
- Try adding an additional geom_ribbon layer
  - set the ymin to 0 and the ymax to `unemploy`.
  - Change the fill of the ribbon to `"darkcyan"`
  - Add transparency through the `alpha` argument
- Change line color to "gray40"
- Alternate which layer comes first - do you notice a difference?

(then demo)

Quickly

Axis labels

```
ggplot(economics, aes(date, unemploy)) +
  geom_line() +
  labs(x = "Date",
       y = "Unemployment Rate",
       title = "Unemployment Rate Over Time",
       subtitle = "This is some additional information")
```


Unemployment Rate Over Time

This is some additional information

Last thing for today

Faceting

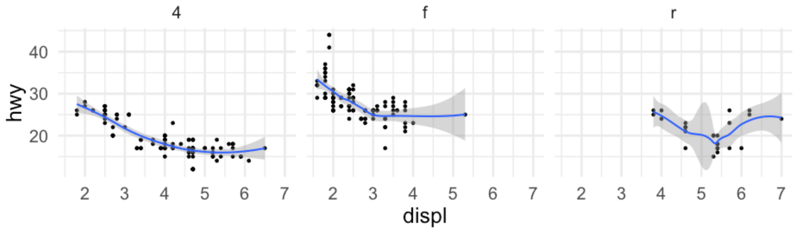
- One of the most powerful features of ggplot, from my perspective
- Produce n plots **by** a specific variable

Last thing for today

Faceting

- One of the most powerful features of ggplot, from my perspective
- Produce n plots **by** a specific variable

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point() +
  geom_smooth() +
   facet_wrap(~drv)
```



Careful about ~

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point() +
  geom_smooth() +
   facet_wrap(drv)
```

```
## Error in as_facets_list(facets): object 'drv' not found
```

Other features

To be covered more in the future

- Colors
- Legends
- Fills
- Other geoms
- Categorical data
- etc.

Lab