Visualize data with ggplot2

Cars and displacement

What is the relationship between a car's engine weight (displacement) and its mileage (miles per gallon)?

Your turn #1

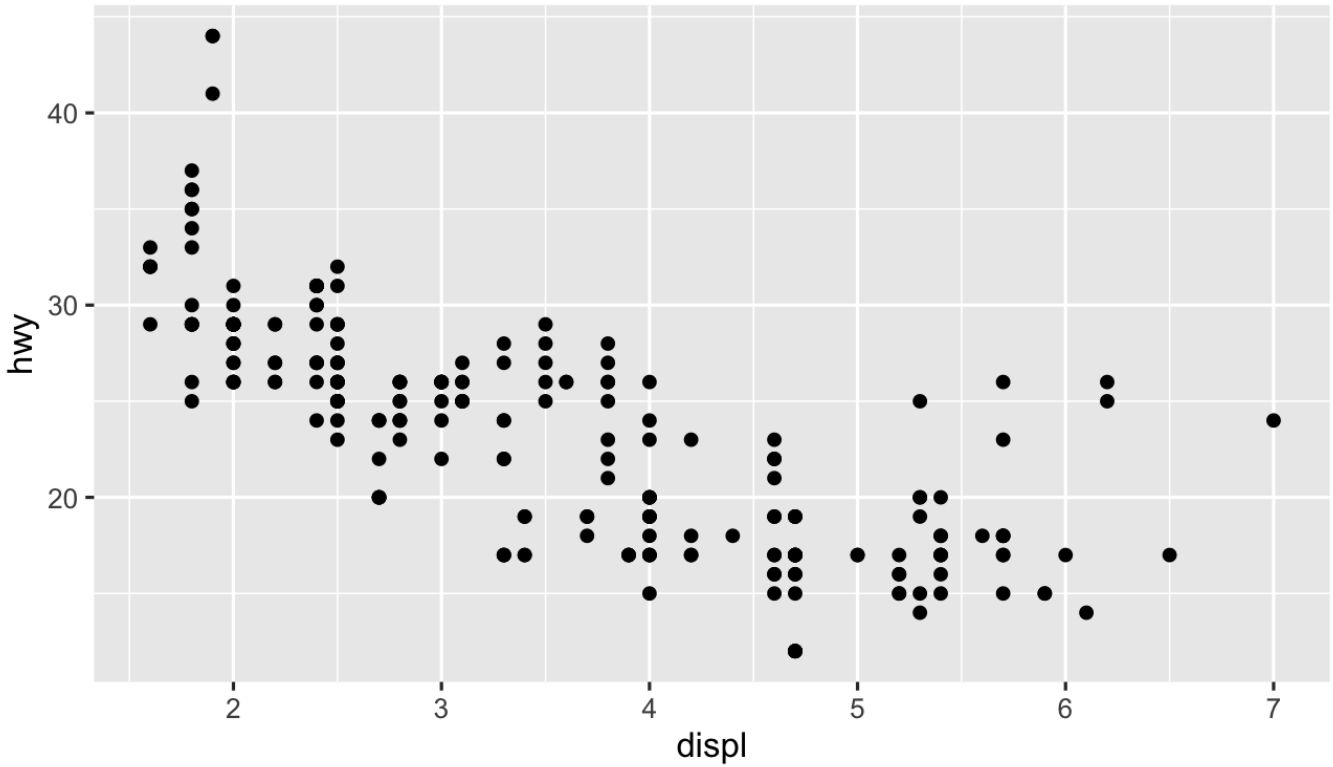
Run this code in your Quarto file to make a graph

Pay attention to spelling, capitalization, and parentheses!

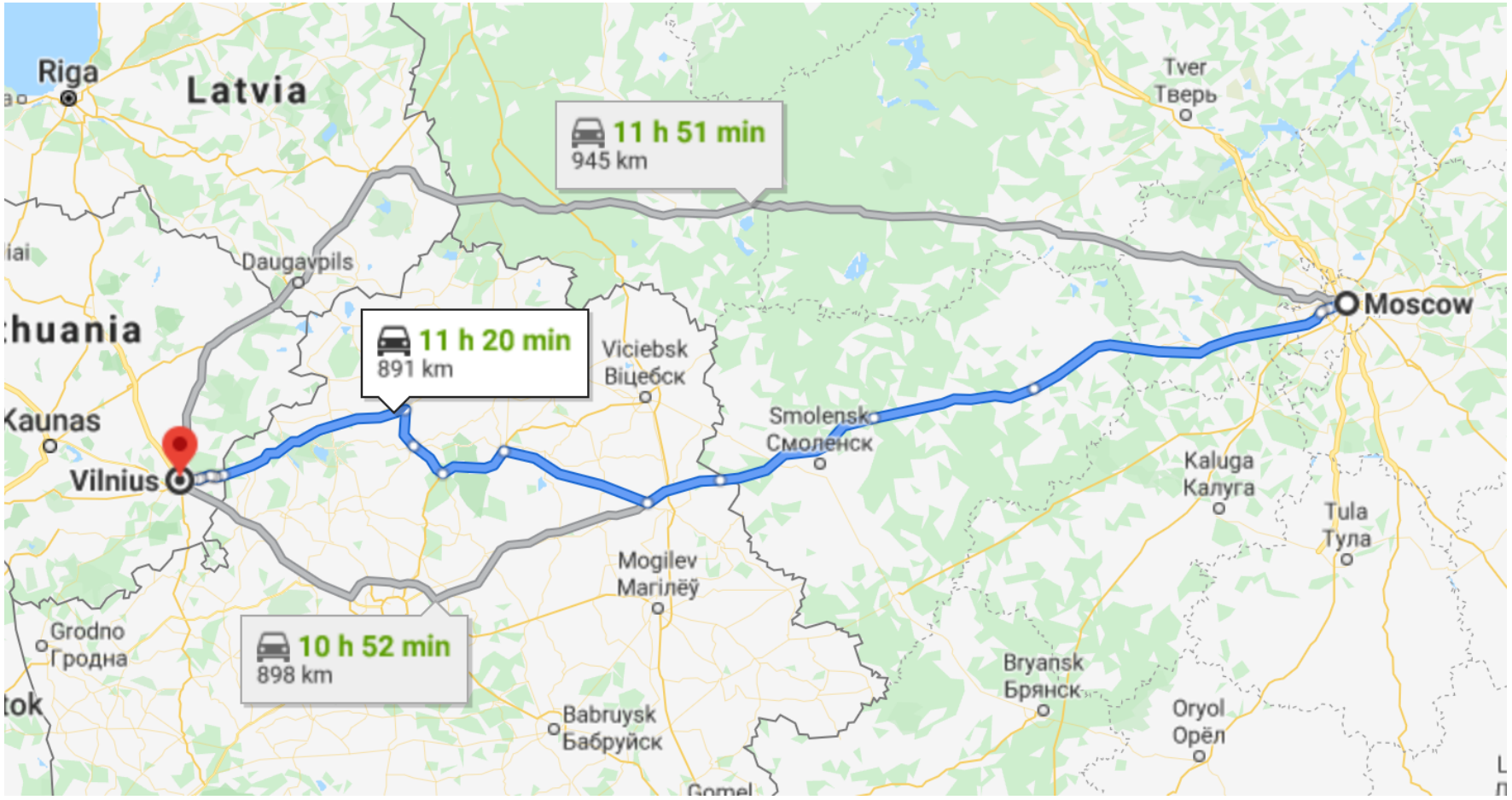
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy))
```

01:00

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

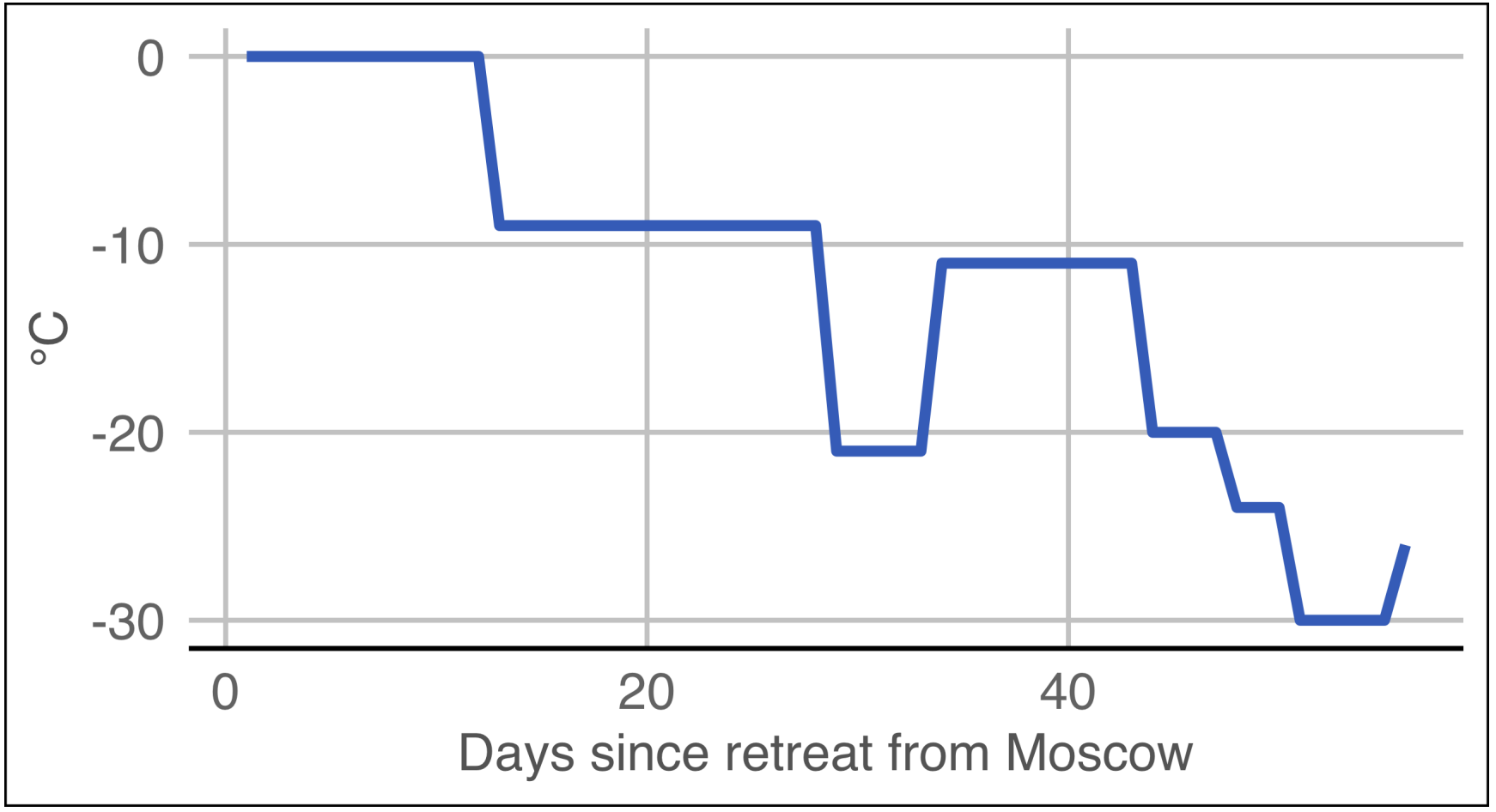


Long distance!

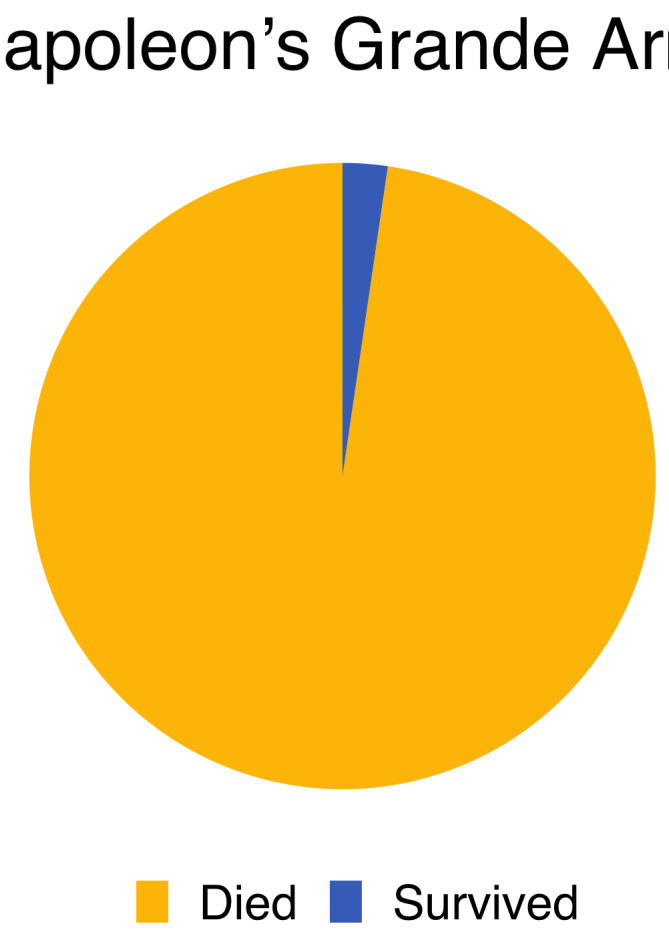


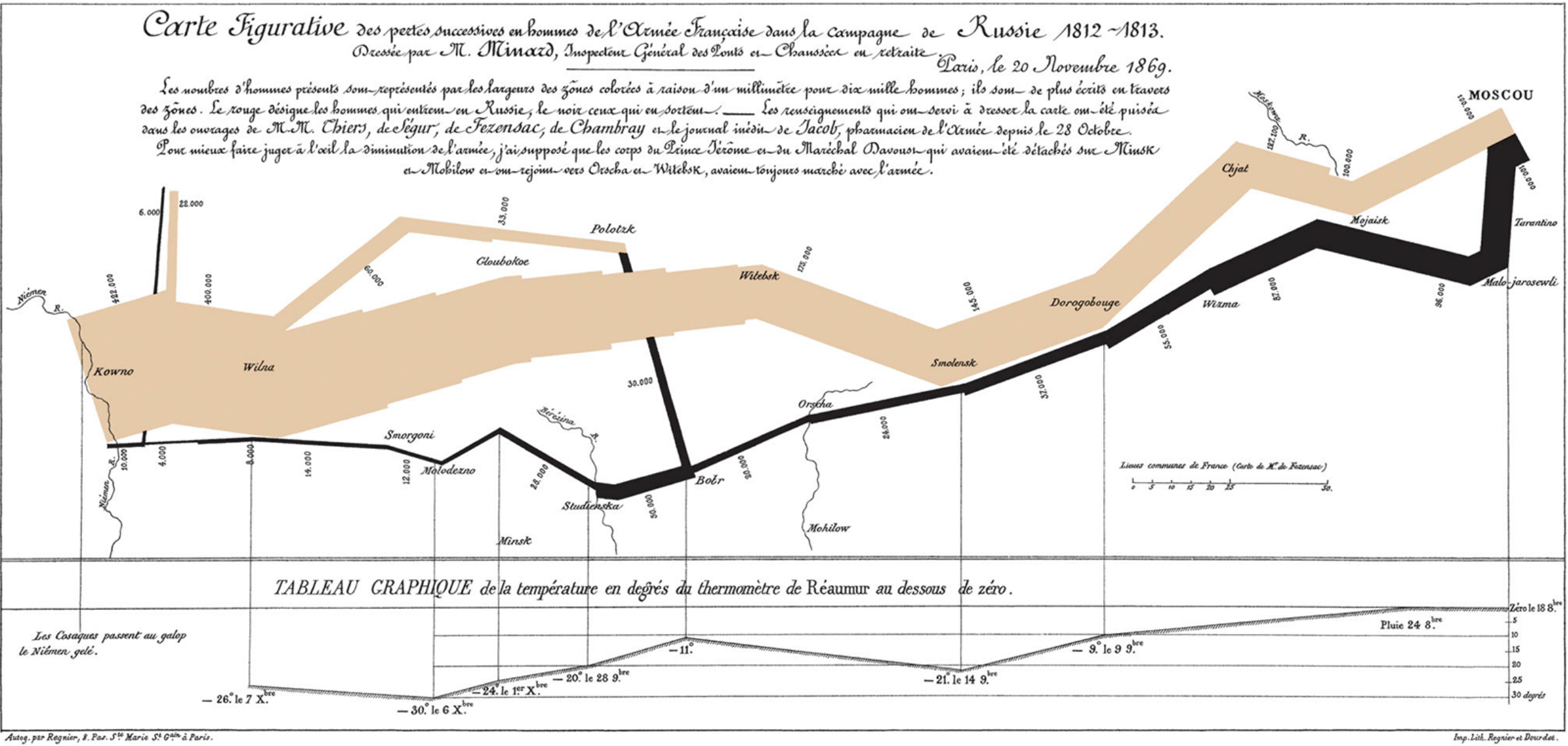
Moscow to Vilnius

Very cold!

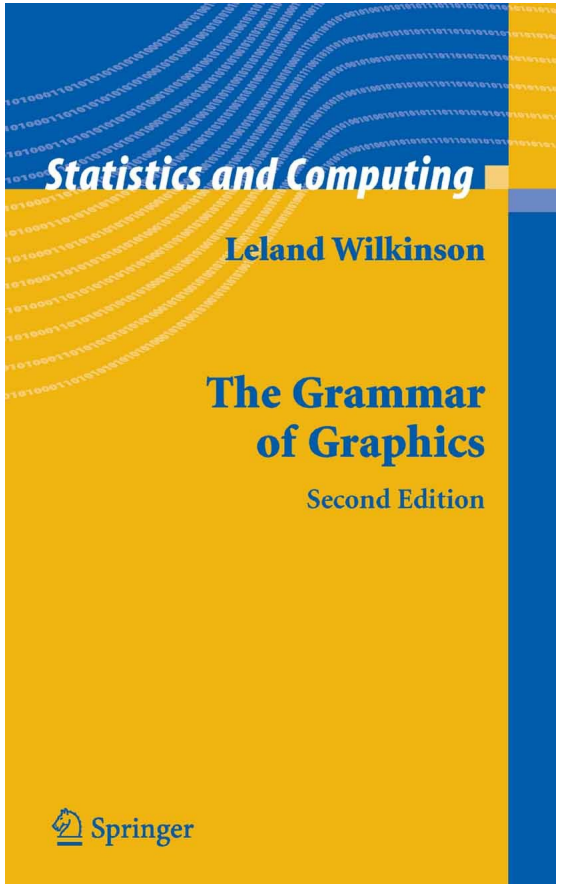


Lots of people died!




Carte Figurative des pertes successives en hommes de l'Armée Française dans la Campagne de Russie 1812-1813.
Dressée par M. Minard, Inspecteur Général des Ponts et Chaussées en retraite.
Paris, le 20 Novembre 1869.
Les nombres d'hommes présents sont représentés par les largeurs des zônes colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zônes. Le rouge désigne les hommes qui entrent en Russie, le noir ceux qui en sortent. ___ Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M.M. Thiers, de Ségur, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre.
Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davoust qui avaient été détachés sur Minsk et Mobilow et ont rejoint vers Orscha et Witebsk, avaient toujours marché avec l'armée.
Niémen R.
Kowno
Wilna
Smorgoni
Molodexno
Minsk
Studienska
Bobr
Polotzk
Gloubokoe
Witebsk
Orscha
Mohilow
Smolensk
Dorogobouge
Wixma
Chjat
Mojaisk
MOSCOU
Tarantino
Malo-jarosewli
Wop R.
Kastrony R.
Mérézina R.
Lieues communes de France (Carte de M. de Fezensac)
TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.
Les Cosaques passent au galop le Niémen gelé.
— 26.° le 7 X.bre
— 30.° le 6 X.bre
— 24.° le 1er X.bre
— 20.° le 28 9.bre
— 11.°
— 21.° le 14 9.bre
— 9.° le 9 9.bre
Pluie 24 8.bre
Zéro le 18 8.bre
5
10
15
20
25
30 degrés
Autog. par Regnier, 8. Pas. Ste Marie St Gdin à Paris.
Imp. Lith. Regnier et Dourdet.

Mapping data to aesthetics



Aesthetic

Visual property of a graph

Position, shape, color, etc.

Data

A column in a dataset

Mapping data to aesthetics

Data	Aesthetic	Graphic/Geometry
Longitude	Position (x-axis)	Point
Latitude	Position (y-axis)	Point
Army size	Size	Path
Army direction	Color	Path
Date	Position (x-axis)	Line + text
Temperature	Position (y-axis)	Line + text

Mapping data to aesthetics

Data	**`aes()`**	**`geom`**
Longitude	`x`	`geom_point()`
Latitude	`y`	`geom_point()`
Army size	`size`	`geom_path()`
Army direction	`color`	`geom_path()`
Date	`x`	`geom_line() + geom_text()`
Temperature	`y`	`geom_line() + geom_text()`

`ggplot()` template

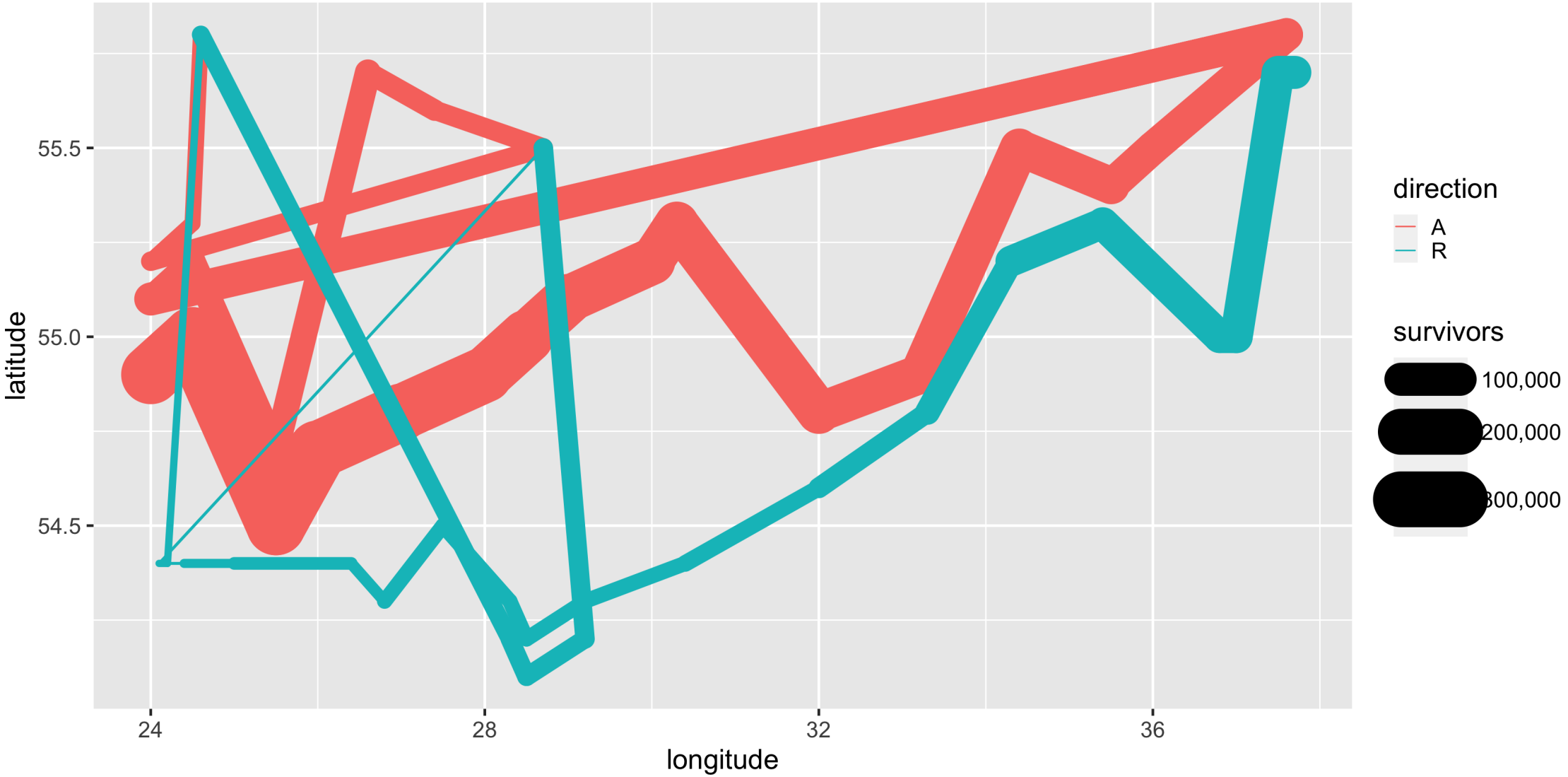
```
ggplot(data = DATA) +
  GEOM_FUNCTION(mapping = aes(AESTHETIC MAPPINGS))
```

```
ggplot(data = troops) +
  geom_path(mapping = aes(x = longitude,
                          y = latitude,
                          color = direction,
                          size = survivors))
```

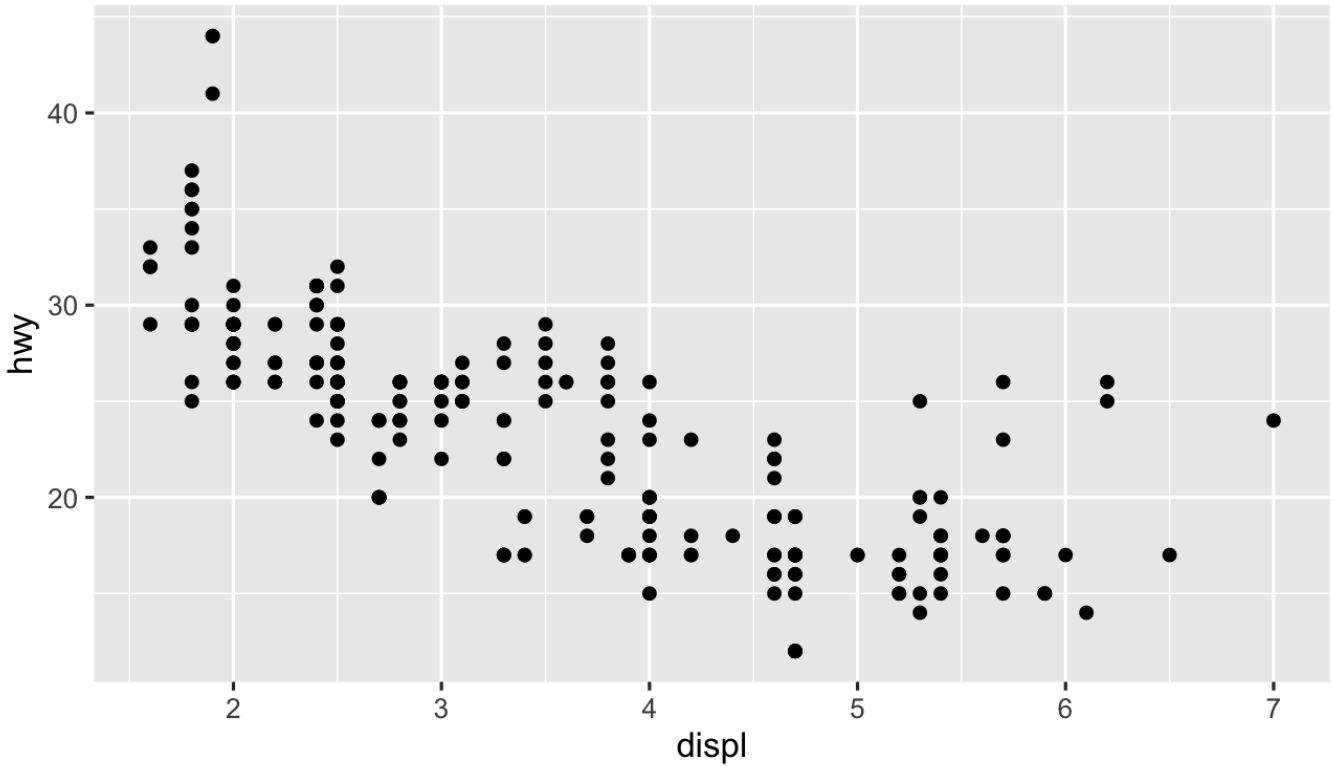
This is a dataset named `troops`:

longitude	latitude	direction	survivors
24	54.9	A	340000
24.5	55	A	340000
…	…	…	…

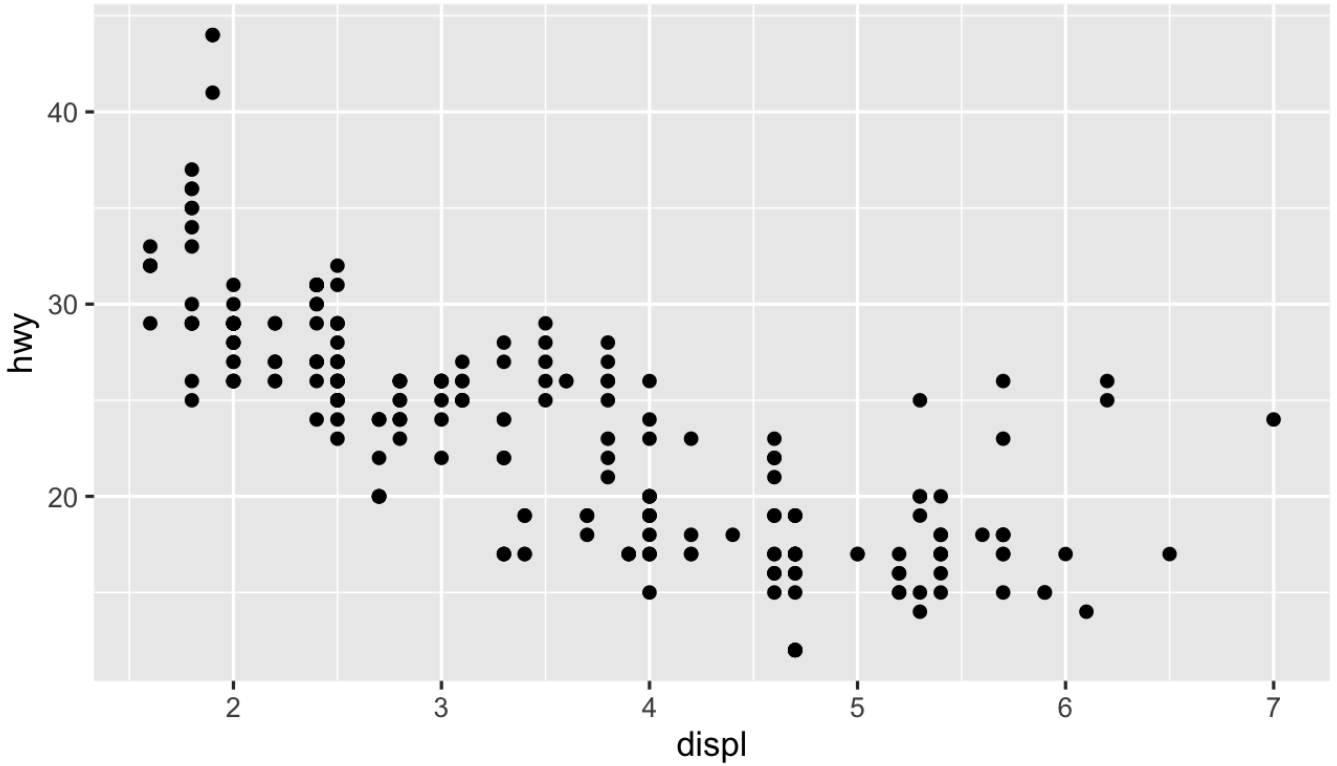
```
ggplot(data = troops) +
  geom_path(mapping = aes(x = longitude,
                          y = latitude,
                          color = direction,
                          size = survivors))
```


55.5
55.0
54.5
latitude
24
28
32
36
longitude
direction
A
R
survivors
100,000
200,000
300,000

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

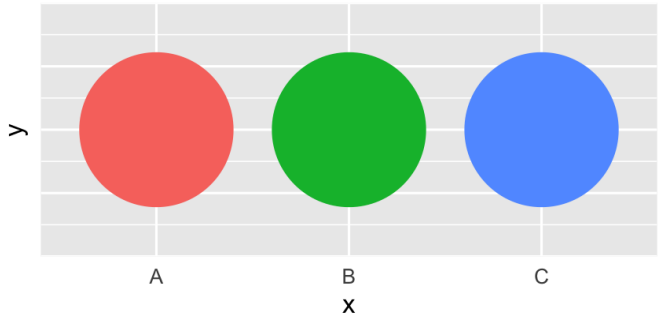


Heavy cars with better mileage?

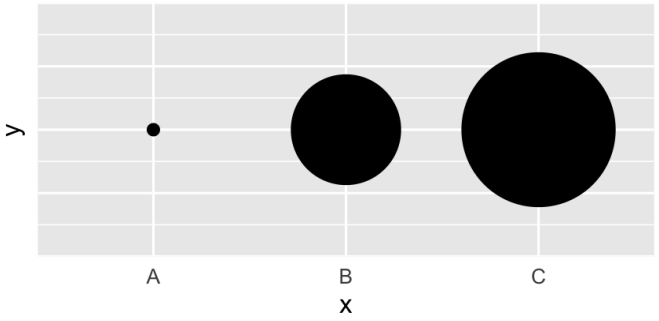


Aesthetics

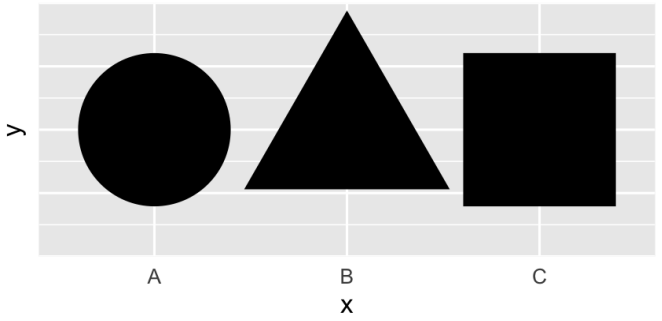
`color` (discrete)



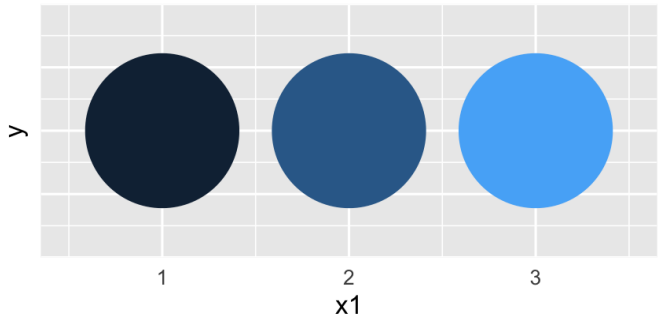
`size`



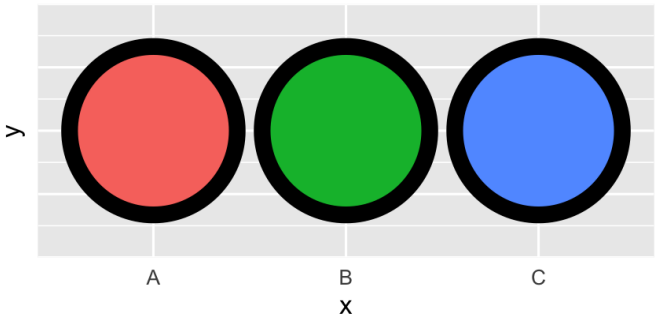
`shape`



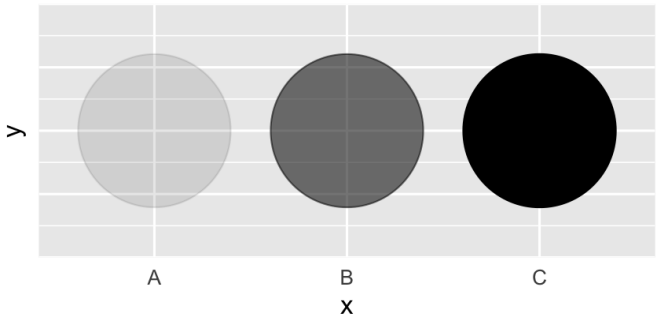
`color` (continuous)



`fill`



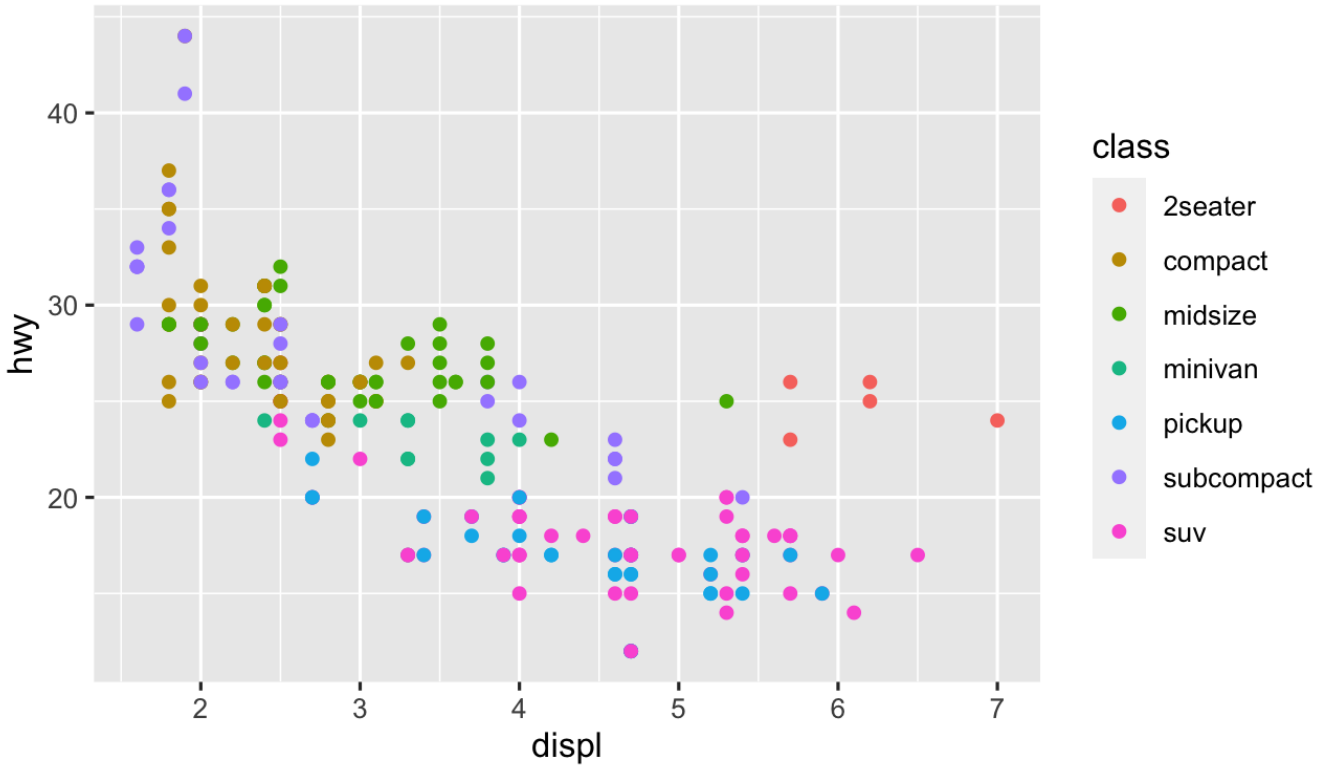
`alpha`



Mapping columns to aesthetics

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

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



Your turn #2

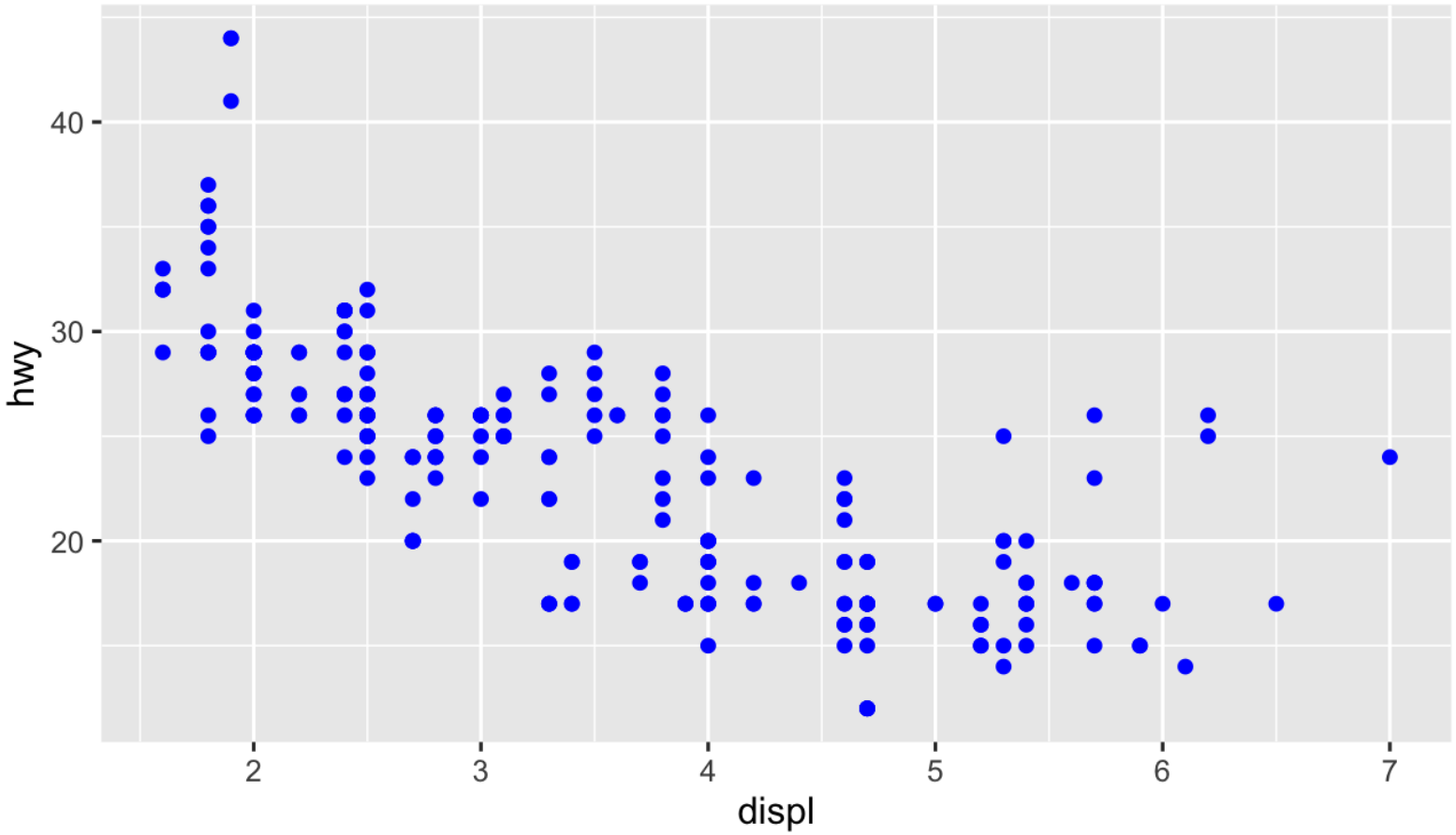
Add color, size, alpha, and shape aesthetics to your graph.

Experiment!

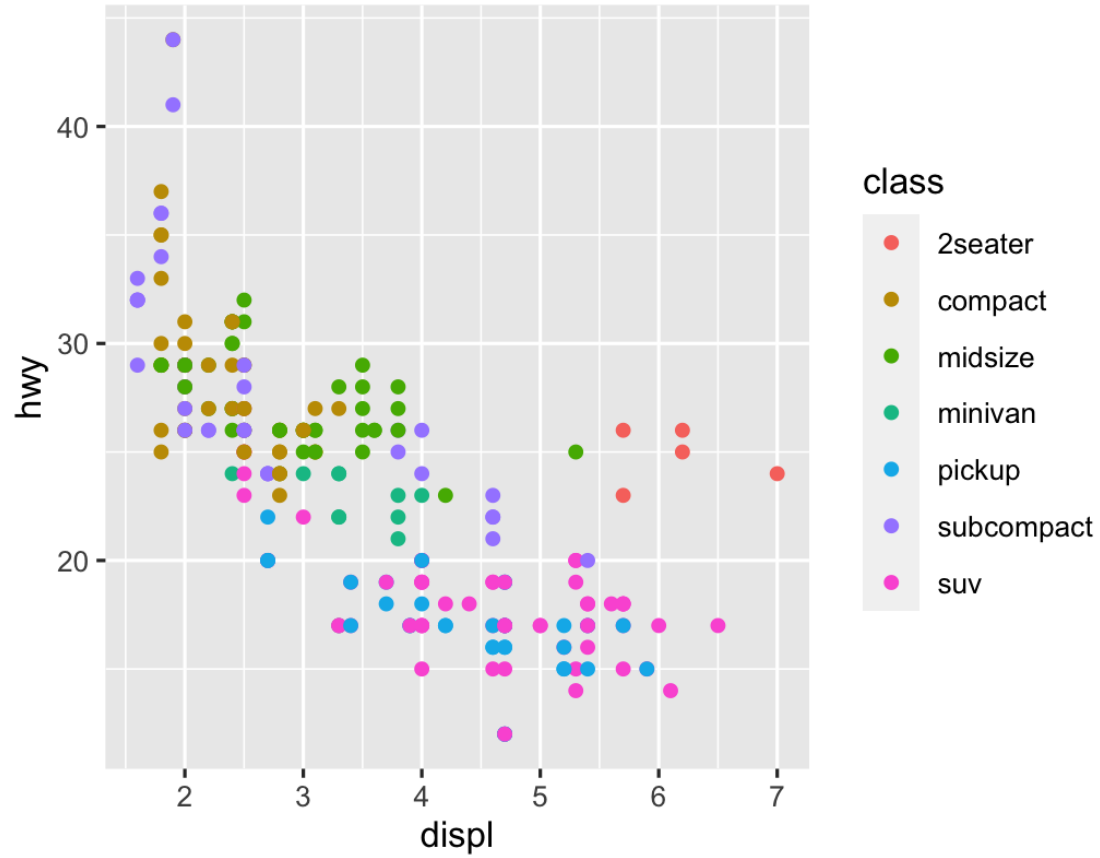
Do different things happen when you map aesthetics to discrete and continuous variables?

What happens when you use more than one aesthetic?

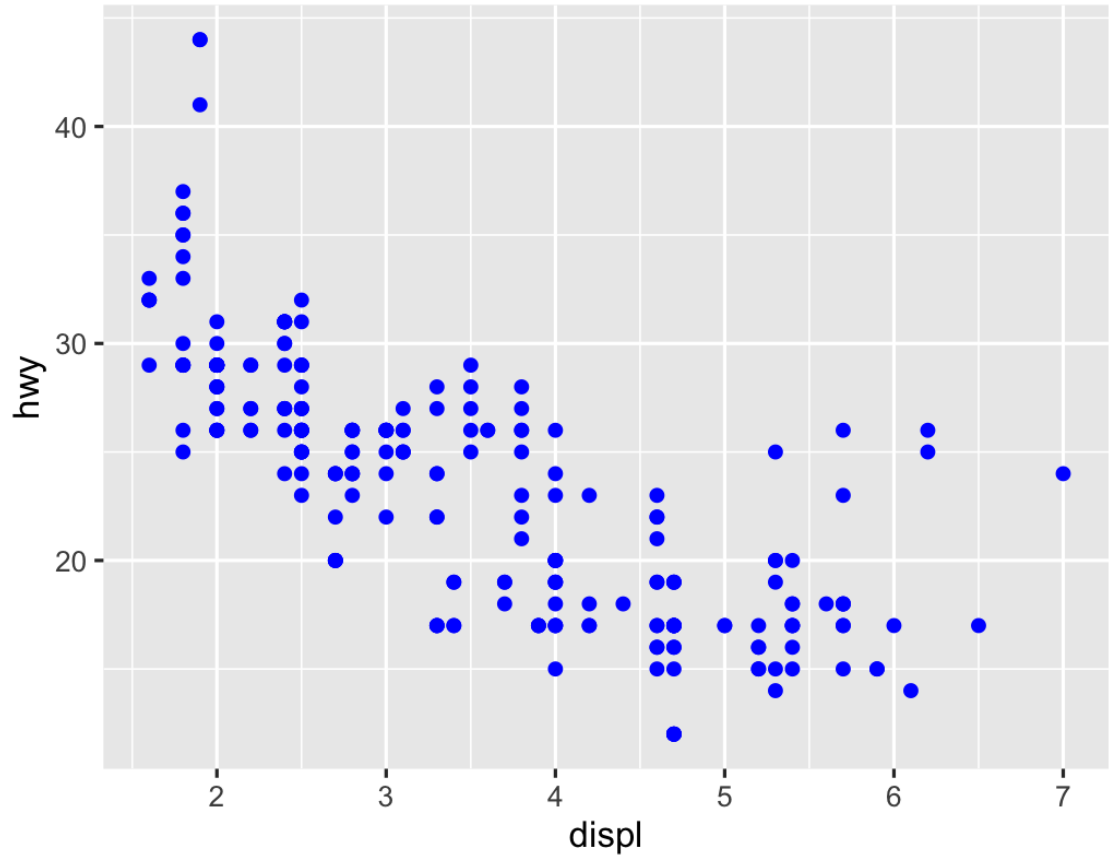
How would you make this plot?



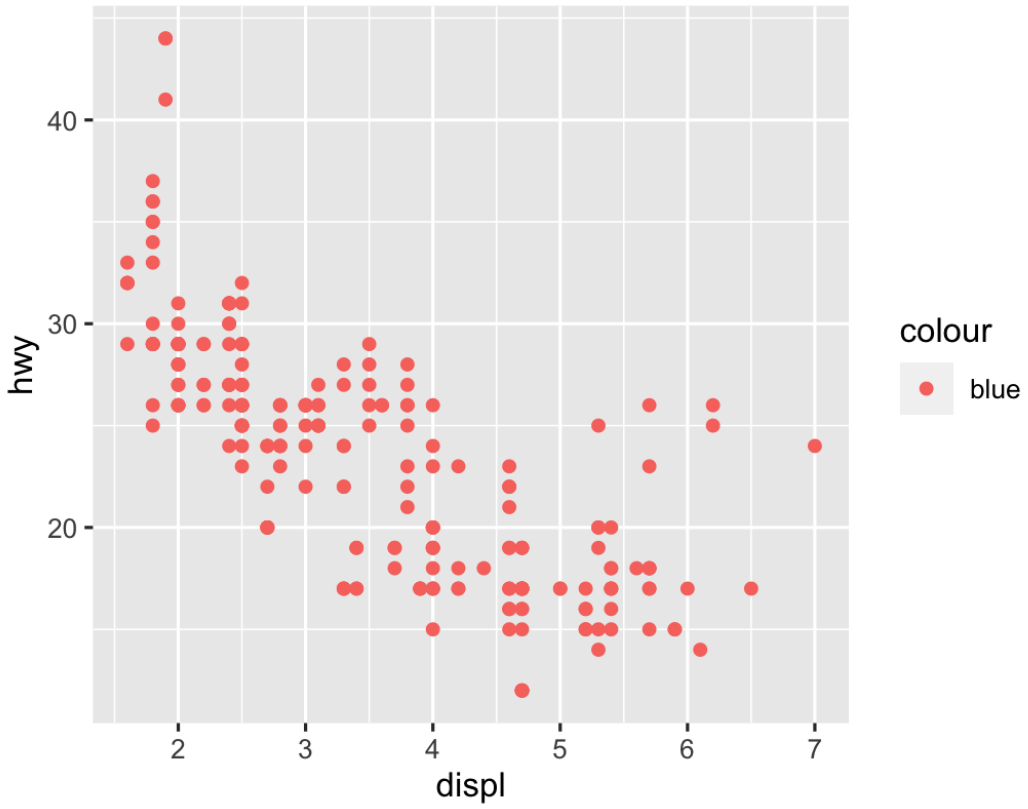
```
ggplot(mpg) +
  geom_point(aes(x = displ, y = hwy,
                 color = class))
```



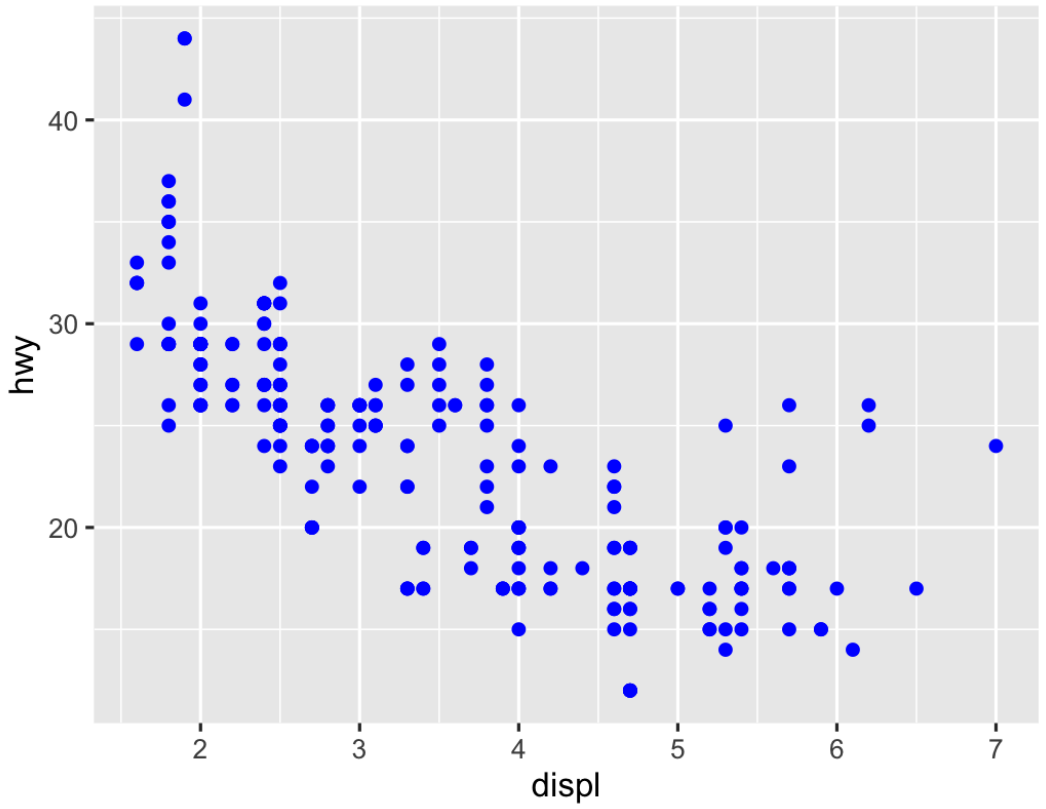
```
ggplot(mpg) +
  geom_point(aes(x = displ, y = hwy),
             color = "blue")
```



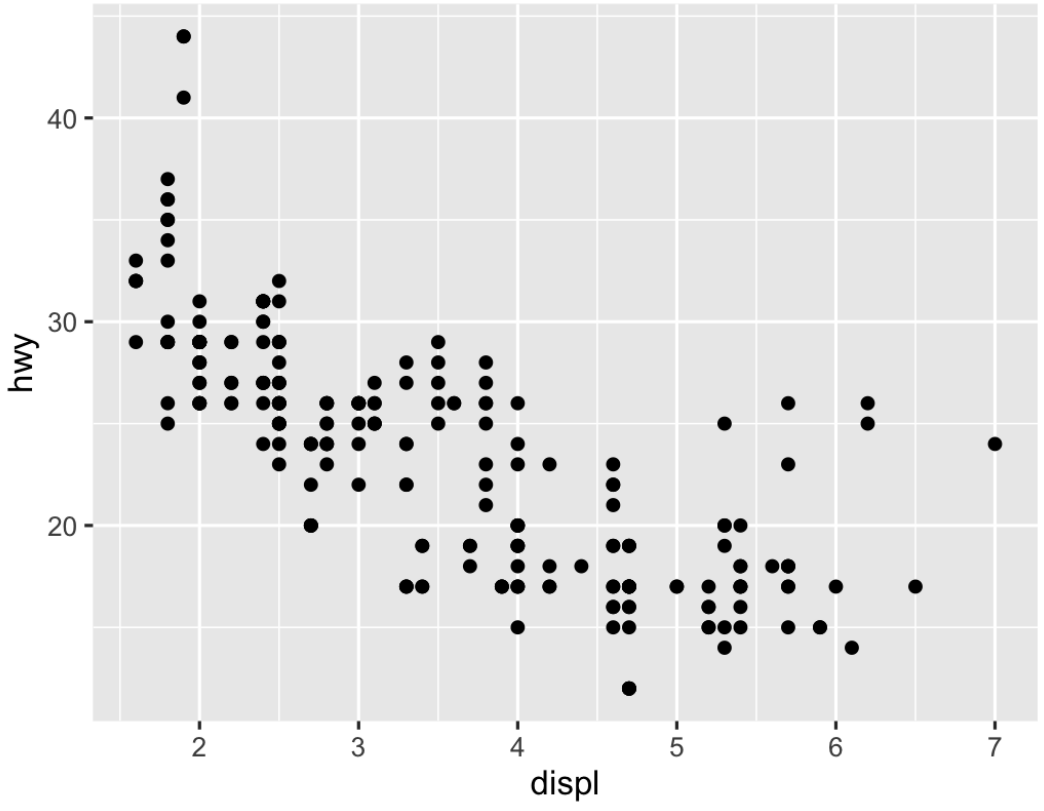
```
ggplot(mpg) +
  geom_point(aes(x = displ, y = hwy,
                 color = "blue"))
```

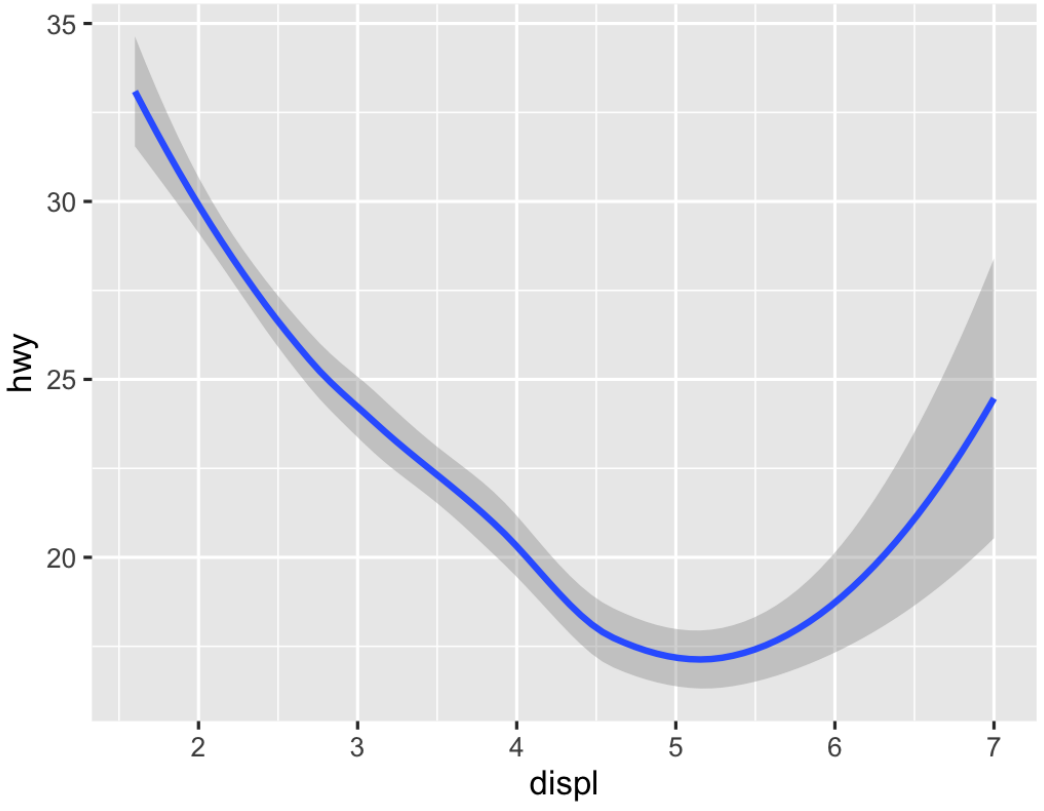



```
ggplot(mpg) +
  geom_point(aes(x = displ, y = hwy)
             color = "blue")
```



What's the same? What's different?

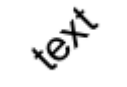




Geoms

```
ggplot(data = DATA) +
  GEOM_FUNCTION(mapping = aes(AESTHETIC MAPPINGS))
```

Possible geoms

	Example geom	What it makes
	`geom_col()`	Bar charts
	`geom_text()`	Text
	`geom_point()`	Points
	`geom_boxplot()`	Boxplots
	`geom_sf()`	Maps

Possible geoms

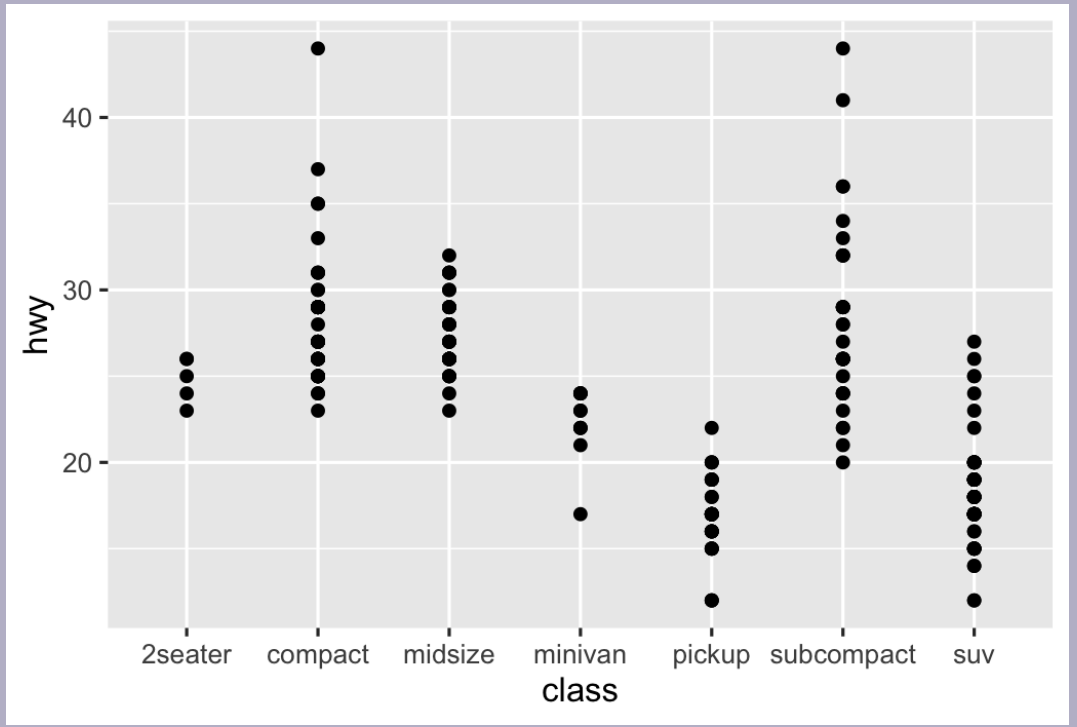
There are dozens of possible geoms!

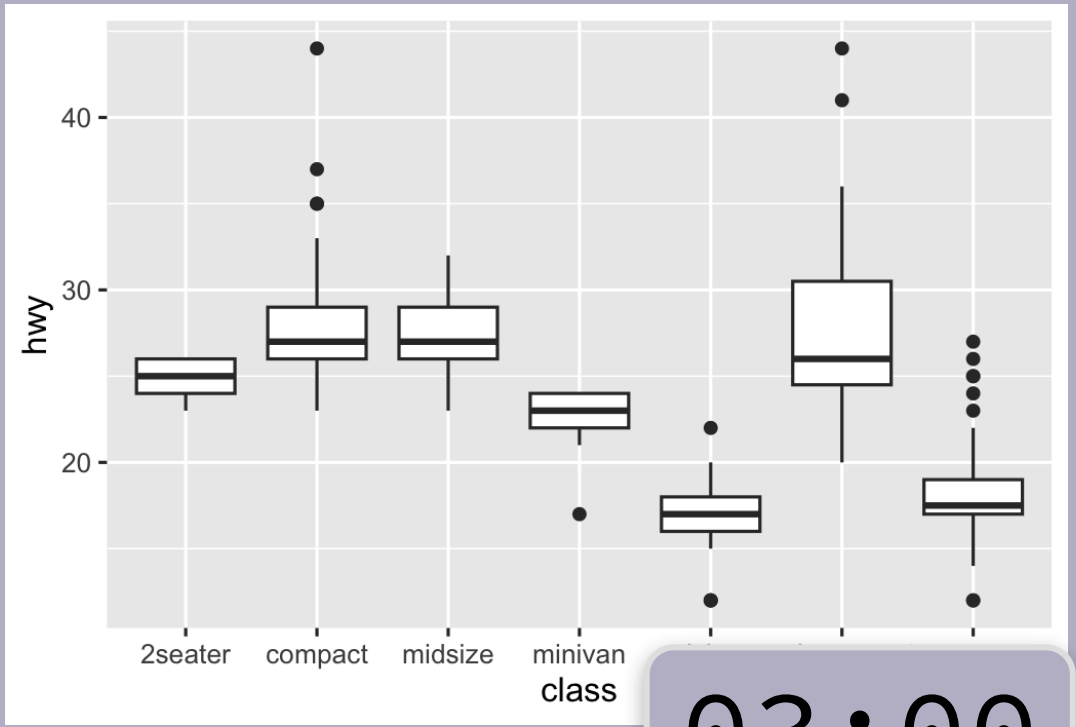
See the ggplot2 documentation for complete examples of all the different geom layers

Also see the ggplot cheatsheet

Your turn #3

Replace this scatterplot with boxplots. Use the cheatsheet.

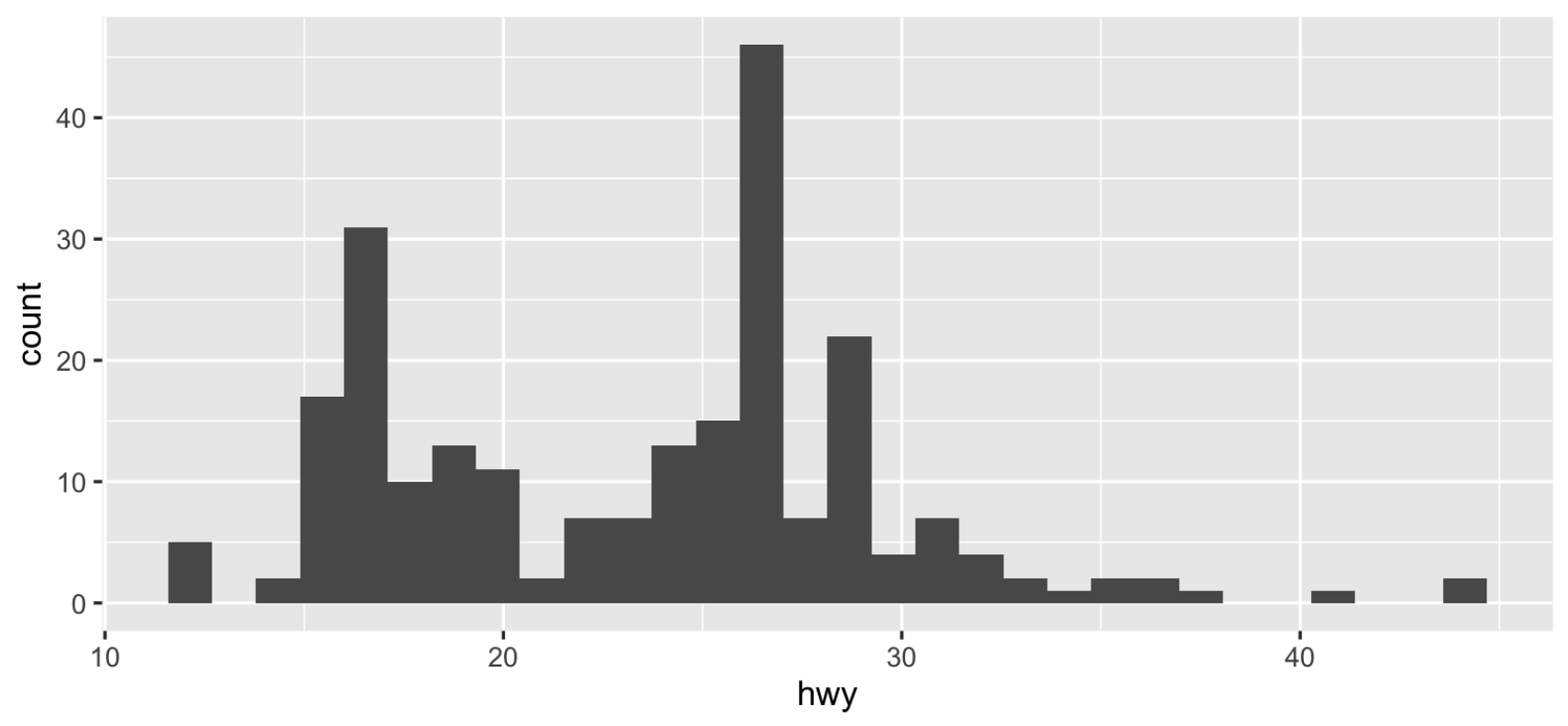




03:00

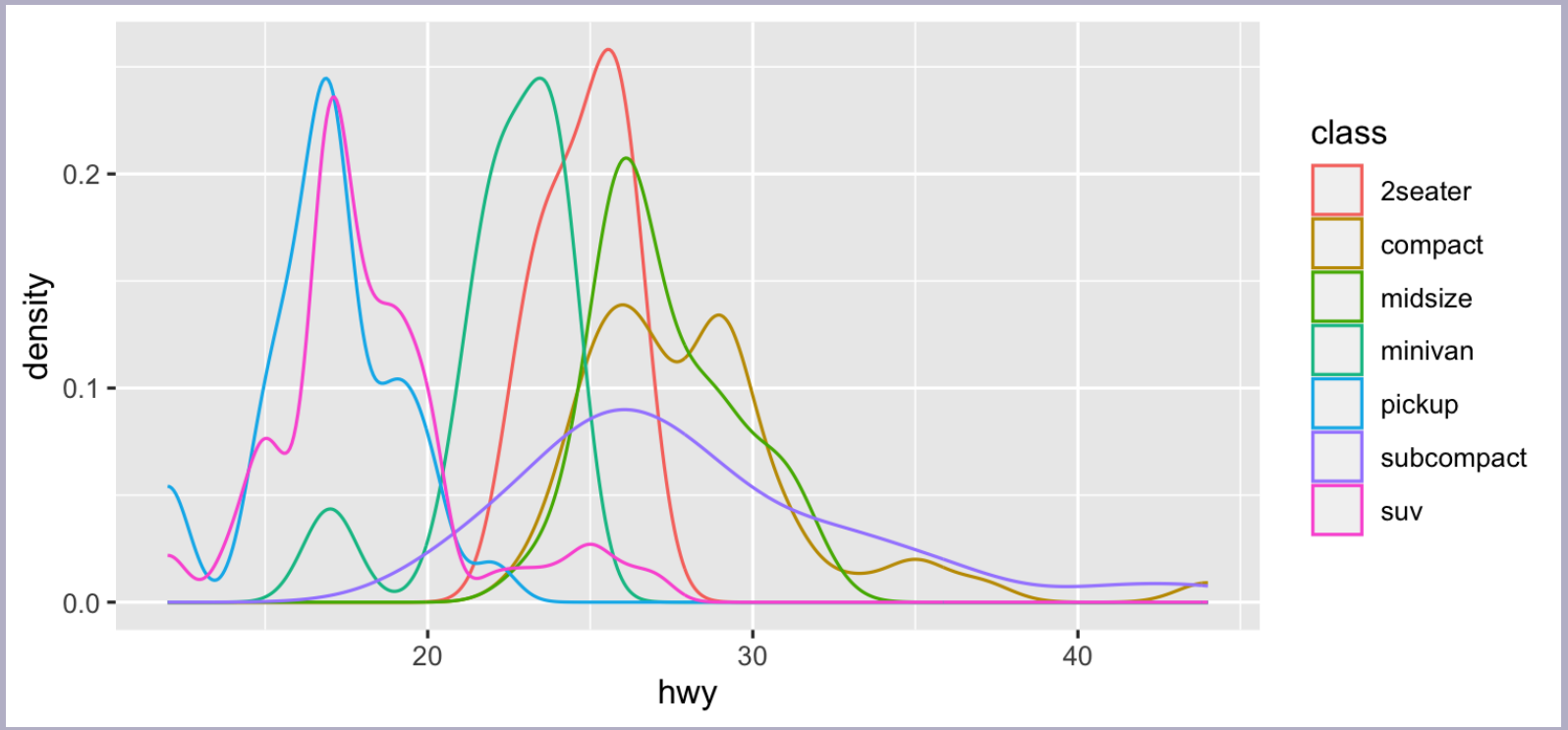
Your turn #4

Make a histogram of `hwy`. Use the cheetsheet. Hint: don't supply a `y` variable.

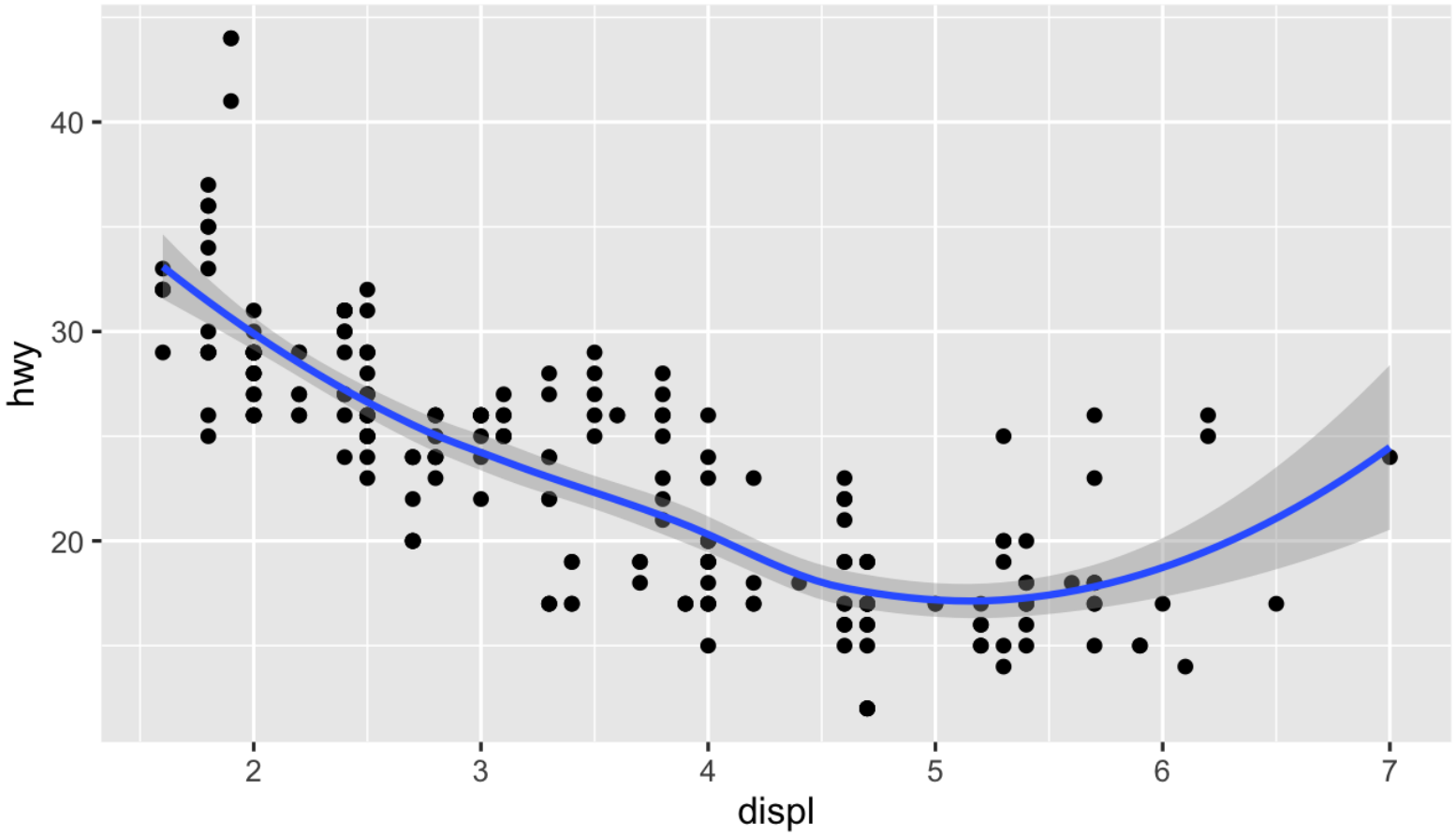


Your turn #5

Make this density plot of `hwy` colored by `class`. Use the cheatsheet. Hint: don't supply a `y` variable.



Complex graphs!



Your turn #6

Predict what this code will do. Then run it.

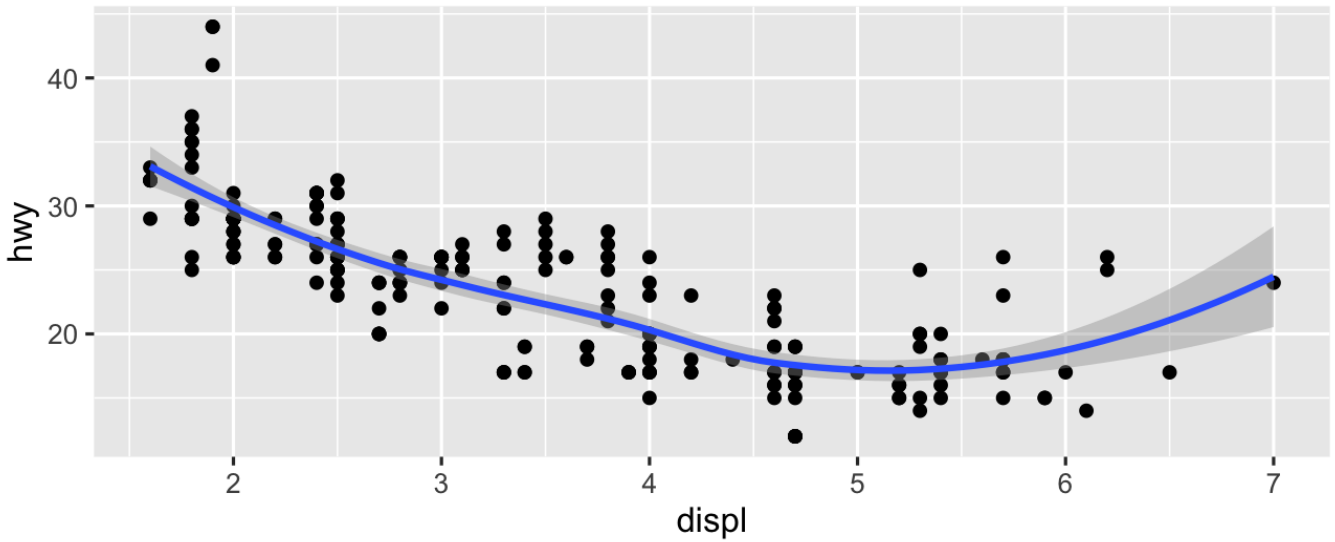
```
ggplot(data = mpg) +
  geom_point(mapping = aes(x = displ, y = hwy)) +
  geom_smooth(mapping = aes(x = displ, y = hwy))
```

02:00

Global vs. local

Any aesthetics in `ggplot()` will show up in all `geom_` layers

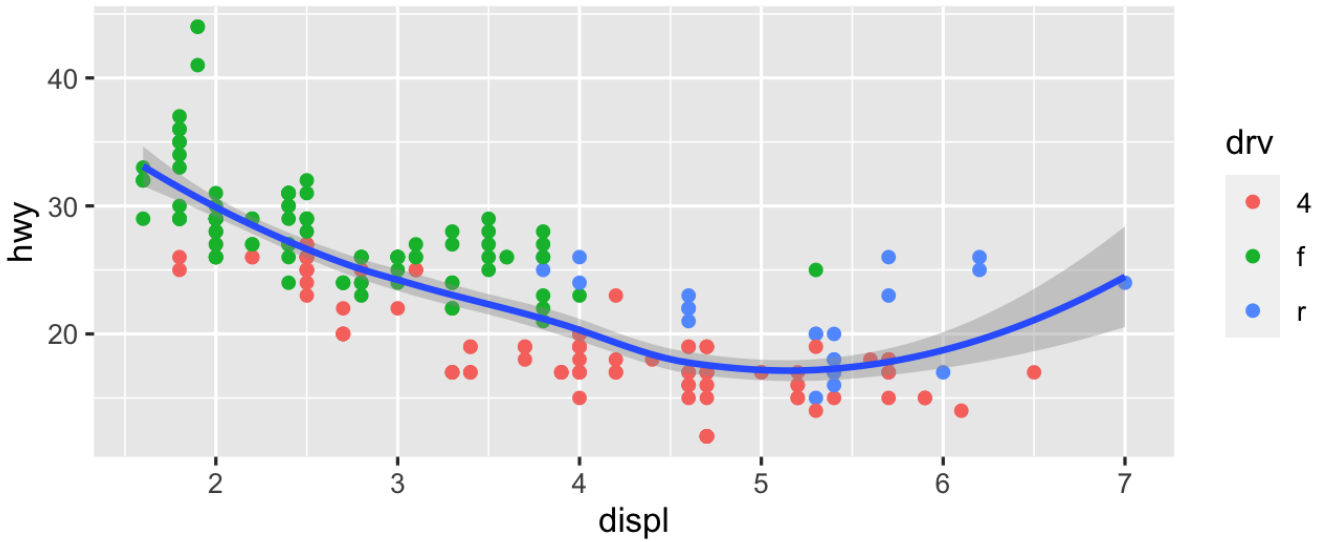
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point() +
  geom_smooth()
```



Global vs. local

Any aesthetics in `geom_` layers only apply to that layer

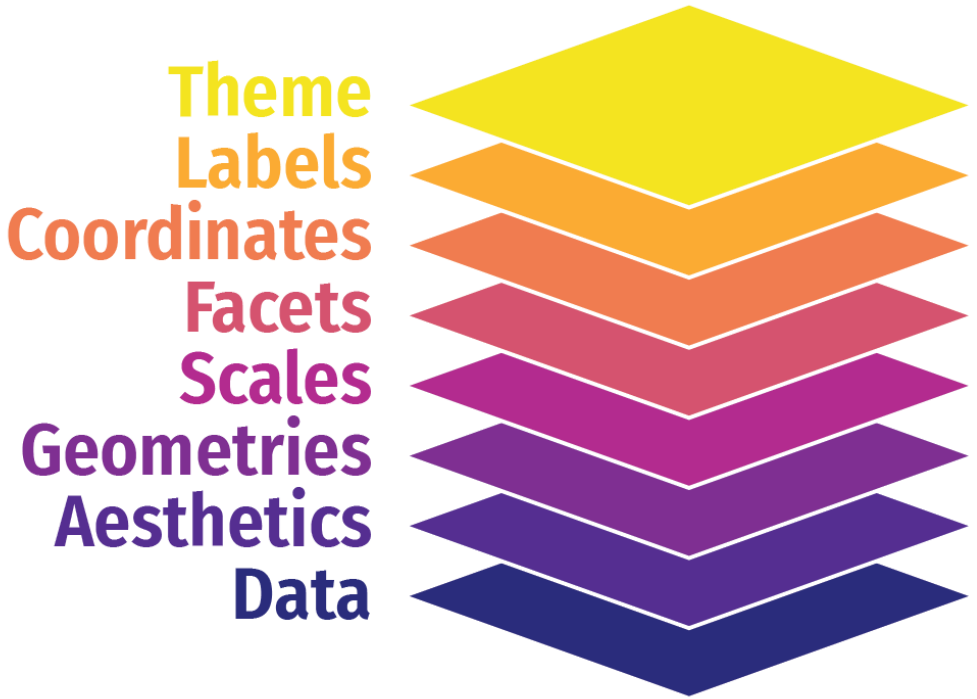
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +
  geom_point(mapping = aes(color = drv)) +
  geom_smooth()
```



So much more!

There are many other layers we can use to make and enhance graphs!

We sequentially add layers onto the foundational `ggplot()` plot to create complex figures

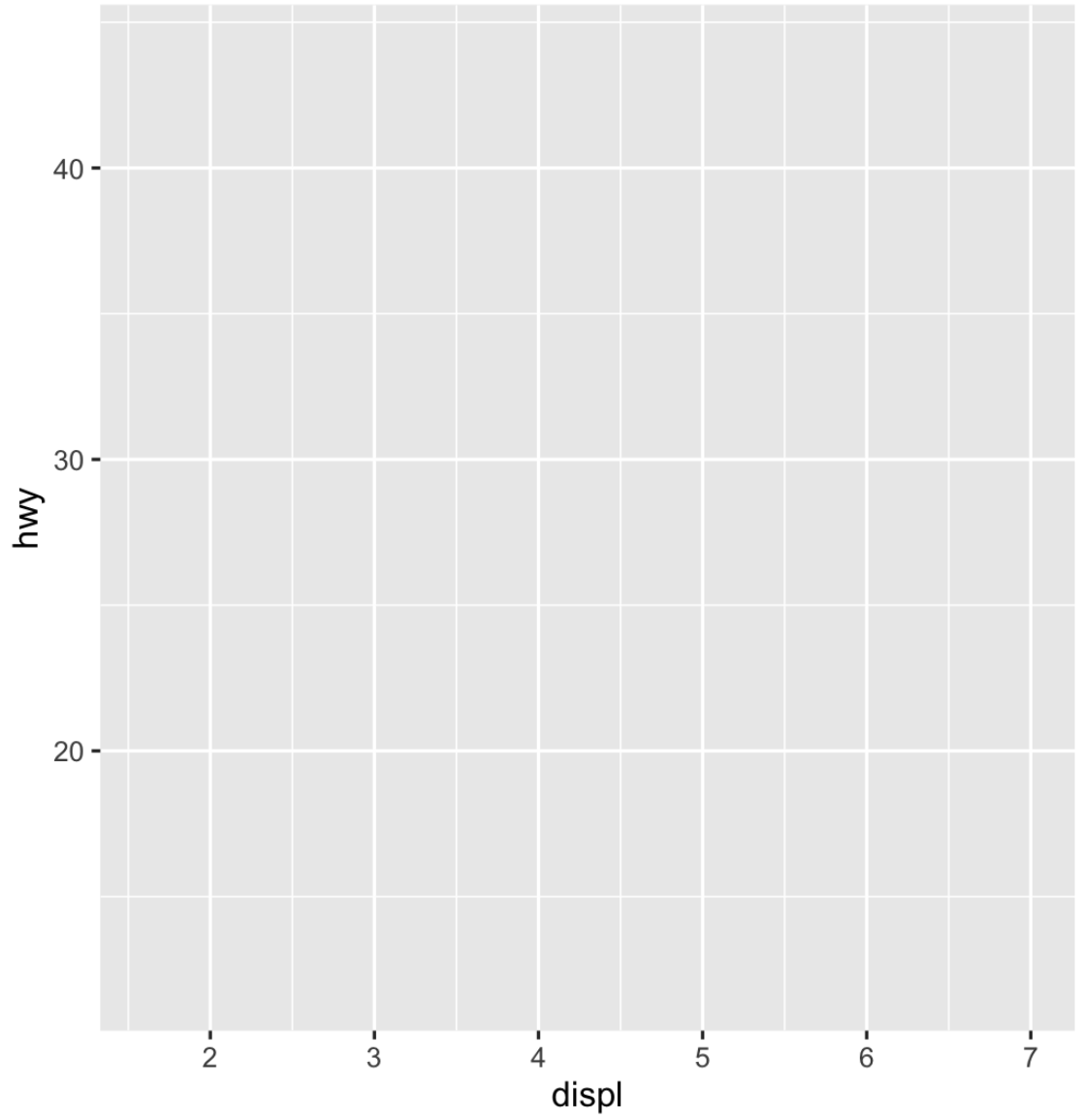


Putting it all together

We can build a plot sequentially to see how each grammatical layer changes the appearance

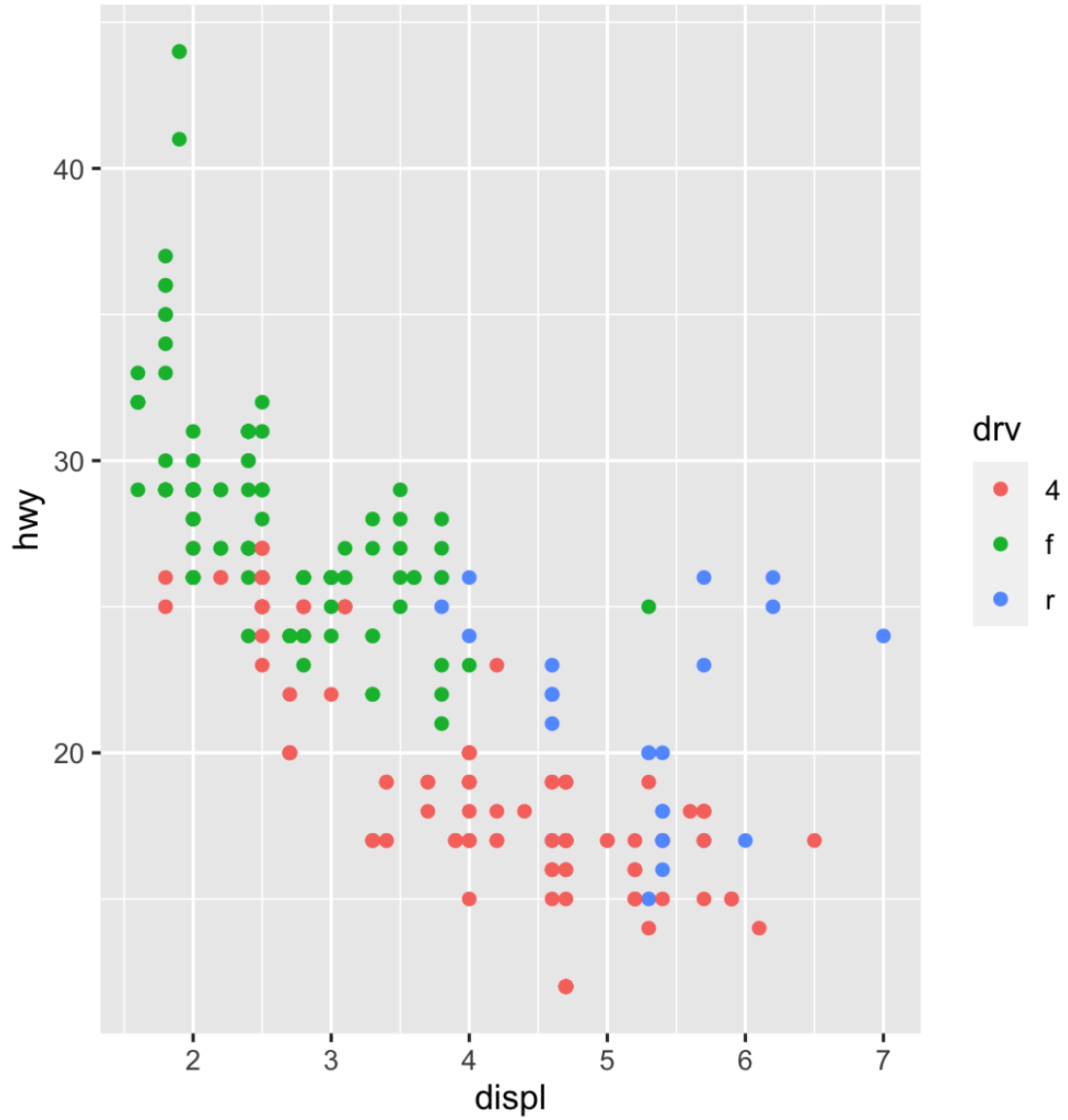
Start with data and aesthetics

```
ggplot(data = mpg,
       mapping = aes(x = displ,
                     y = hwy,
                     color = drv))
```



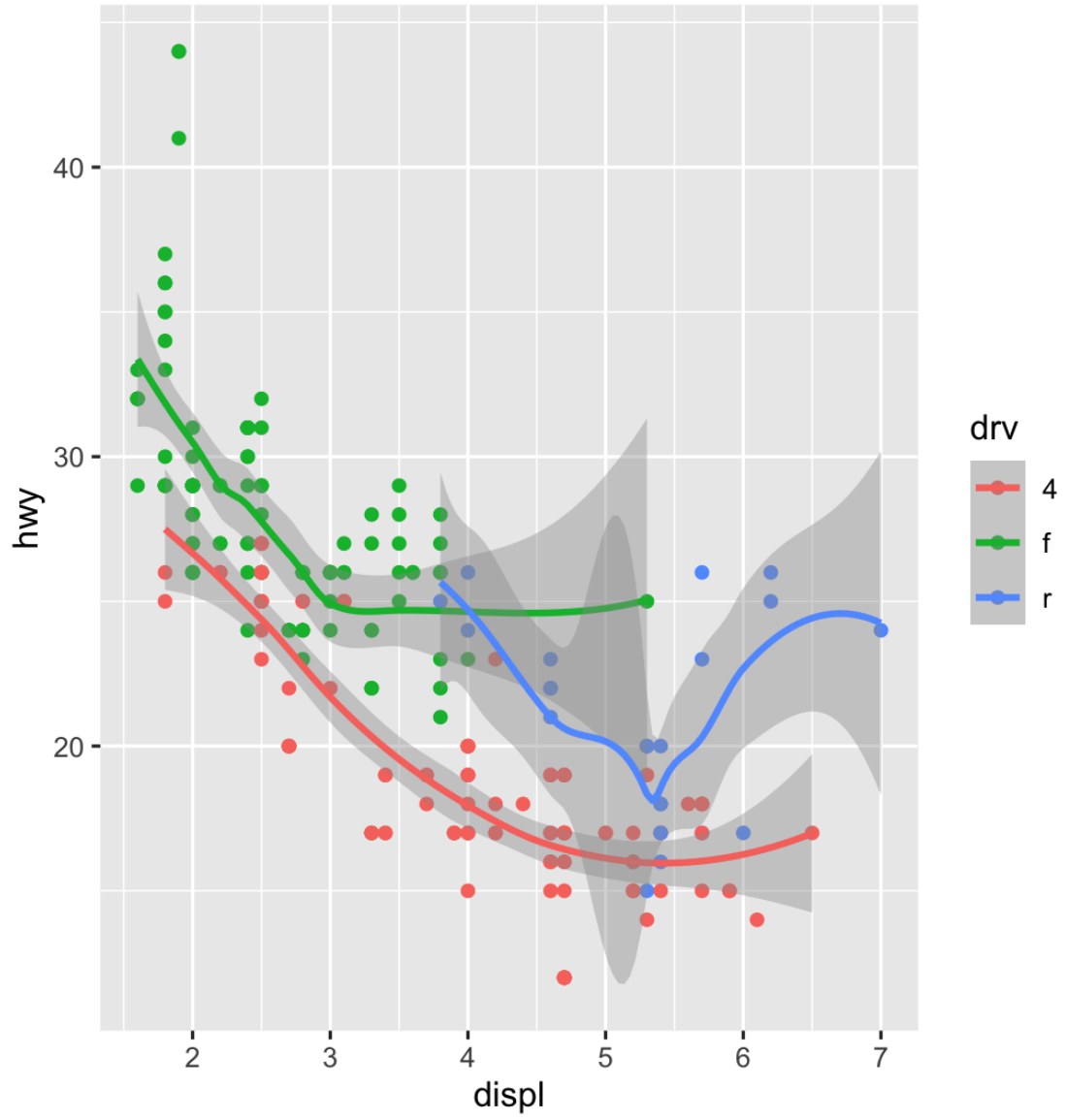
Add a point geom

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



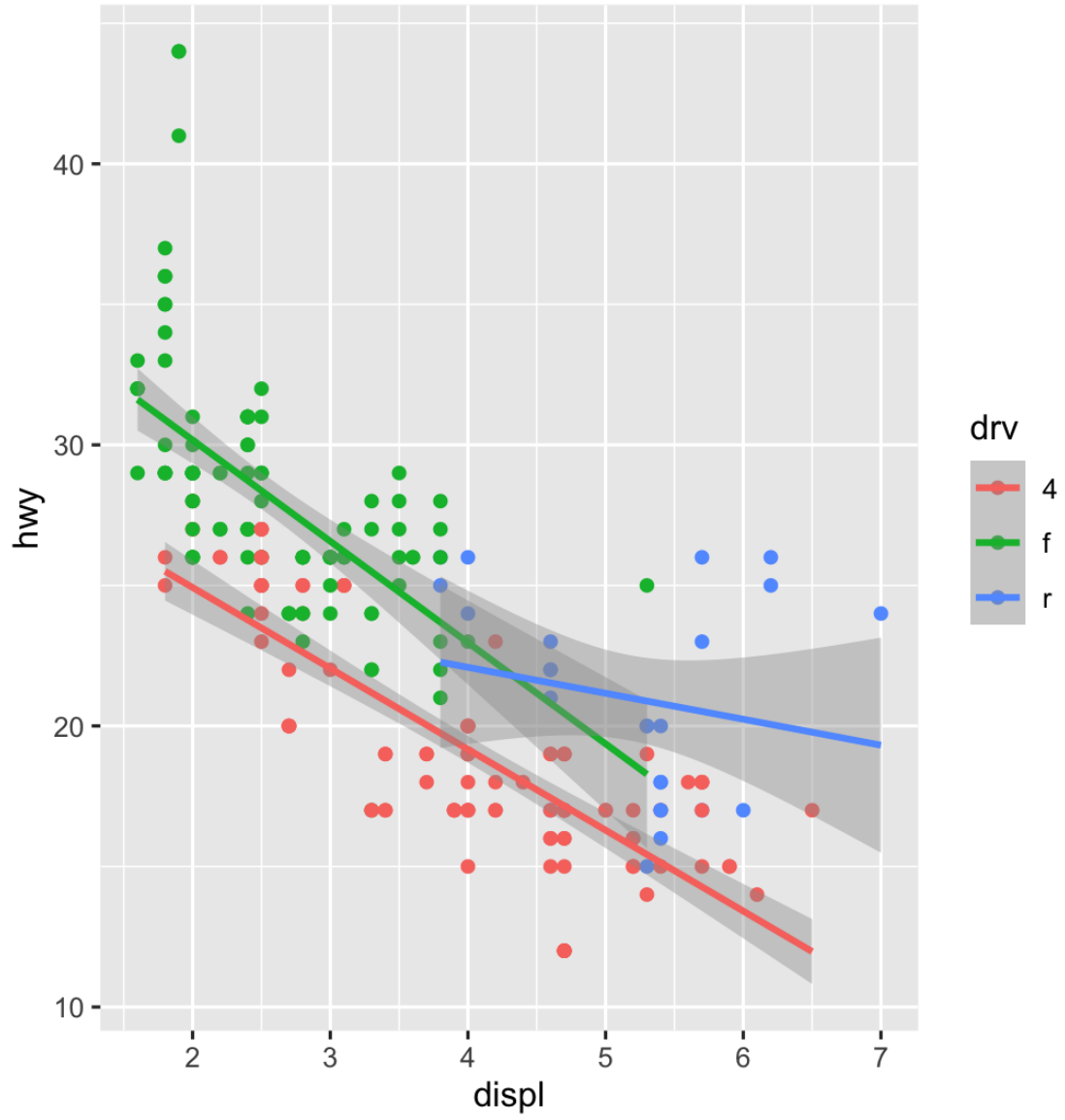
Add a smooth geom

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



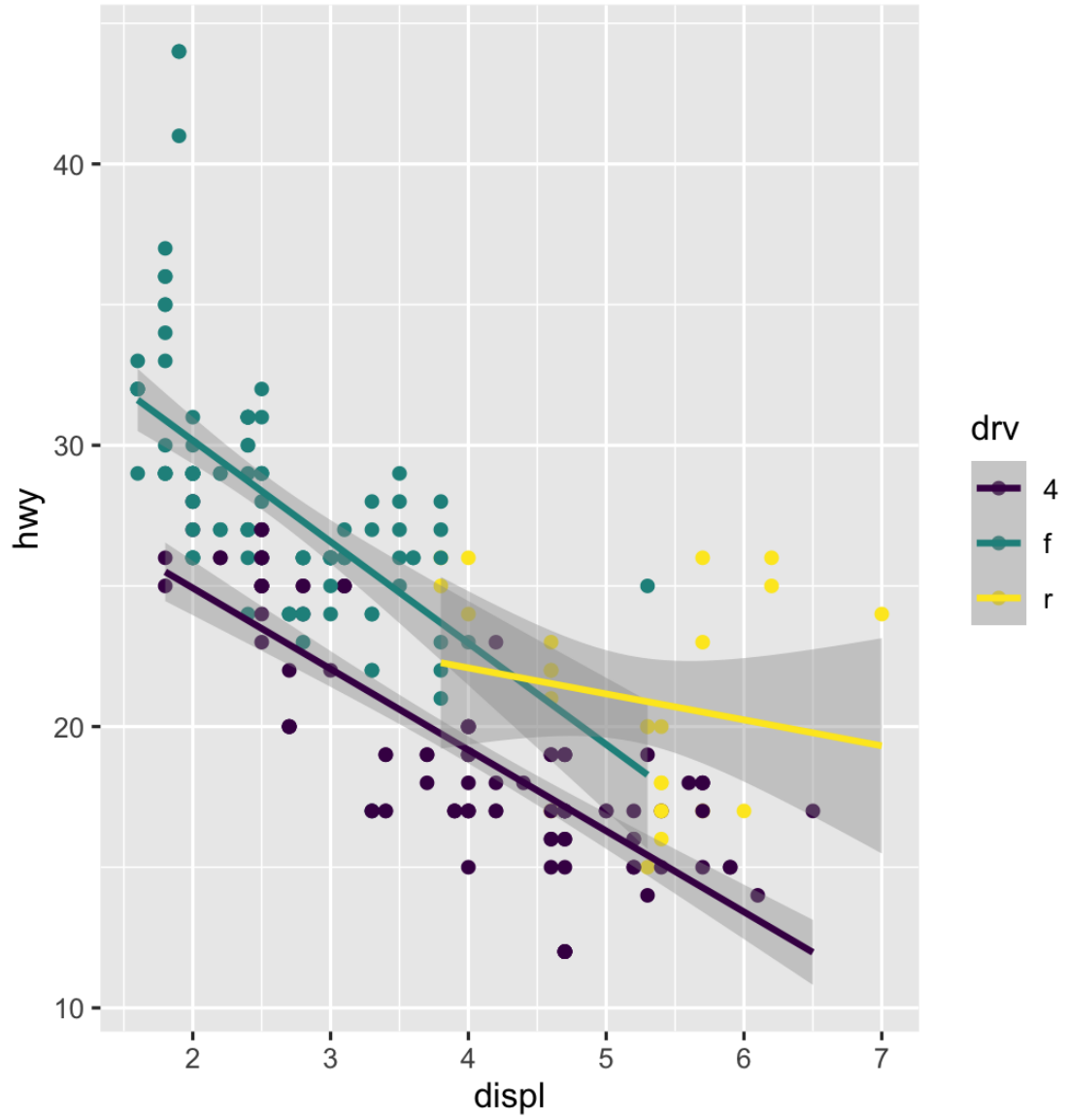
Make it straight

```
ggplot(data = mpg,
       mapping = aes(x = displ,
                     y = hwy,
                     color = drv)) +
  geom_point() +
  geom_smooth(method = "lm")
```



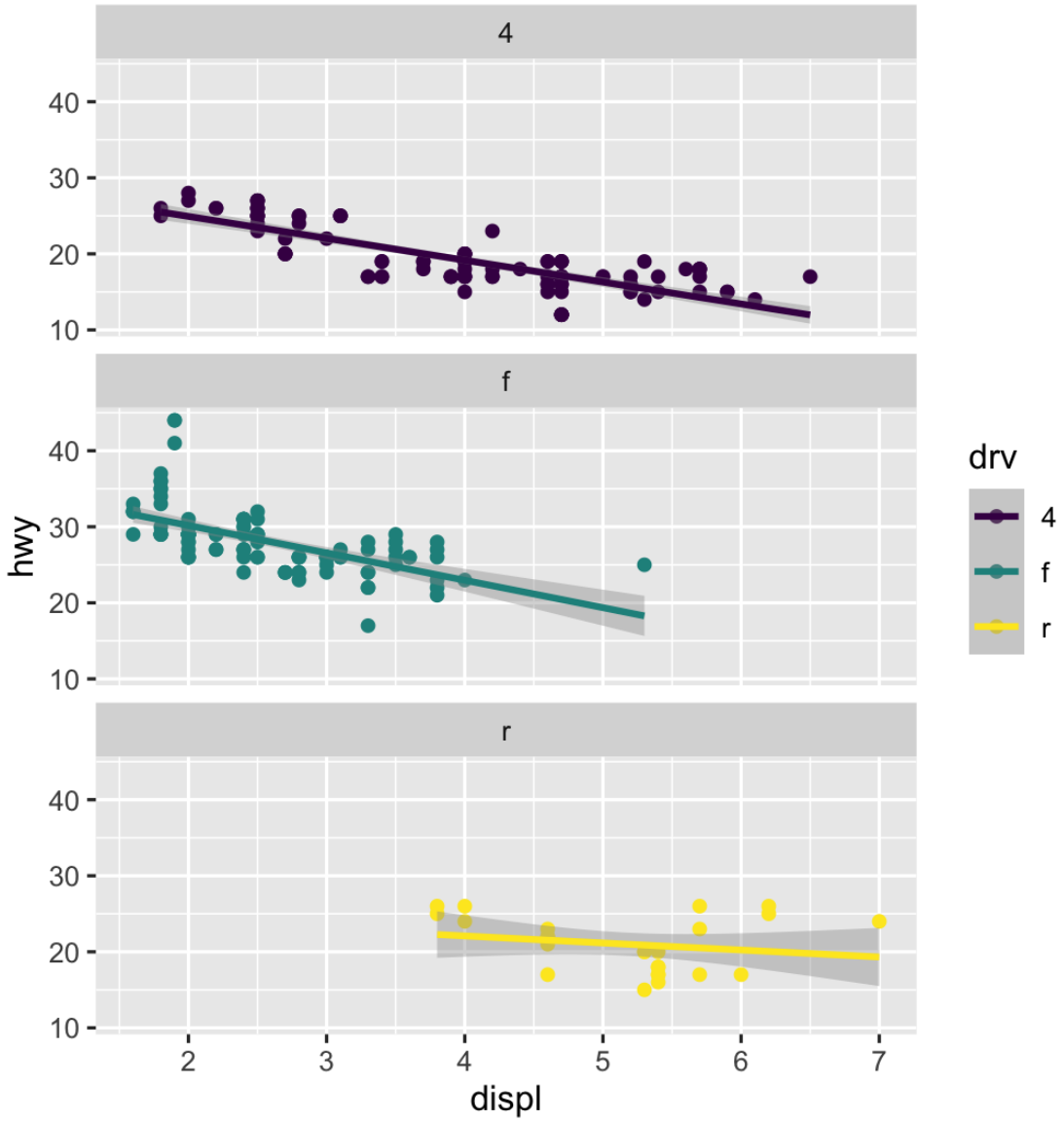
Use a viridis color scale

```
ggplot(data = mpg,
       mapping = aes(x = displ,
                     y = hwy,
                     color = drv)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_color_viridis_d()
```



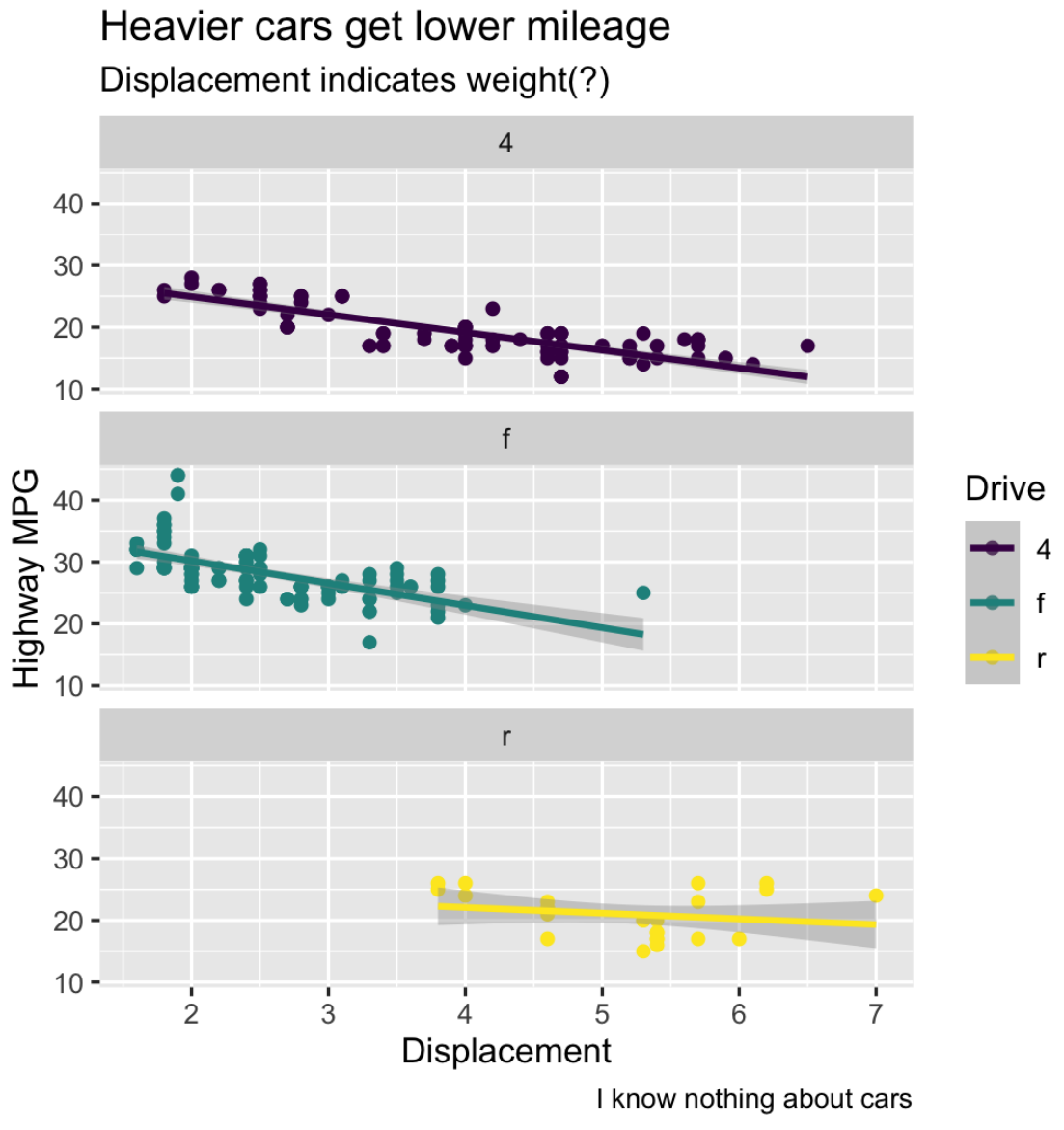
Facet by drive

```
ggplot(data = mpg,
       mapping = aes(x = displ,
                     y = hwy,
                     color = drv)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_color_viridis_d() +
  facet_wrap(vars(drv), ncol = 1)
```



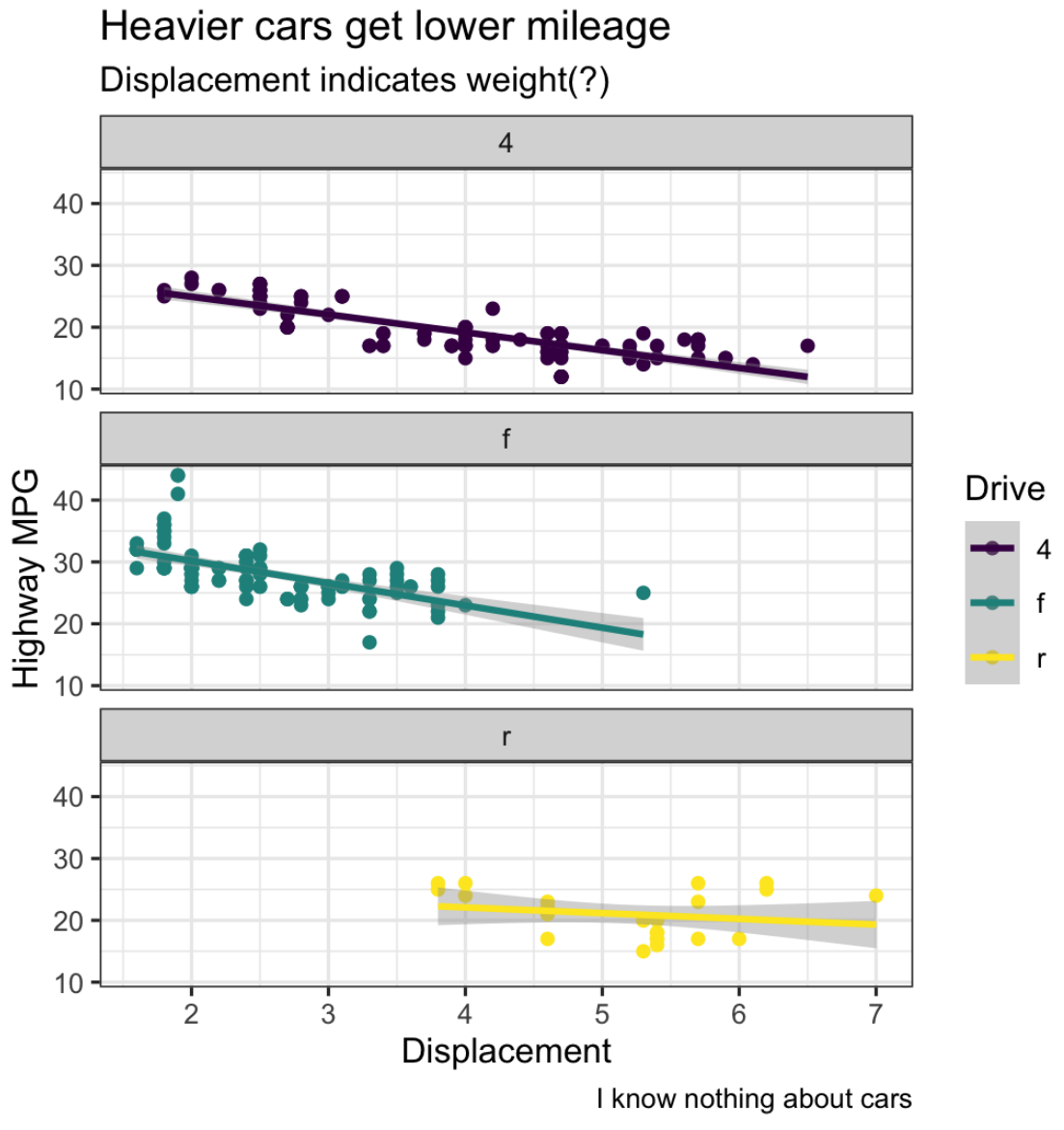
Add labels

```
ggplot(data = mpg,
       mapping = aes(x = displ,
                     y = hwy,
                     color = drv)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_color_viridis_d() +
  facet_wrap(vars(drv), ncol = 1) +
  labs(x = "Displacement", y = "Highway MPG",
       color = "Drive",
       title = "Heavier cars get lower mileag
       subtitle = "Displacement indicates we
       caption = "I know nothing about cars")
```



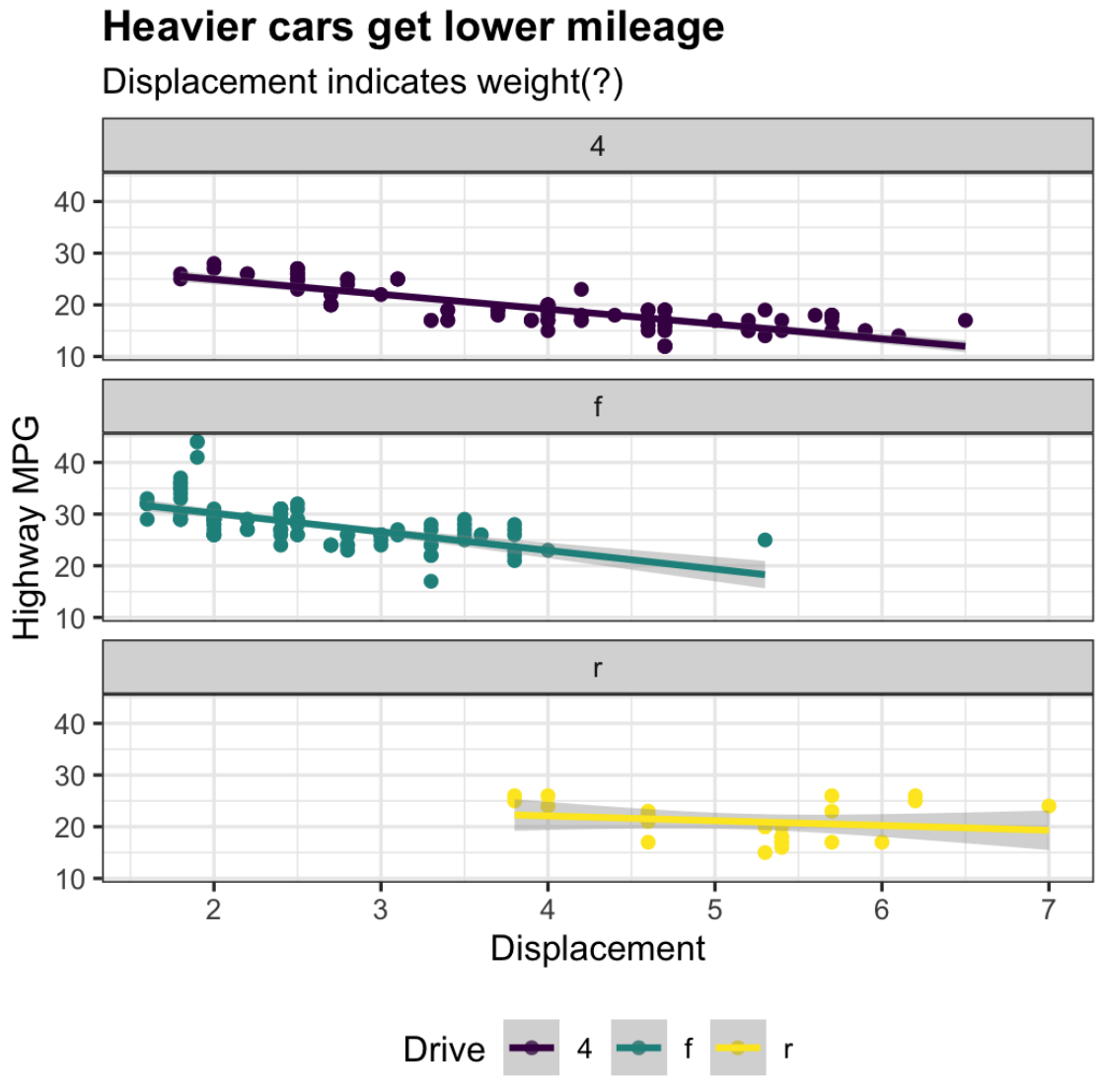
Add a theme

```
ggplot(data = mpg,
       mapping = aes(x = displ,
                     y = hwy,
                     color = drv)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_color_viridis_d() +
  facet_wrap(vars(drv), ncol = 1) +
  labs(x = "Displacement", y = "Highway MPG",
       color = "Drive",
       title = "Heavier cars get lower mileag
       subtitle = "Displacement indicates wei
       caption = "I know nothing about cars")
  theme_bw()
```



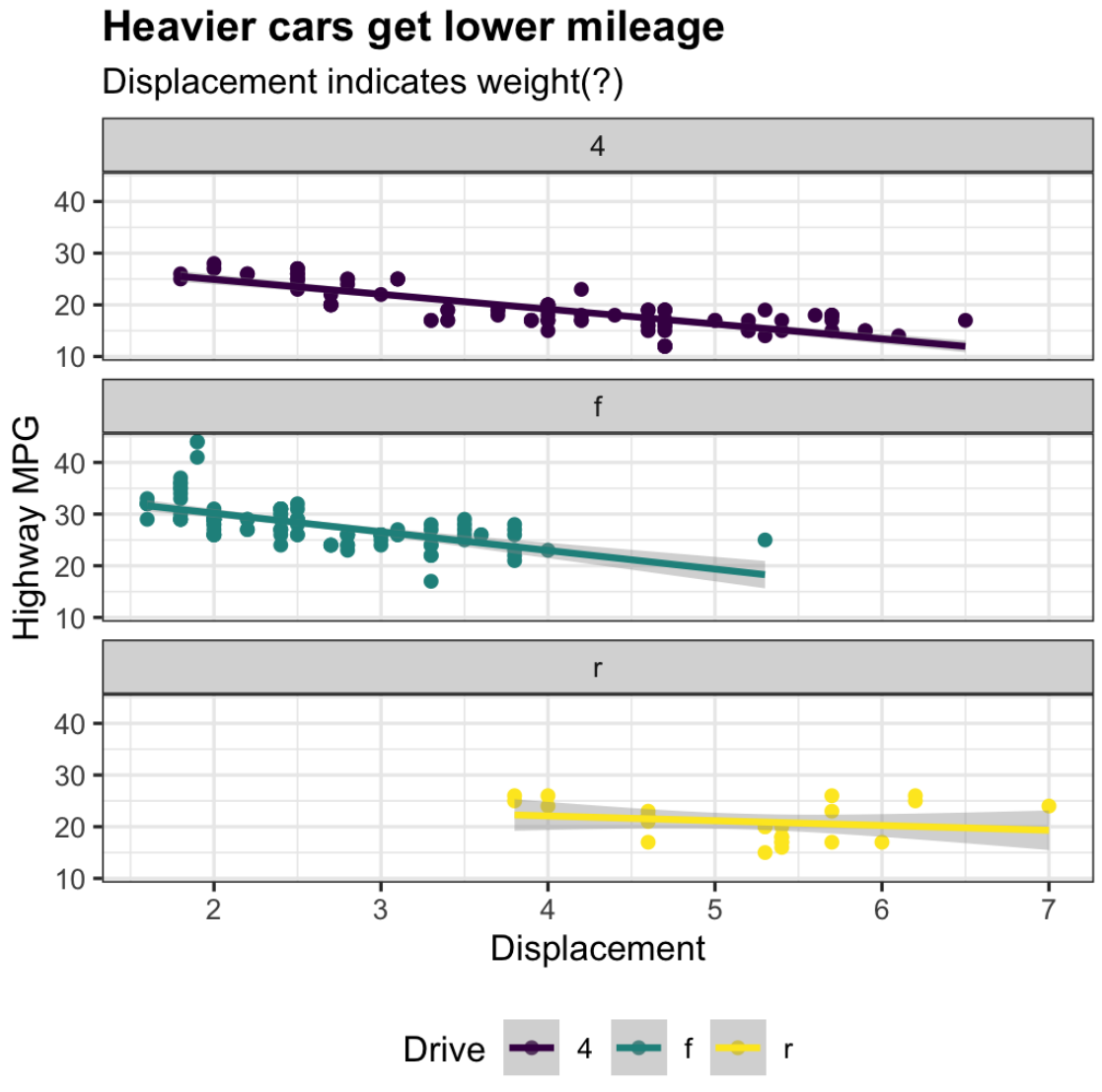
Modify the theme

```
ggplot(data = mpg,
       mapping = aes(x = displ,
                     y = hwy,
                     color = drv)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_color_viridis_d() +
  facet_wrap(vars(drv), ncol = 1) +
  labs(x = "Displacement", y = "Highway MPG"
       color = "Drive",
       title = "Heavier cars get lower mileag
       subtitle = "Displacement indicates we
       caption = "I know nothing about cars")
  theme_bw() +
  theme(legend.position = "bottom",
        plot.title = element_text(face = "bo
```

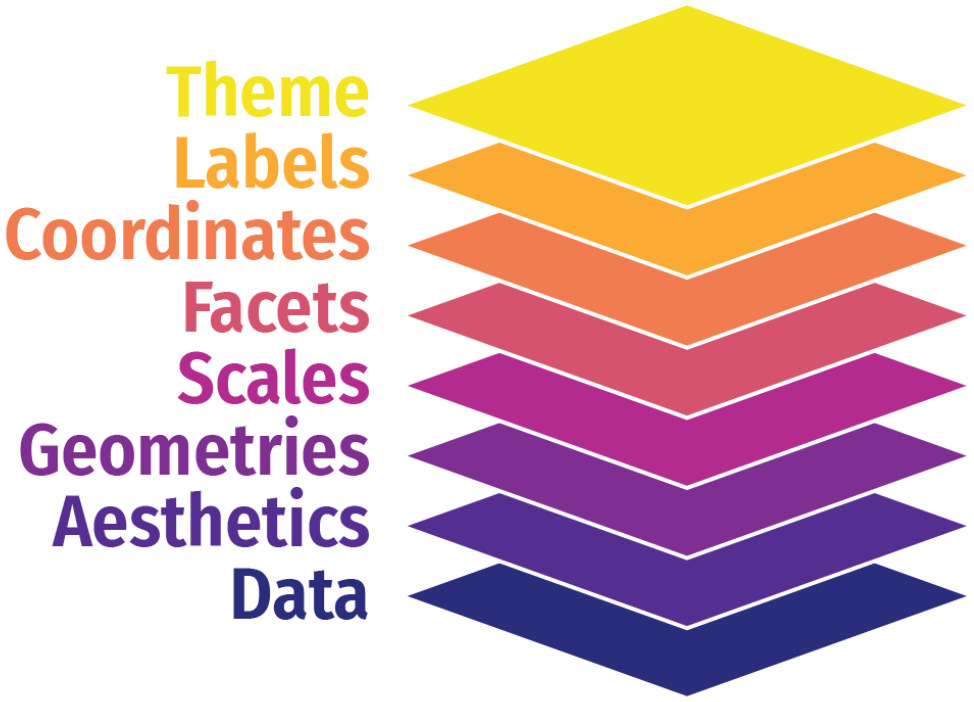


Finished!

```
ggplot(data = mpg,
       mapping = aes(x = displ,
                     y = hwy,
                     color = drv)) +
  geom_point() +
  geom_smooth(method = "lm") +
  scale_color_viridis_d() +
  facet_wrap(vars(drv), ncol = 1) +
  labs(x = "Displacement", y = "Highway MPG"
       color = "Drive",
       title = "Heavier cars get lower mileag
       subtitle = "Displacement indicates we
       caption = "I know nothing about cars"
  theme_bw() +
  theme(legend.position = "bottom",
        plot.title = element_text(face = "bo
```



So many possibilities!



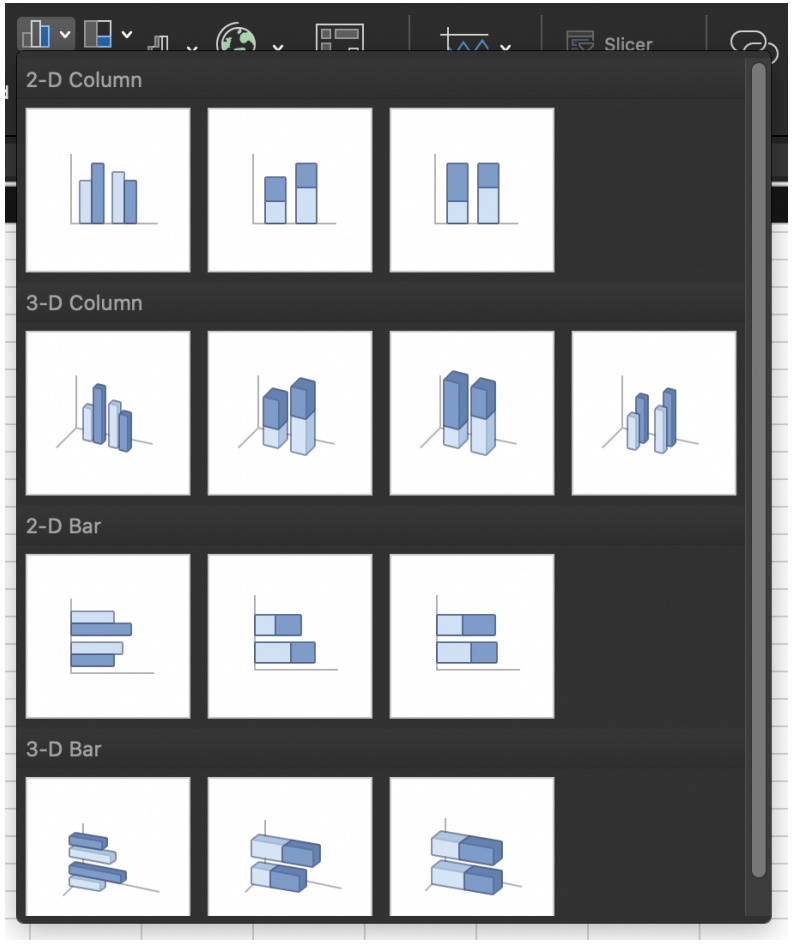
These were just a few examples of layers!

See the ggplot2 documentation for complete examples of everything you can do

A true grammar

With the grammar of graphics, we don't talk about specific chart *types*

Hunt through Excel menus for a stacked bar chart and manually reshape your data to work with it

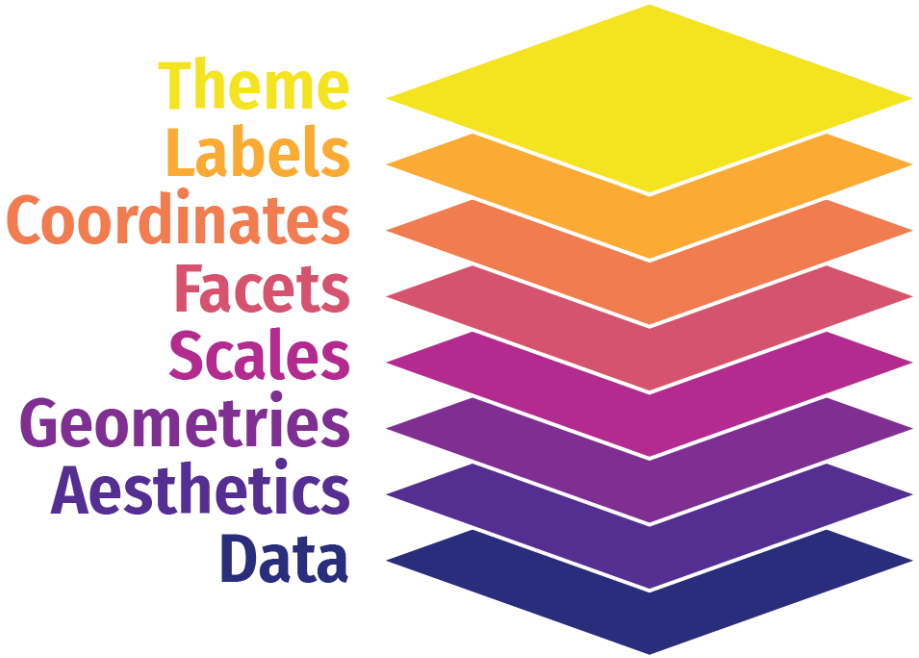


A true grammar

With the grammar of graphics, we *do* talk about specific chart *elements*

Map a column to the x-axis, fill by a different variable, and `geom_col()` to get stacked bars

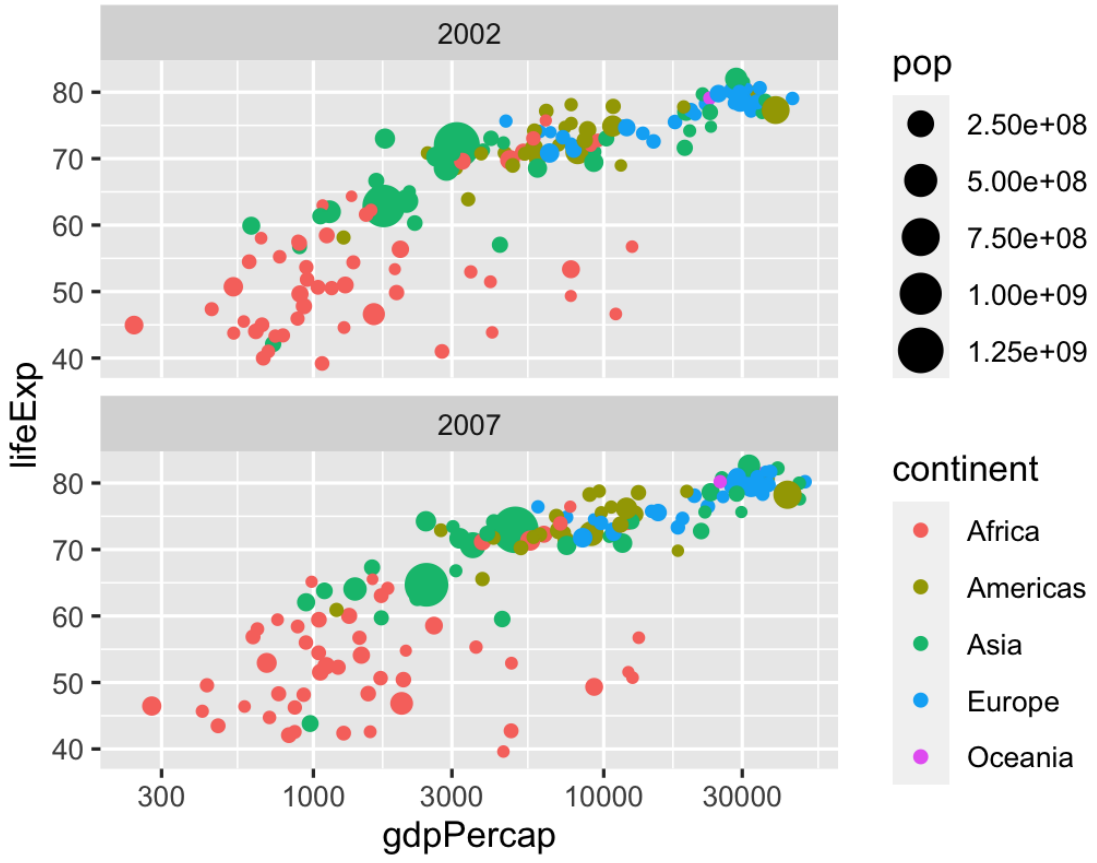
Geoms can be interchangable (e.g. switch `geom_violin()` to `geom_boxplot()`)



Describing graphs with the grammar

Map wealth to the x-axis, health to the y-axis, add points, color by continent, size by population, scale the y-axis with a log, and facet by year

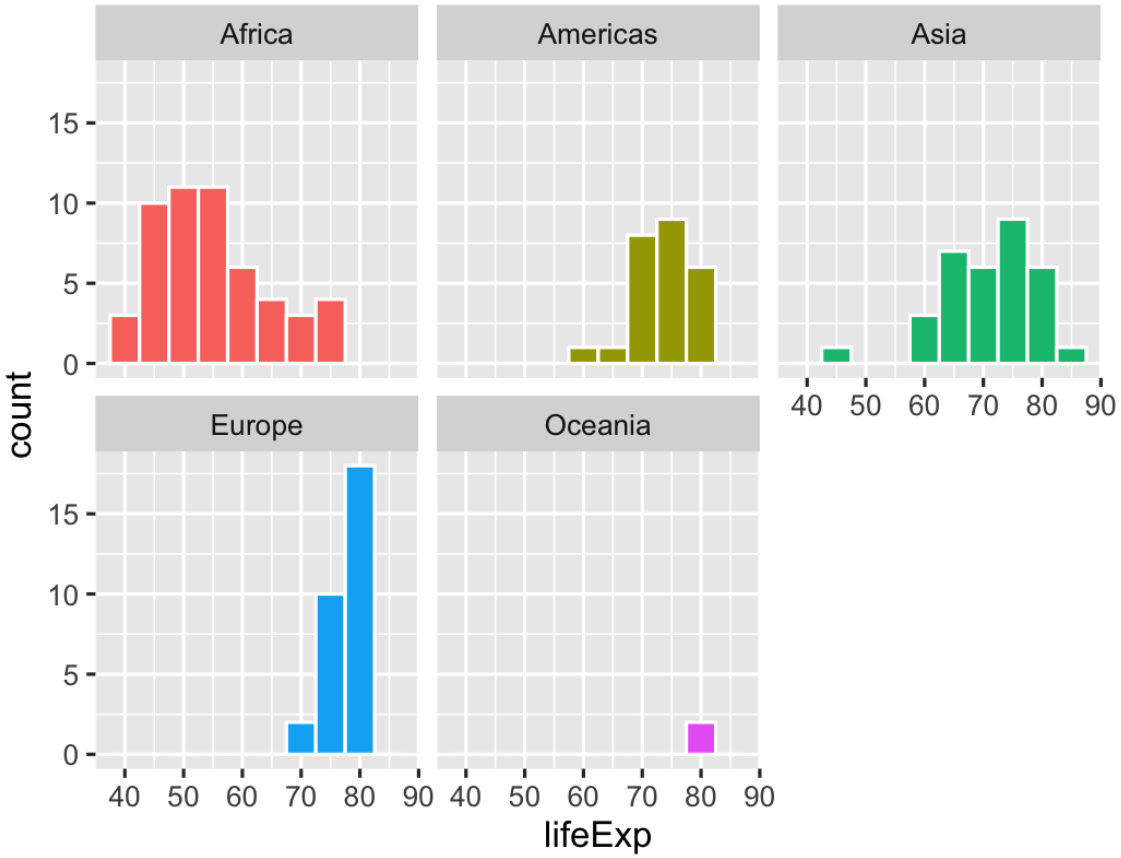
```
ggplot(filter(gapminder,
              year %in% c(2002, 2007)),
       aes(x = gdpPercap,
           y = lifeExp,
           color = continent,
           size = pop)) +
  geom_point() +
  scale_x_log10() +
  facet_wrap(vars(year), ncol = 1)
```



Describing graphs with the grammar

Map health to the x-axis, add a histogram with bins for every 5 years, fill and facet by continent

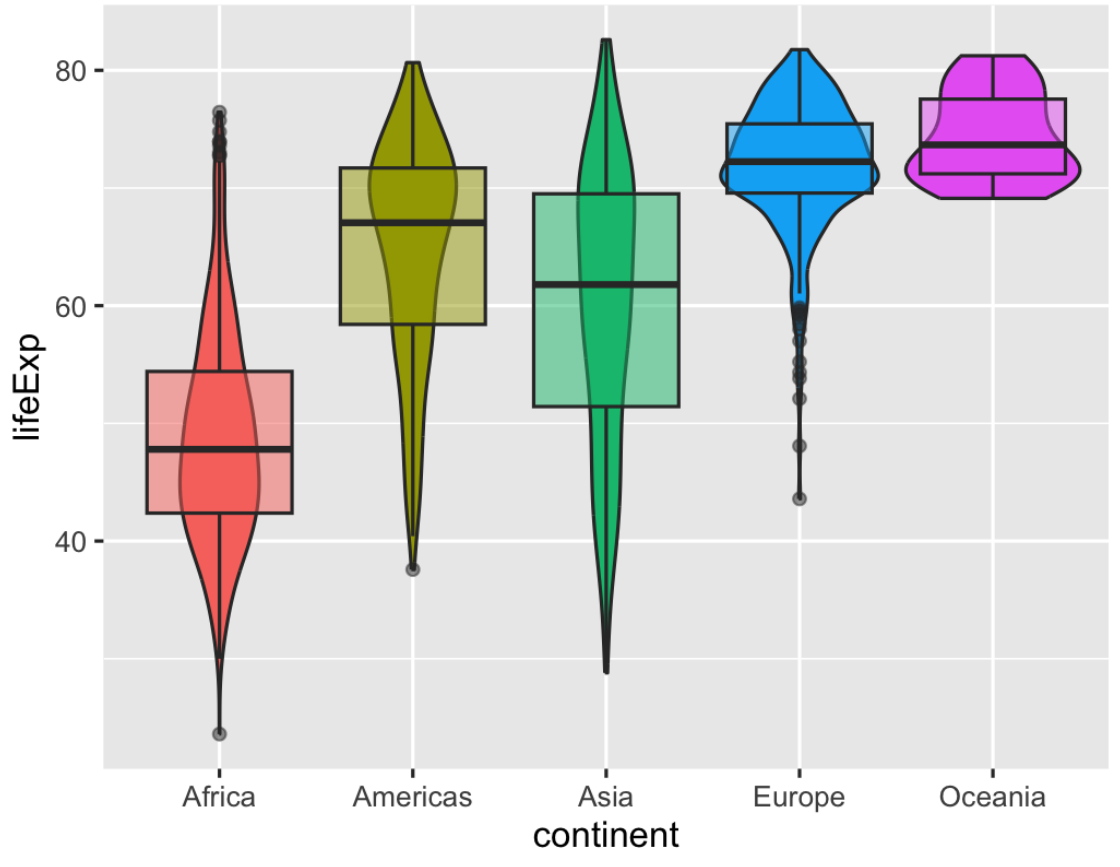
```
ggplot(gapminder_2007,
       aes(x = lifeExp,
           fill = continent)) +
  geom_histogram(binwidth = 5,
                 color = "white") +
  guides(fill = FALSE) +  # Turn off legend
  facet_wrap(vars(continent))
```



Describing graphs with the grammar

Map continent to the x-axis, health to the y-axis, add violin plots and semi-transparent boxplots, fill by continent

```
ggplot(gapminder,
       aes(x = continent,
           y = lifeExp,
           fill = continent)) +
  geom_violin() +
  geom_boxplot(alpha = 0.5) +
  guides(fill = FALSE)  # Turn off legend
```



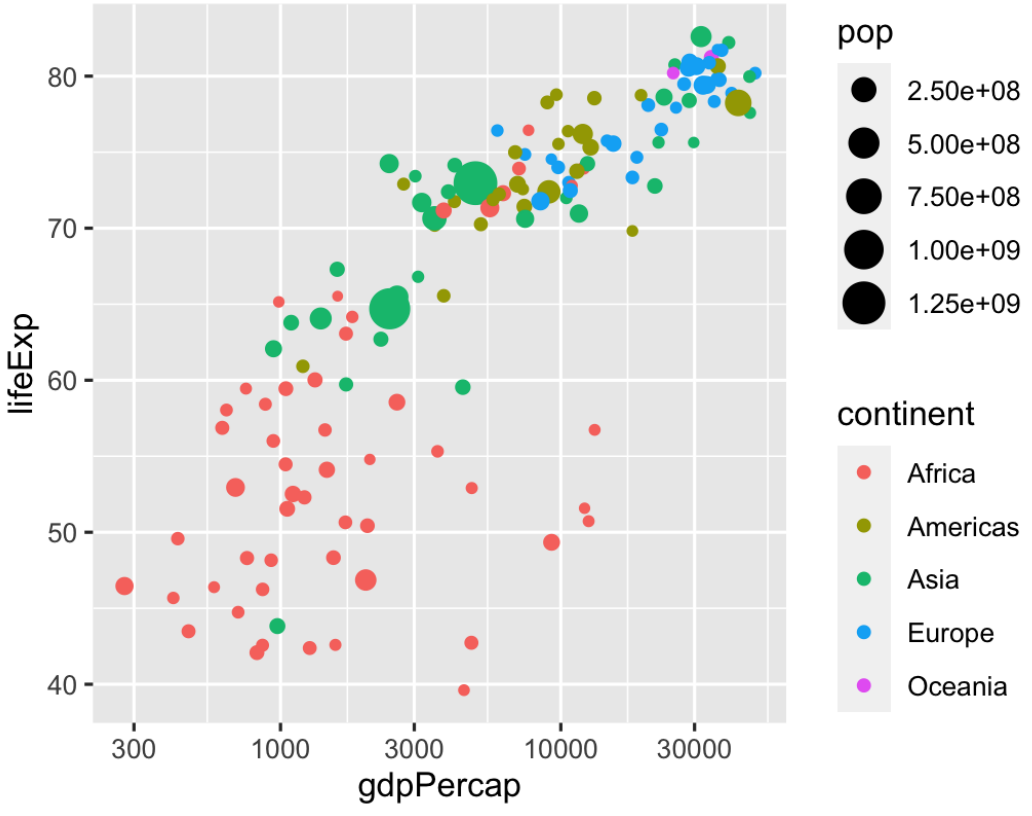
Scales

Scales change the properties of the variable mapping

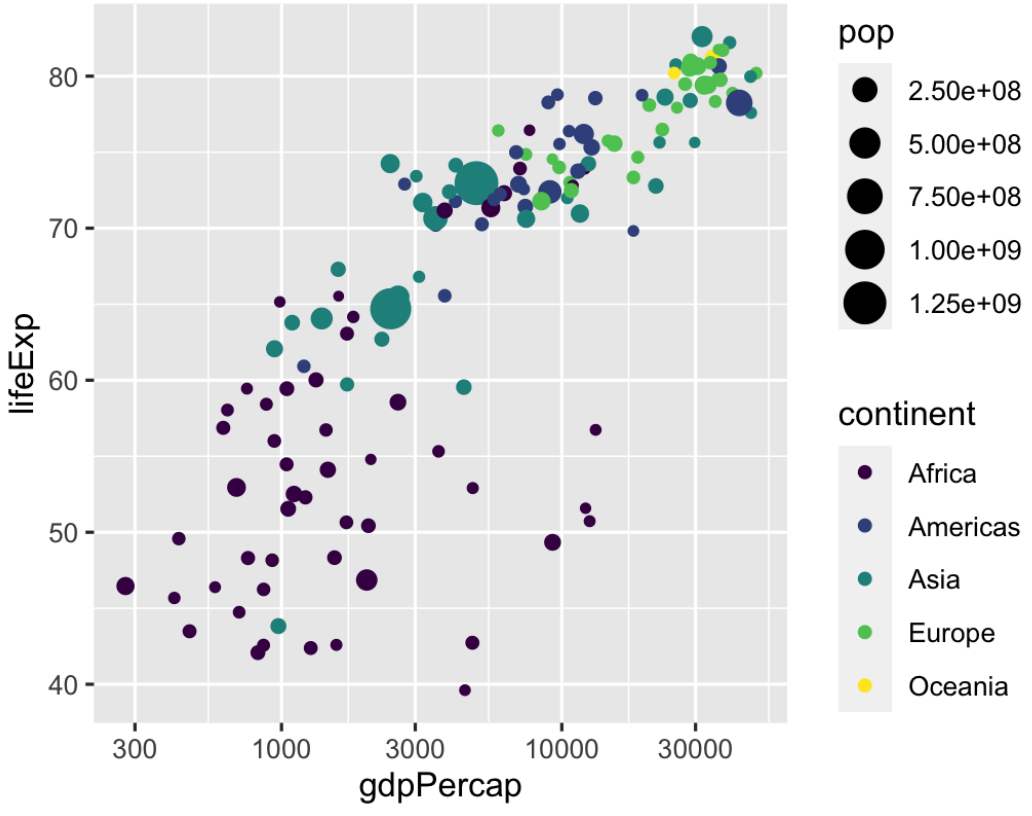
Example layer	What it does
`scale_x_continuous()`	Make the x-axis continuous
`scale_x_continuous(breaks = 1:5)`	Manually specify axis ticks
`scale_x_log10()`	Log the x-axis
`scale_color_gradient()`	Use a gradient
`scale_fill_viridis_d()`	Fill with discrete viridis colors

Scales

`scale_x_log10()`



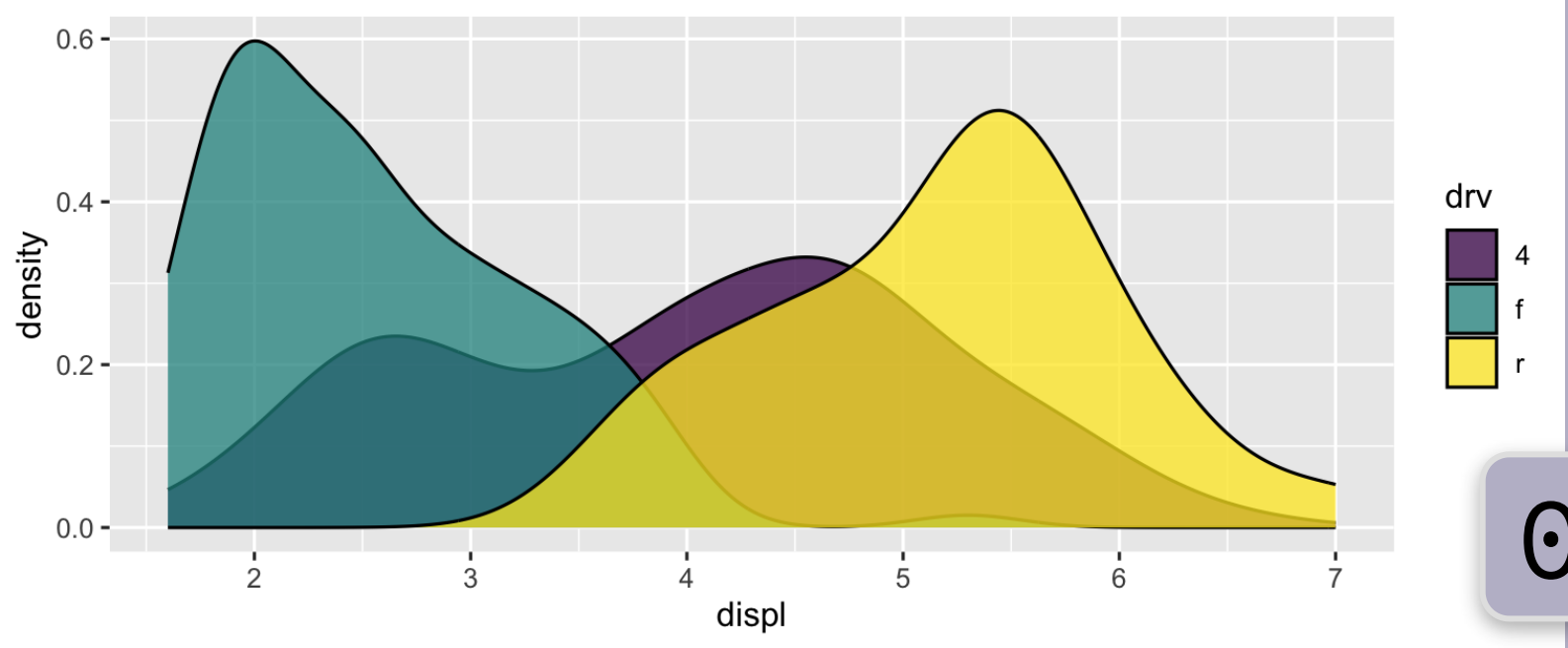
`scale_color_viridis_d()`



Your turn #7

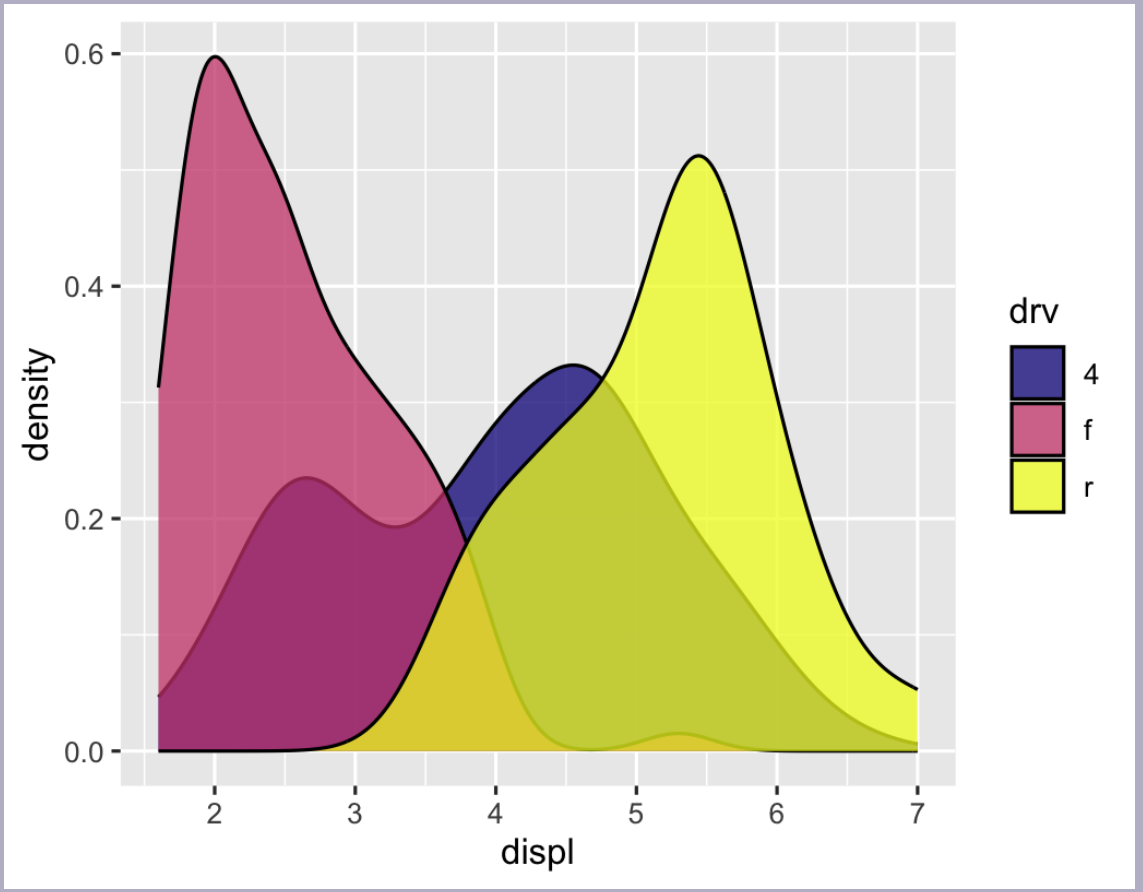
Make this density plot of `displ` filled by `drv`. Use the viridis fill scale.

For bonus fun, try a different viridis option like `plasma` or `inferno`.



03:00

```
ggplot(mpg,
       aes(x = displ,
           fill = drv)) +
  geom_density(alpha = 0.75) +
  scale_fill_viridis_d(option = "plasma")
```



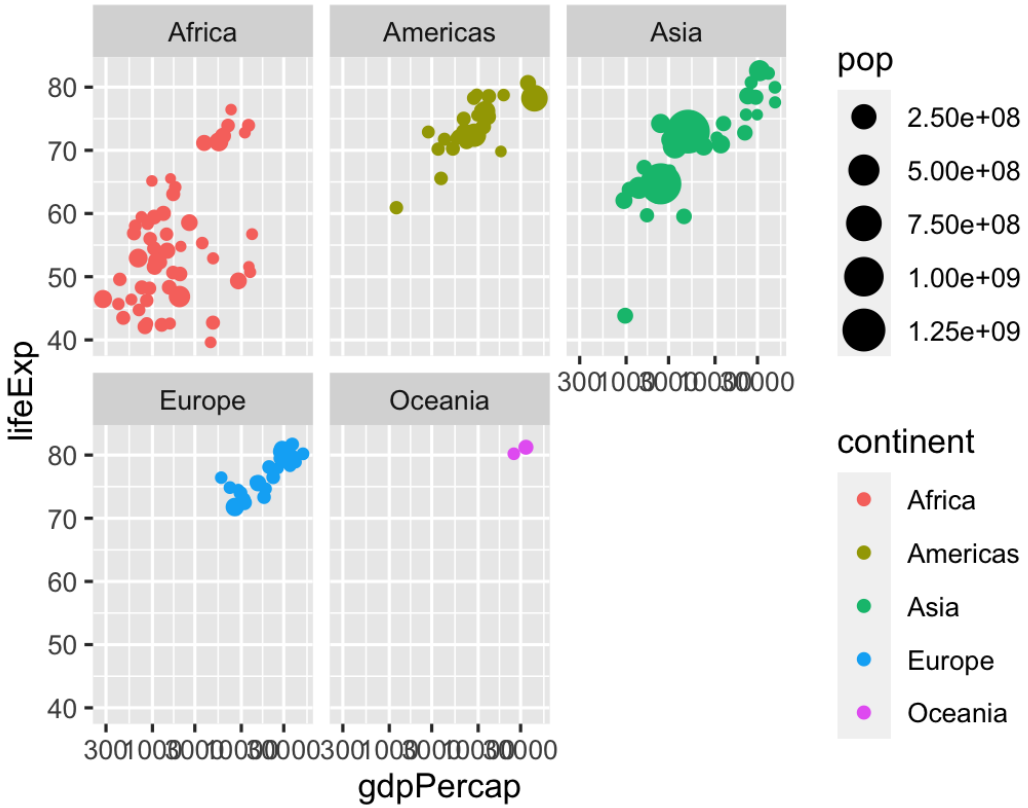
Facets

Facets show subplots for different subsets of data

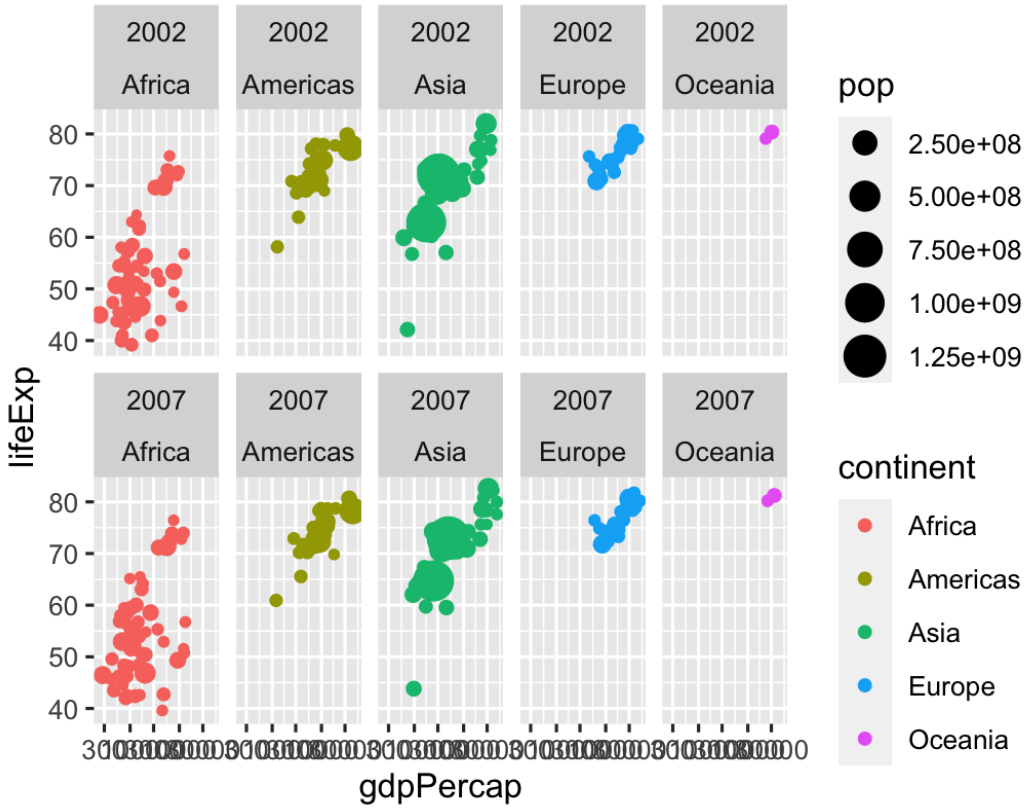
Example layer	What it does
`facet_wrap(vars(continent))`	Plot for each continent
`facet_wrap(vars(continent, year))`	Plot for each continent/year
`facet_wrap(..., ncol = 1)`	Put all facets in one column
`facet_wrap(..., nrow = 1)`	Put all facets in one row

Facets

`facet_wrap(vars(continent))`

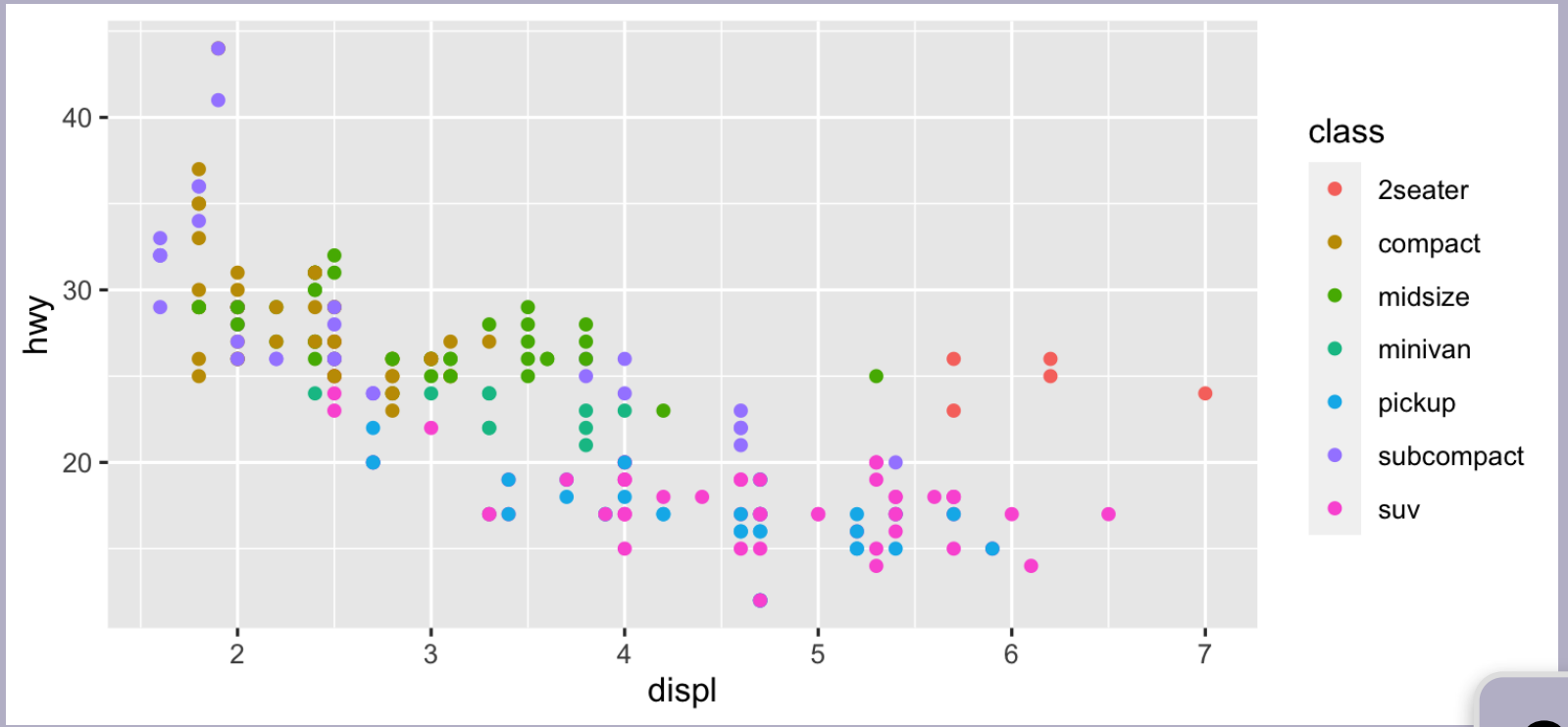


`facet_wrap(vars(continent, year))`



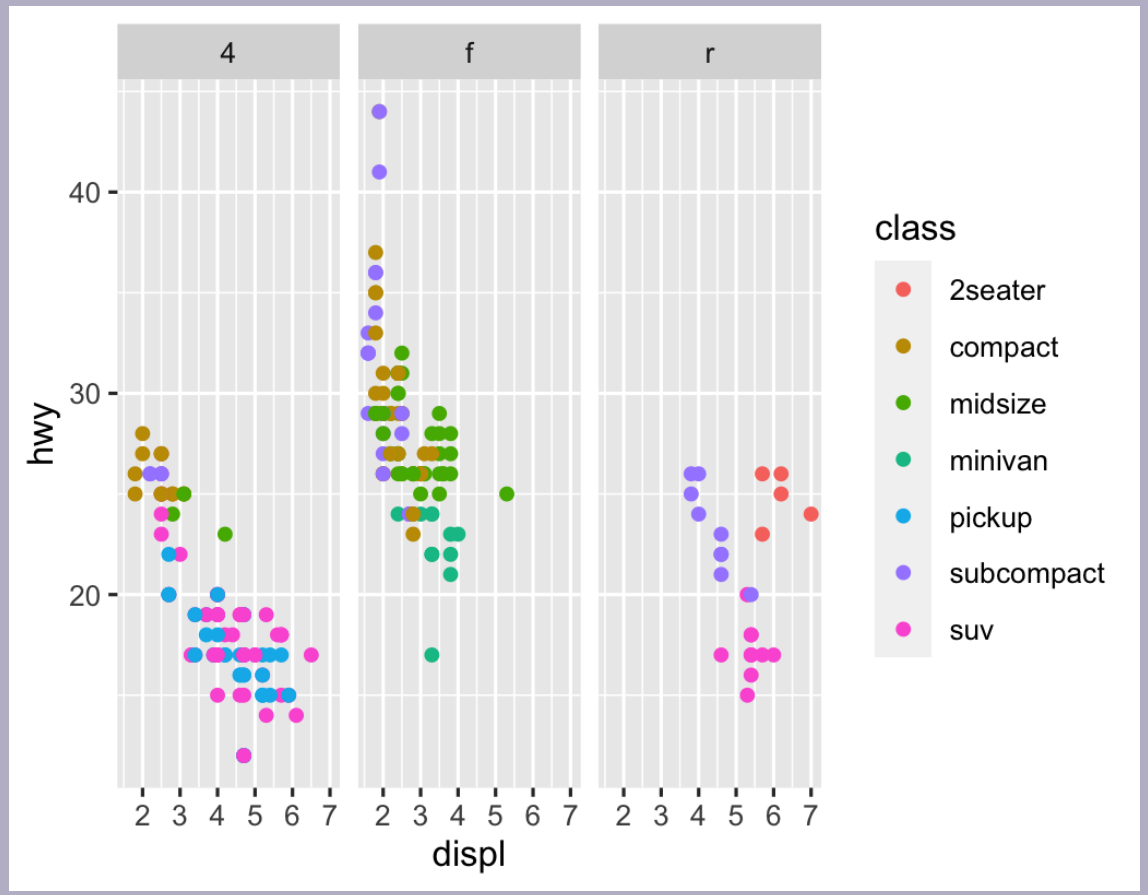
Your turn #8

Facet this scatterplot by `drv`. Are there any interesting trends?



03:00

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



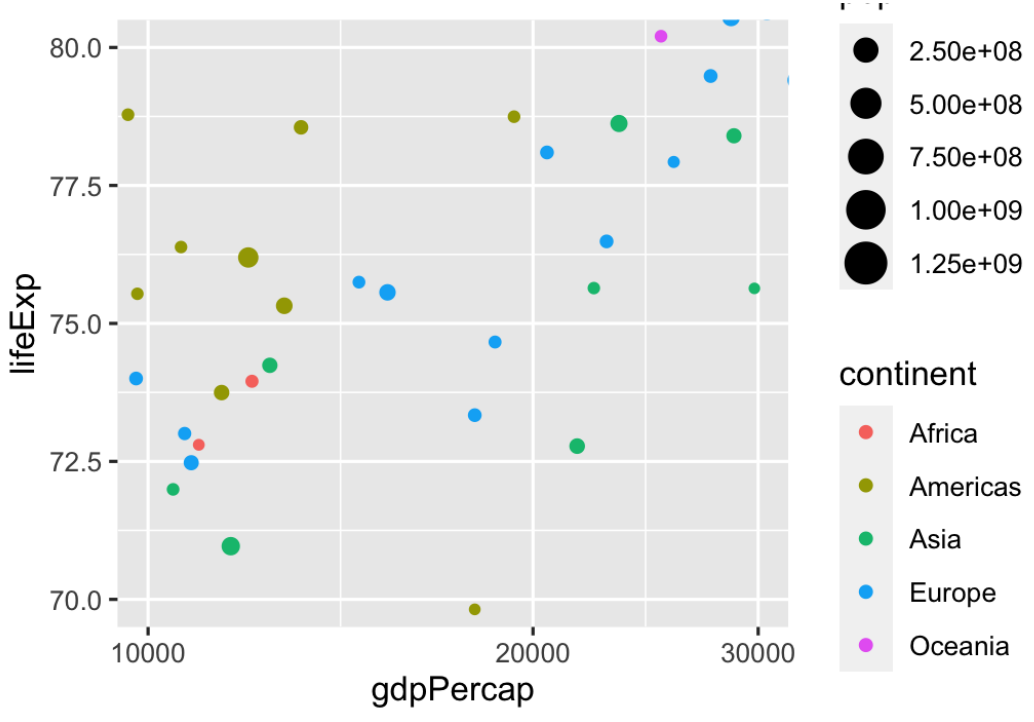
Coordinates

Change the coordinate system

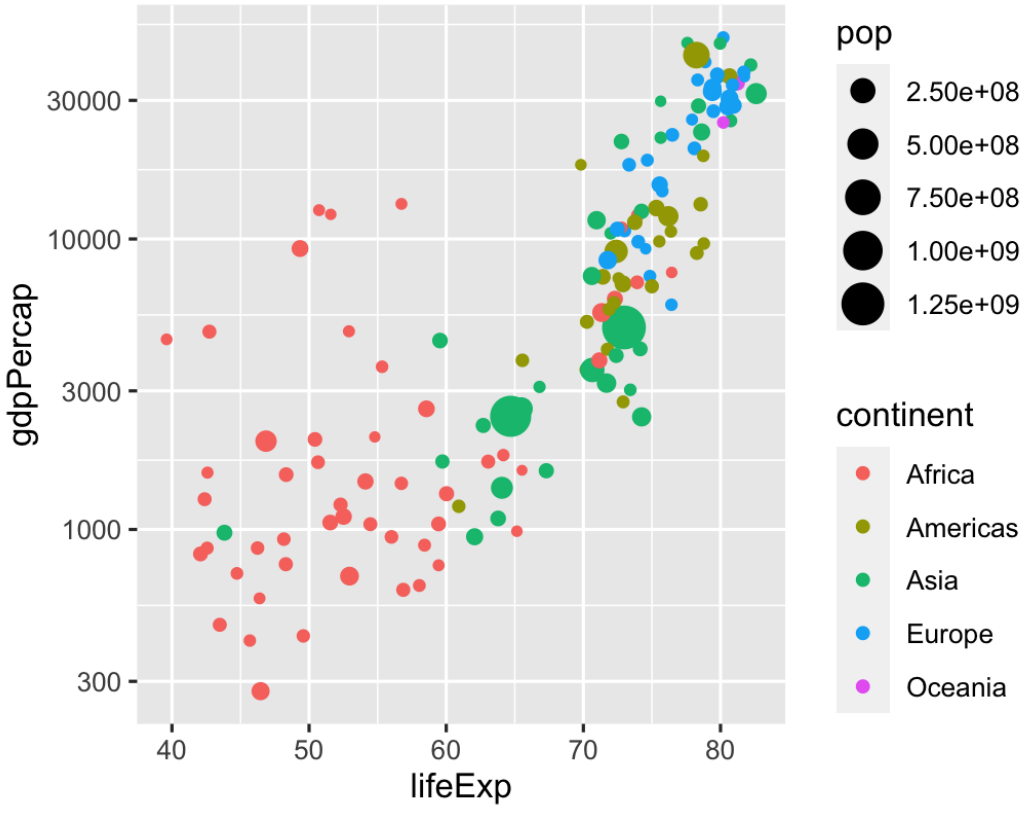
Example layer	What it does
`coord_cartesian()`	Standard x-y coordinate system
`coord_cartesian(ylim = c(1, 10))`	Zoom in where y is 1–10
`coord_flip()`	Switch x and y
`coord_polar()`	Use circular polar system

Coordinates

```
coord_cartesian(ylim = c(70, 80),
xlim = c(10000, 30000))
```




```
coord_flip()
```



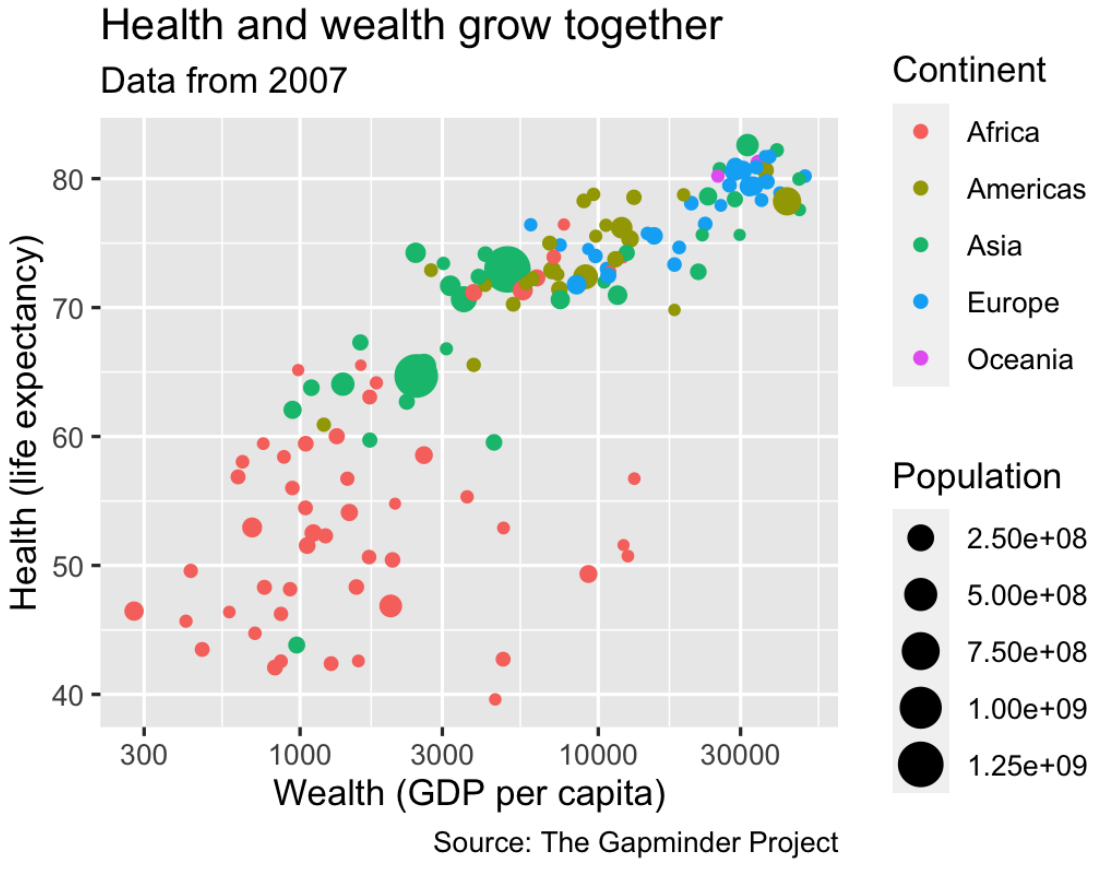
Labels

Add labels to the plot with a single `labs()` layer

Example layer	What it does
`labs(title = "Neat title")`	Title
`labs(caption = "Something")`	Caption
`labs(y = "Something")`	y-axis
`labs(size = "Population")`	Title of size legend

Labels

```
ggplot(gapminder_2007,
       aes(x = gdpPercap, y = lifeExp,
           color = continent, size = pop)) +
  geom_point() +
  scale_x_log10() +
  labs(title = "Health and wealth grow togeth
       subtitle = "Data from 2007",
       x = "Wealth (GDP per capita)",
       y = "Health (life expectancy)",
       color = "Continent",
       size = "Population",
       caption = "Source: The Gapminder Proje
```



Theme

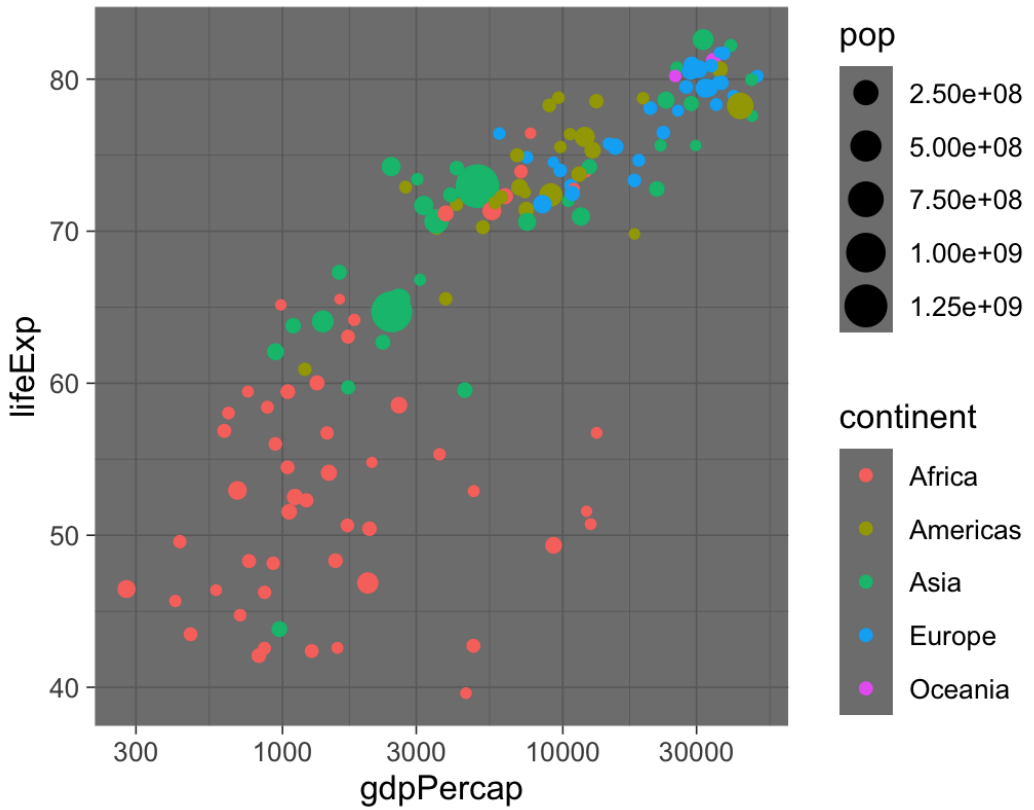
Change the appearance of anything in the plot

There are many built-in themes

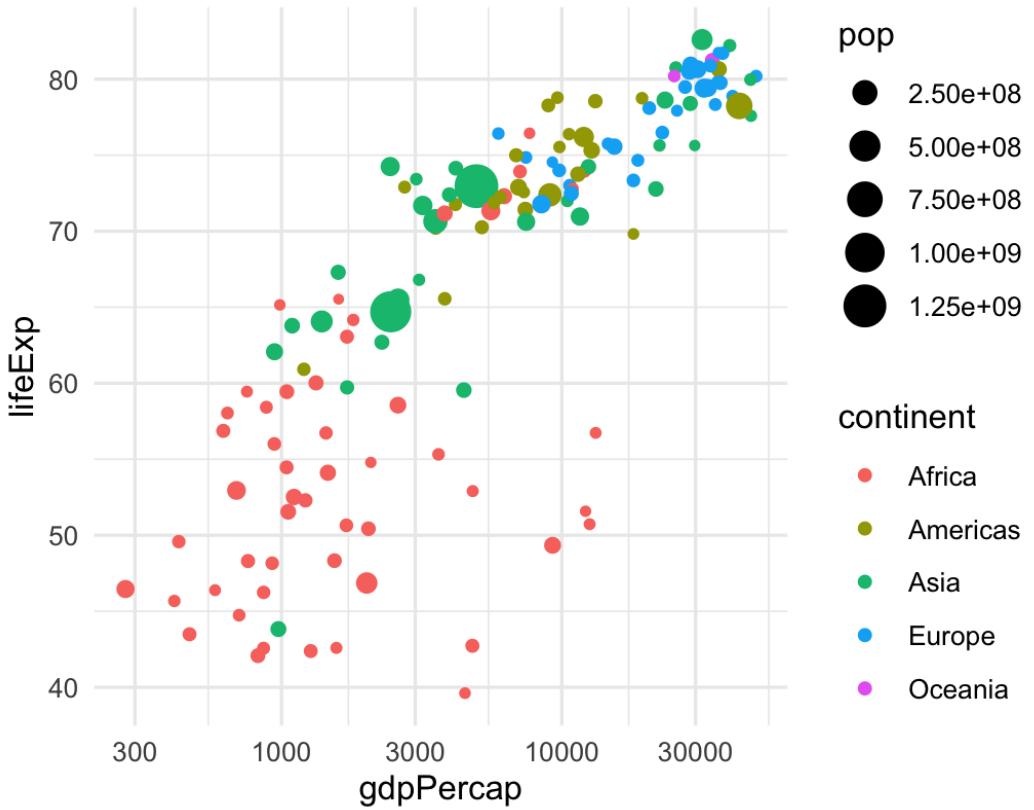
Example layer	What it does
`theme_grey()`	Default grey background
`theme_bw()`	Black and white
`theme_dark()`	Dark
`theme_minimal()`	Minimal

Theme

`theme_dark()`



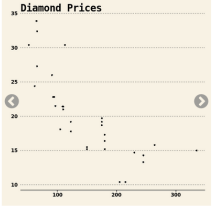
`theme_minimal()`



Theme

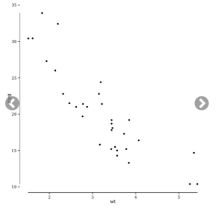
There are collections of pre-built themes online, like the ggthemes package

ggthemes



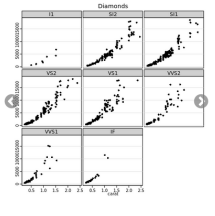
theme_wsj

Wall Street Journal theme



theme_tufte

Tufte Maximal Data, Minimal Ink Theme

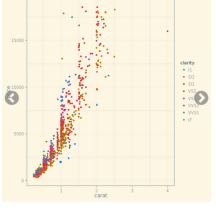


theme_stata

Themes based on Stata graph schemes

theme_solid

Theme with nothing other than a background color

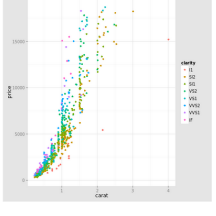


theme_solarized

ggplot color themes based on the Solarized palette

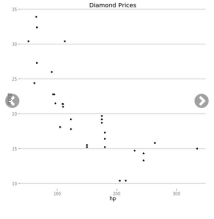
theme_map

Clean theme for maps



theme_igray

Inverse gray theme

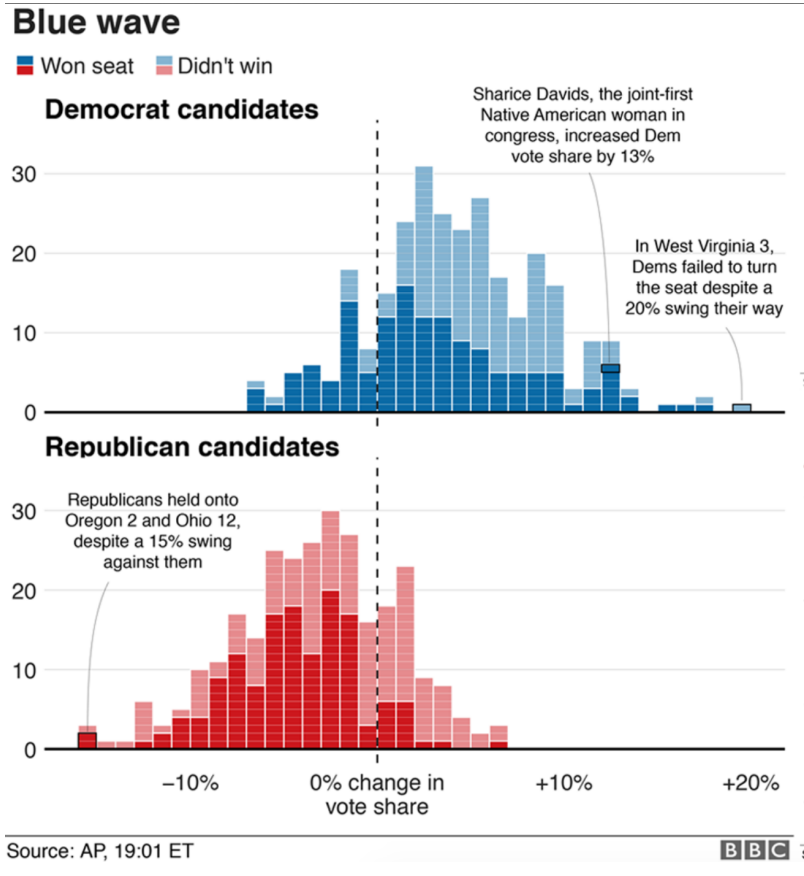


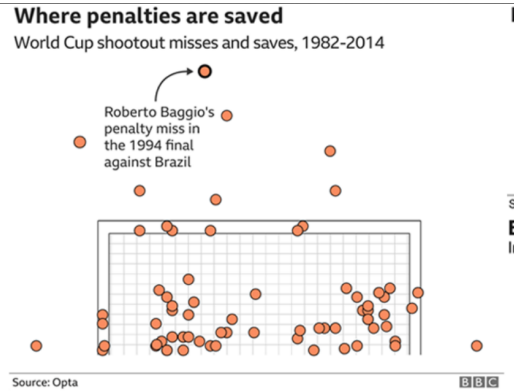
theme_hc

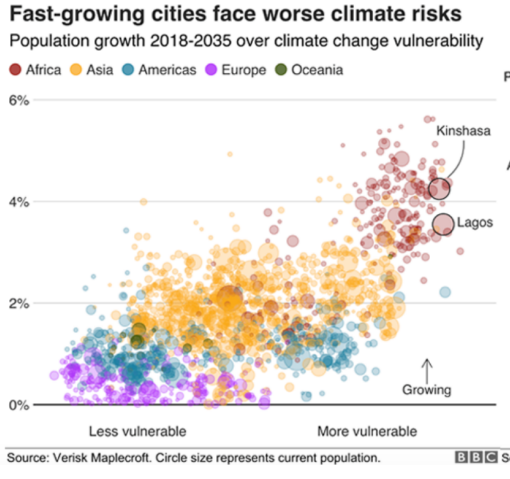
Highcharts JS theme

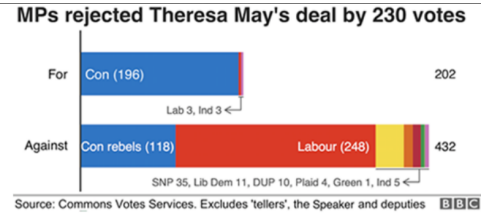
Theme

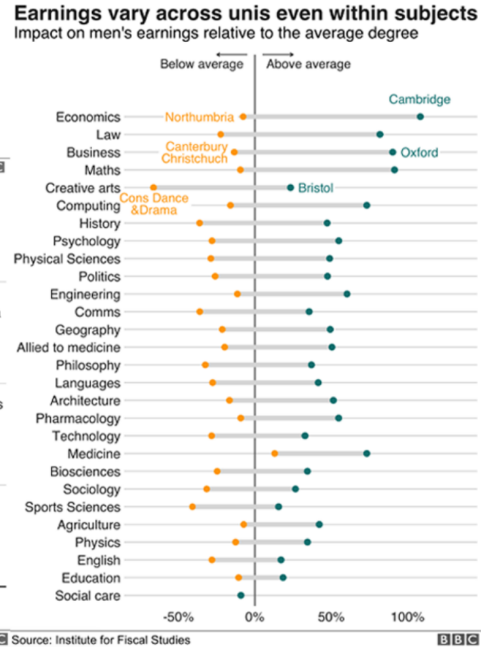
Organizations often make their own custom themes, like the BBC











Theme options

Make theme adjustments with `theme()`

There are a billion options here!

```
theme_bw() +
theme(legend.position = "bottom",
      plot.title = element_text(face = "bold"),
      panel.grid = element_blank(),
      axis.title.y = element_text(face = "italic"))
```

Saving graphs

Use `ggsave()` to save a plot to your computer

Store plot as an object, feed it to `ggsave()`

```
my_plot <- ggplot(...)

ggsave("plot_name.pdf", my_plot, width = 5, height = 3.5)
ggsave("plot_name.png", my_plot, width = 5, height = 3.5)
```

Next up

Transforming and manipulating data with dplyr