

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

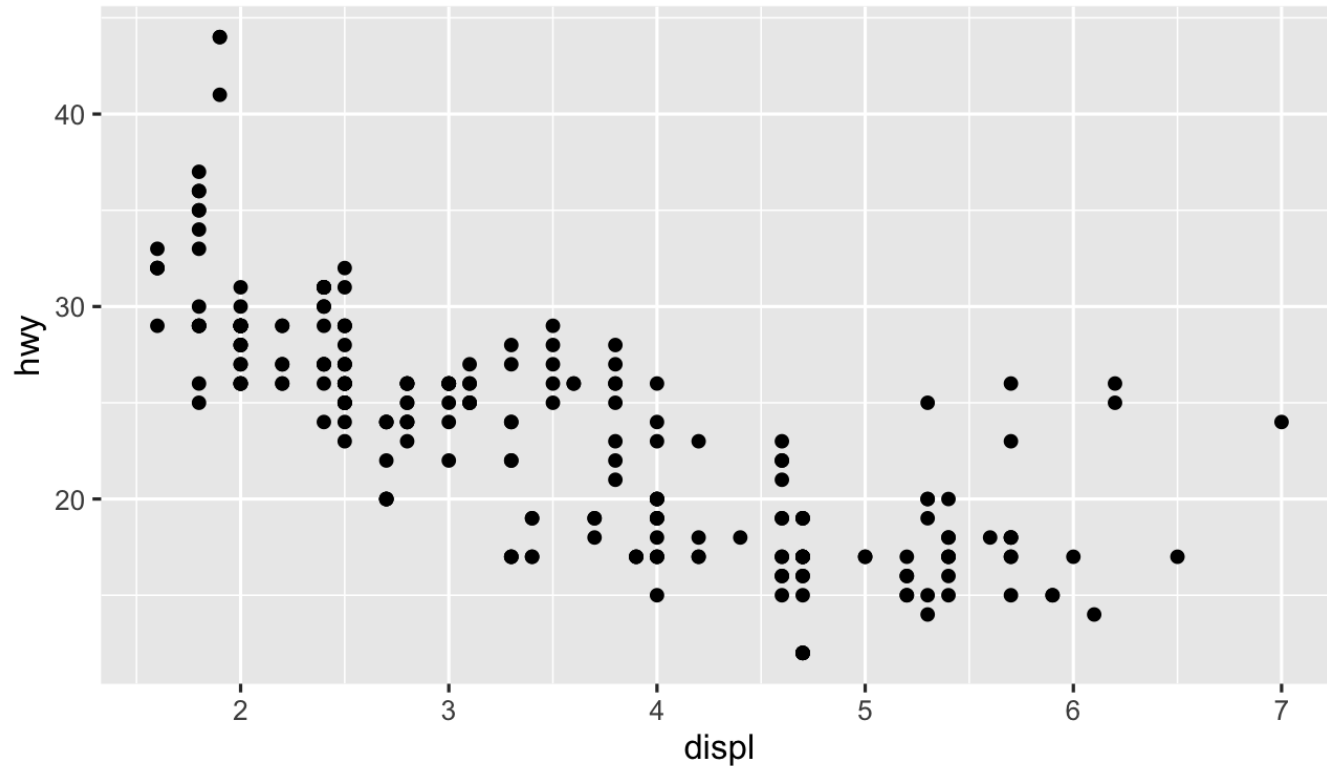
Run this code in your Rmd file to make a graph

Pay attention to spelling, capitalization, and parentheses!

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

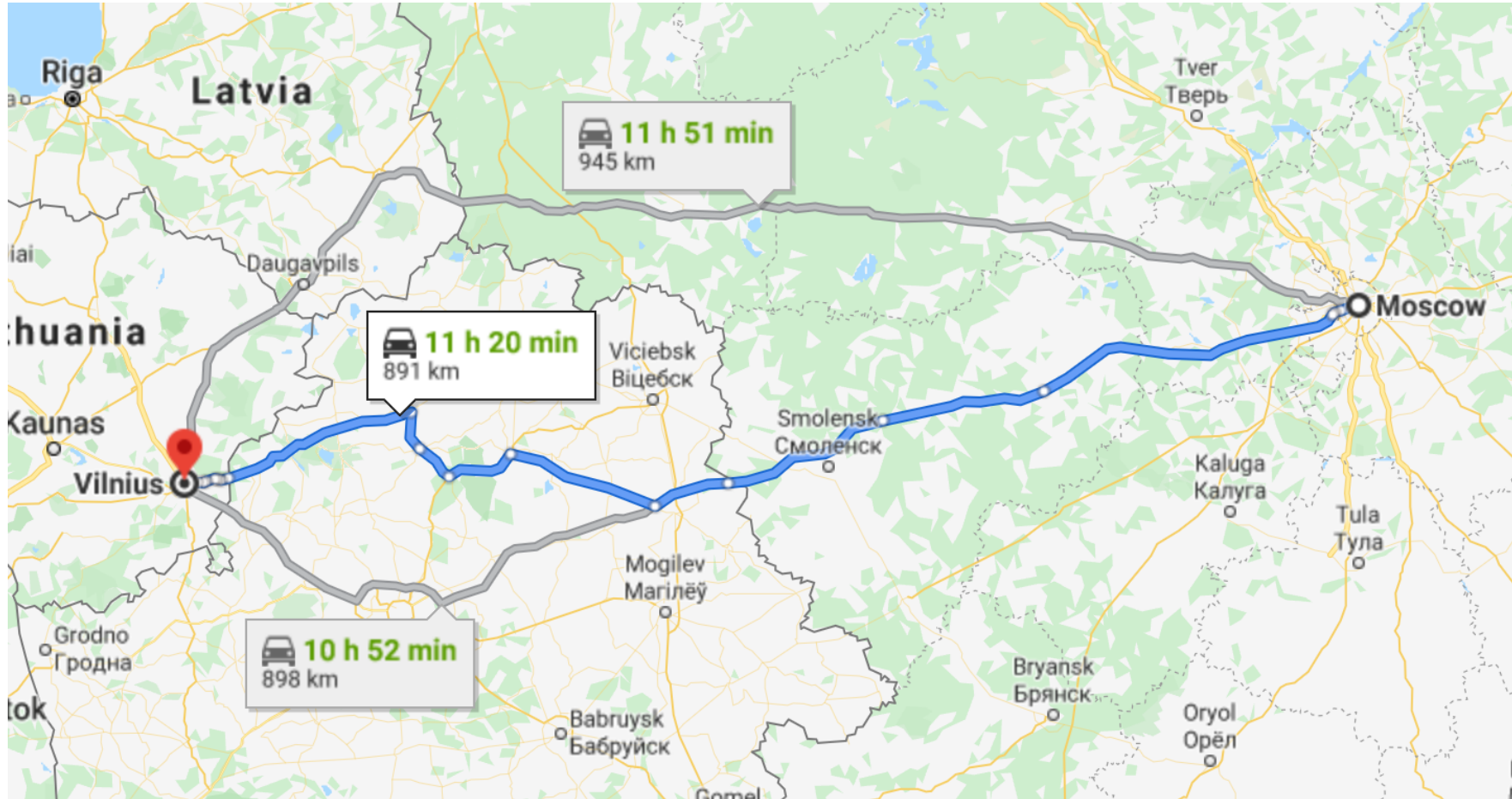
02:00

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



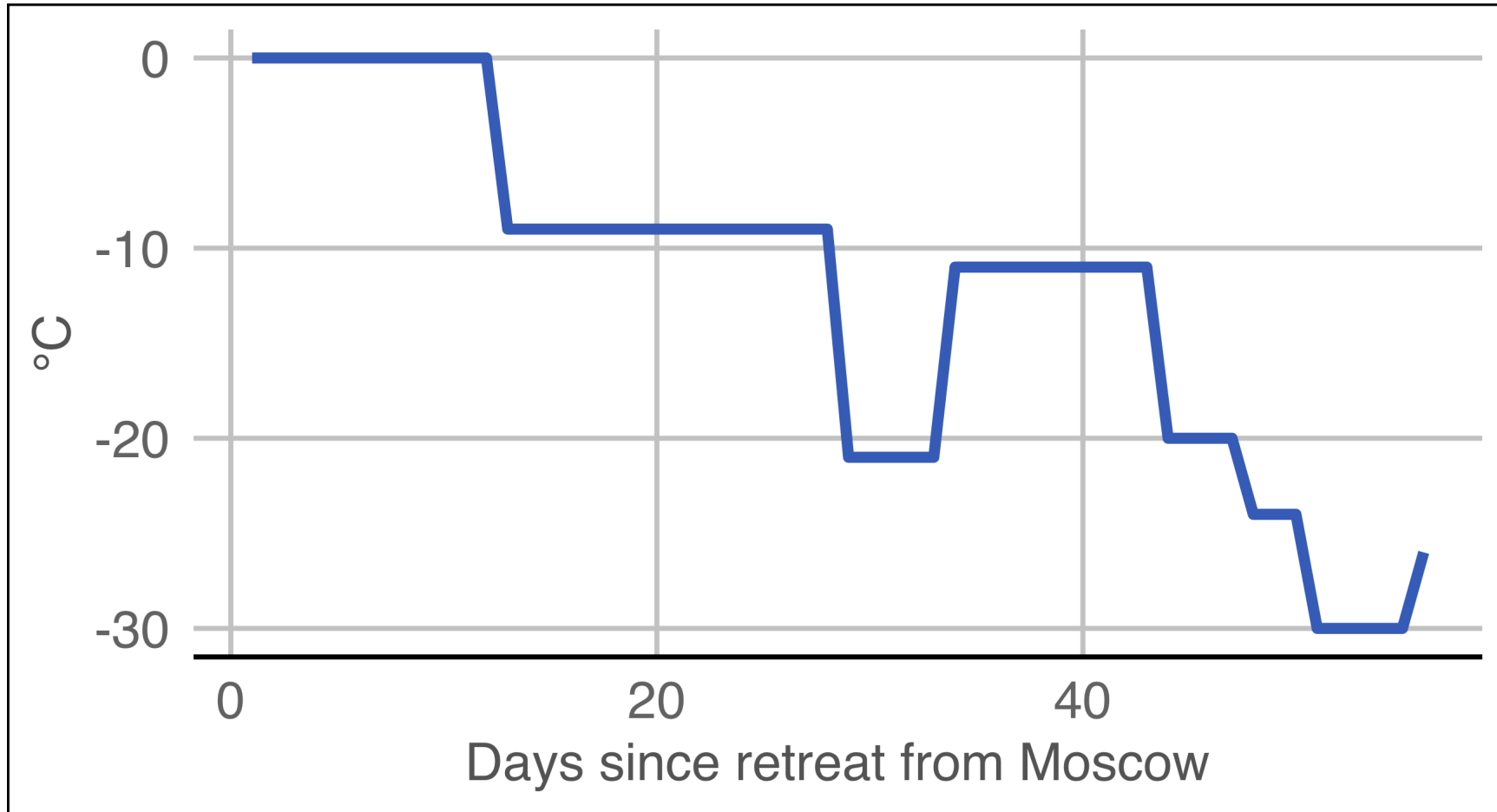


Long distance!



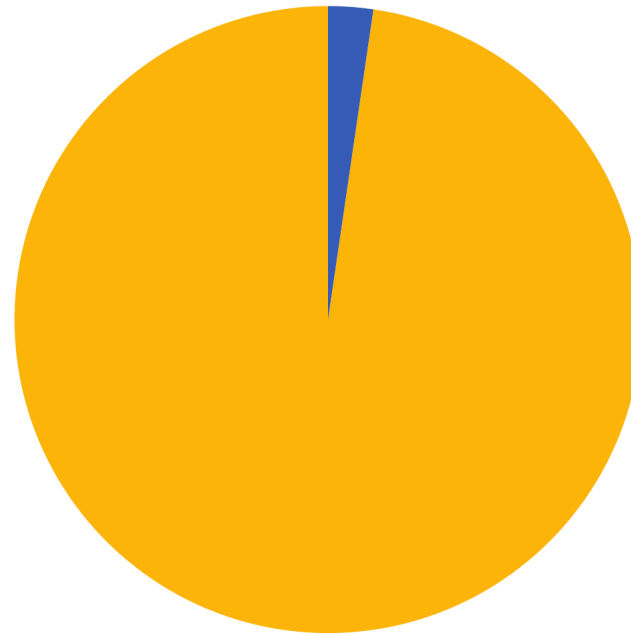
Moscow to Vilnius

Very cold!



Lots of people died!

Napoleon's Grande Armée



■ Died ■ Survived

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 zones colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zones. 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. Chiers, 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 Davout qui avaient été détachés sur Minsk et Mohilow et ont rejoint vers Orscha ou Witebsk, avaient toujours marché avec l'armée.

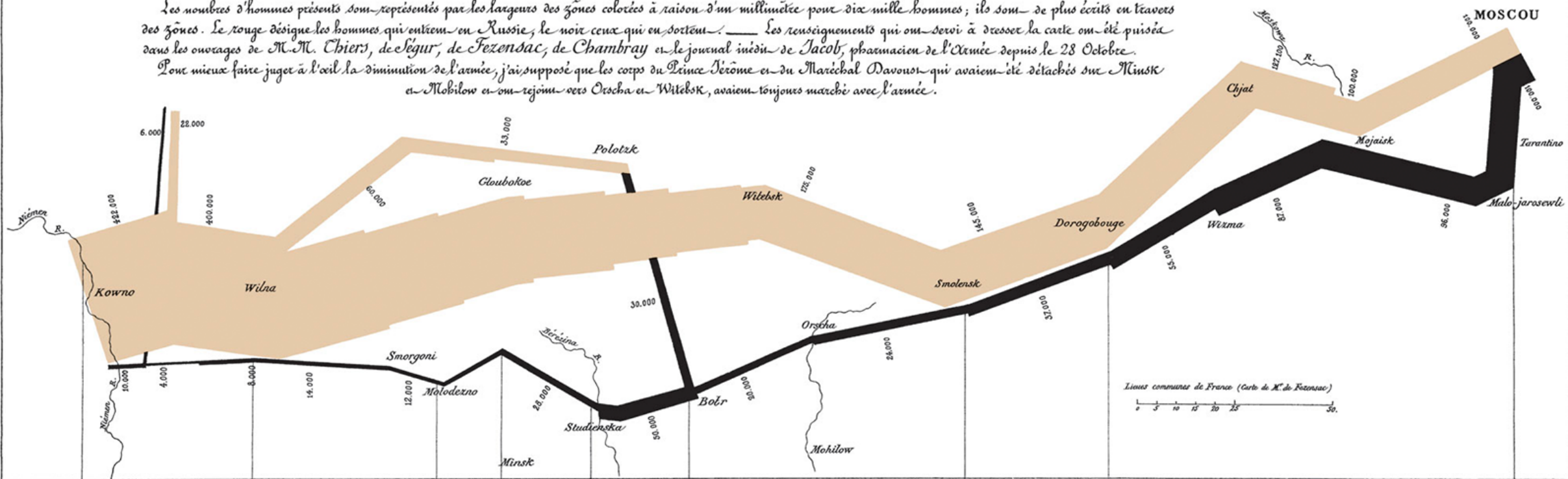
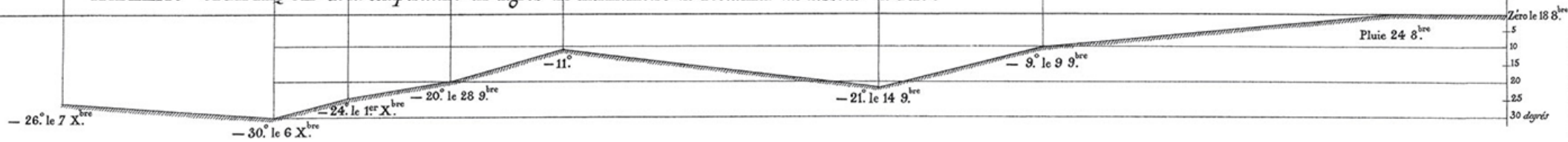
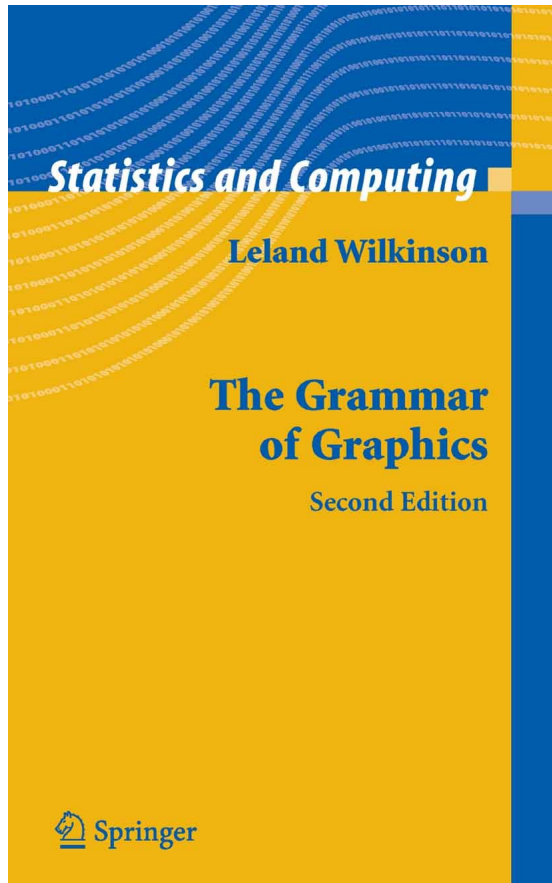


TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.



Les Cosaques passent au galop le Niémar gelé.

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

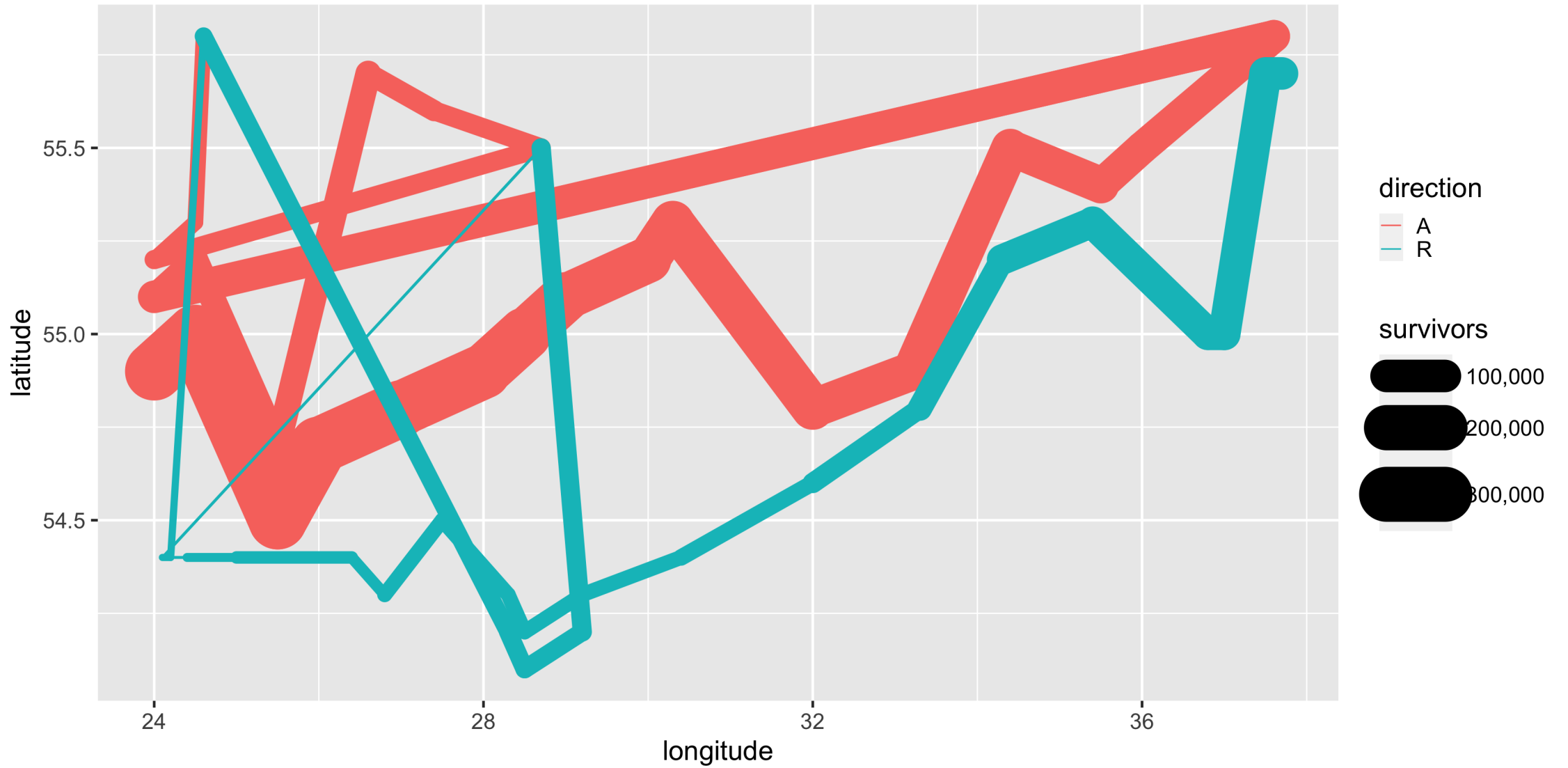
```
ggplot(data = DATA) +  
  GEOM_FUNCTION(mapping = aes(AESTHETIC MAPPINGS))
```

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

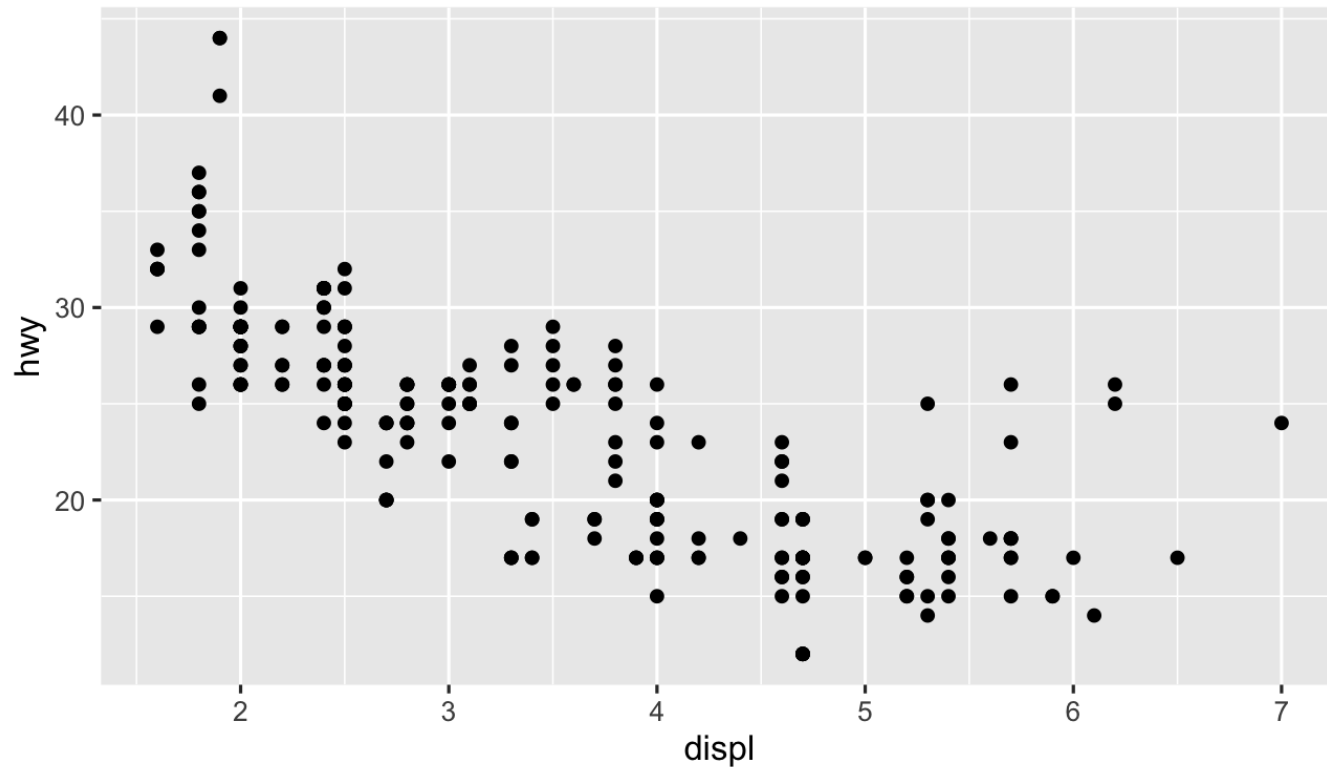
This is a dataset named `troops`:

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

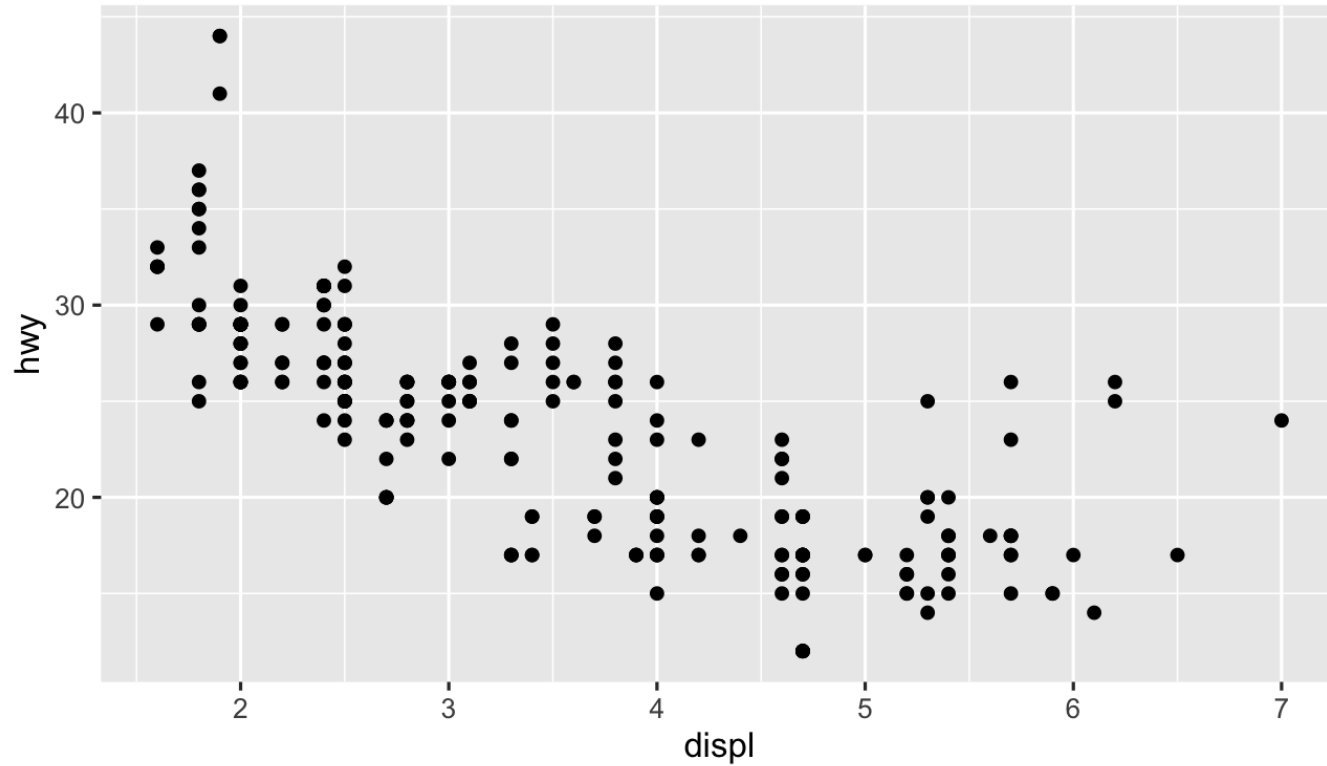
```
ggplot(data = troops) +  
  geom_path(mapping = aes(x = longitude,  
                           y = latitude,  
                           color = direction,  
                           size = survivors))
```



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

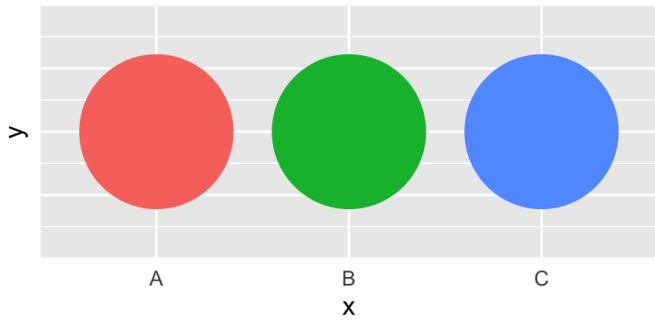


Heavy cars with better mileage?

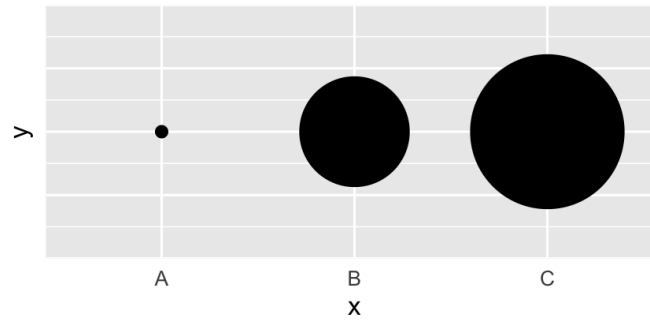


Aesthetics

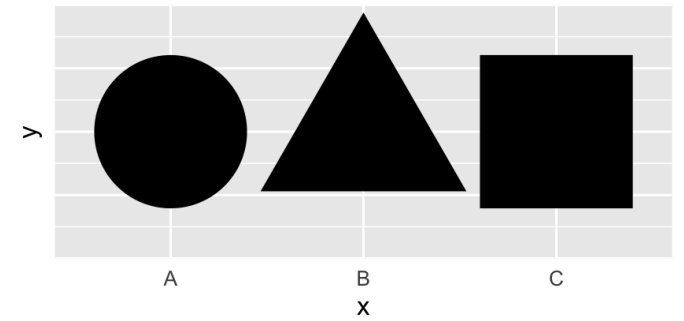
color (discrete)



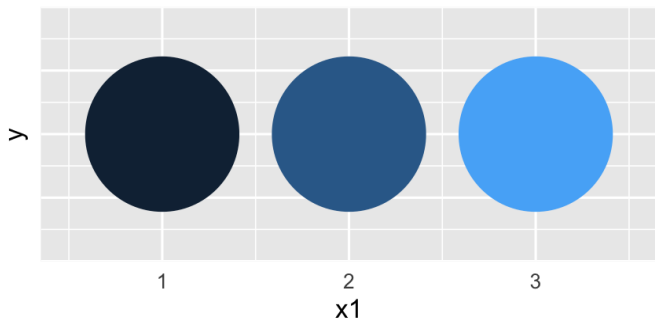
size



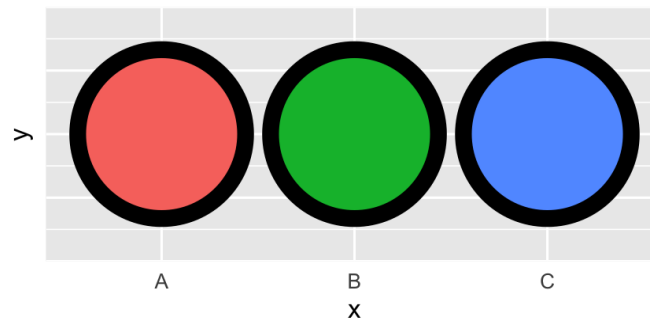
shape



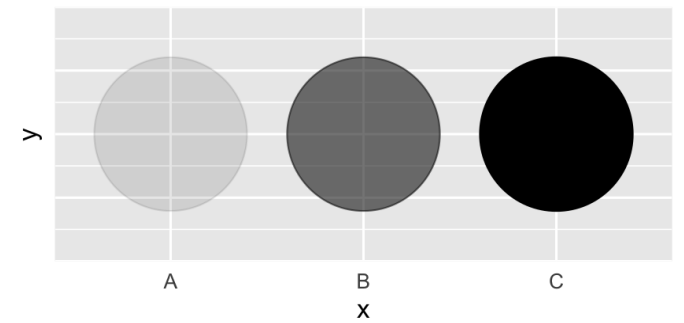
color (continuous)



fill



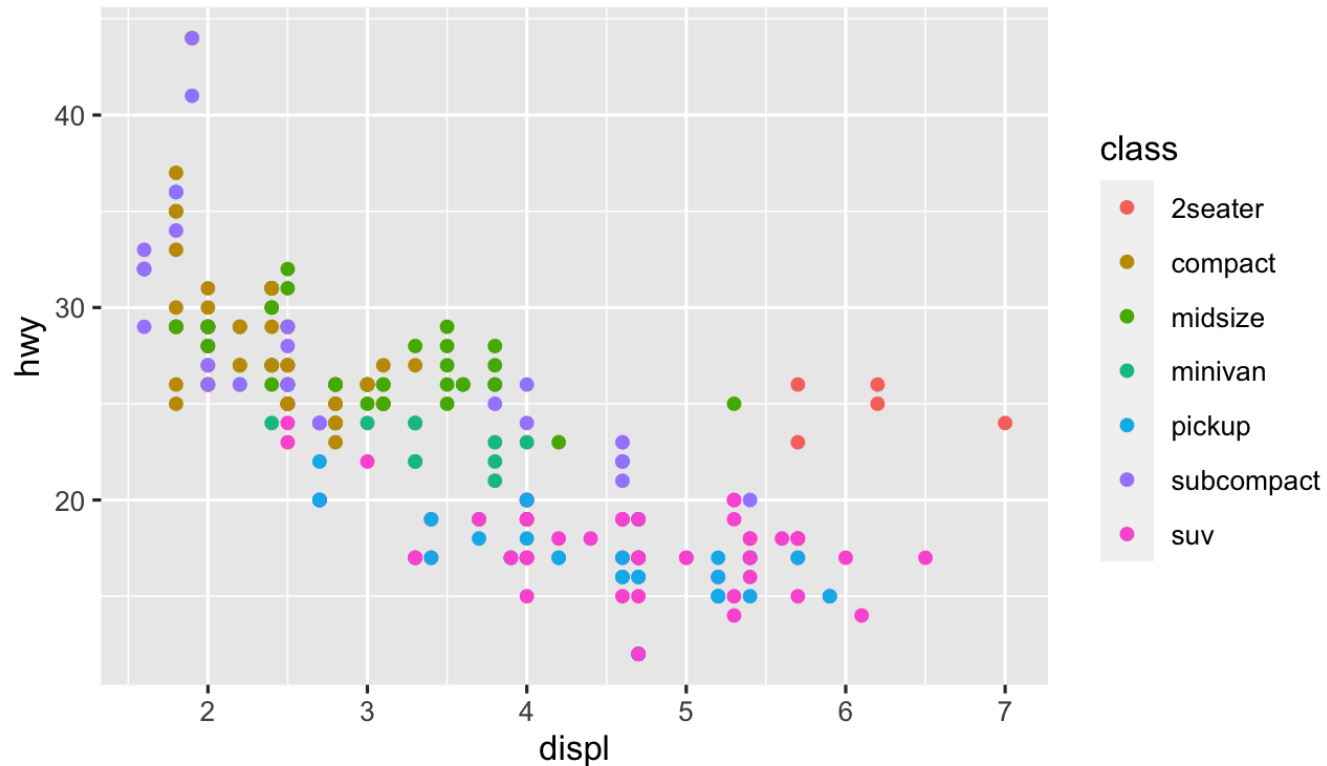
alpha



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

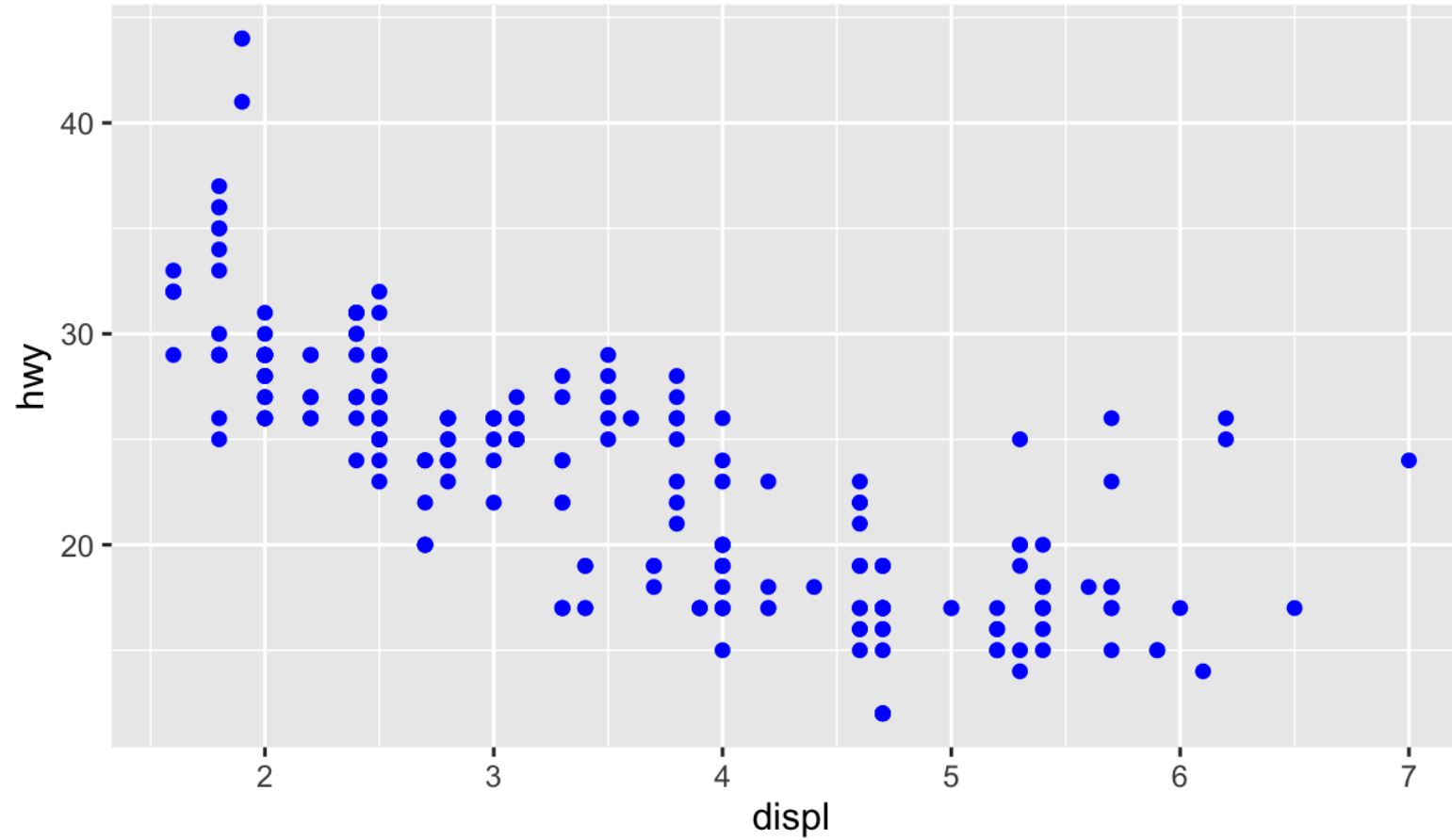
Add color, size, alpha, and shape aesthetics to your graph.

Experiment!

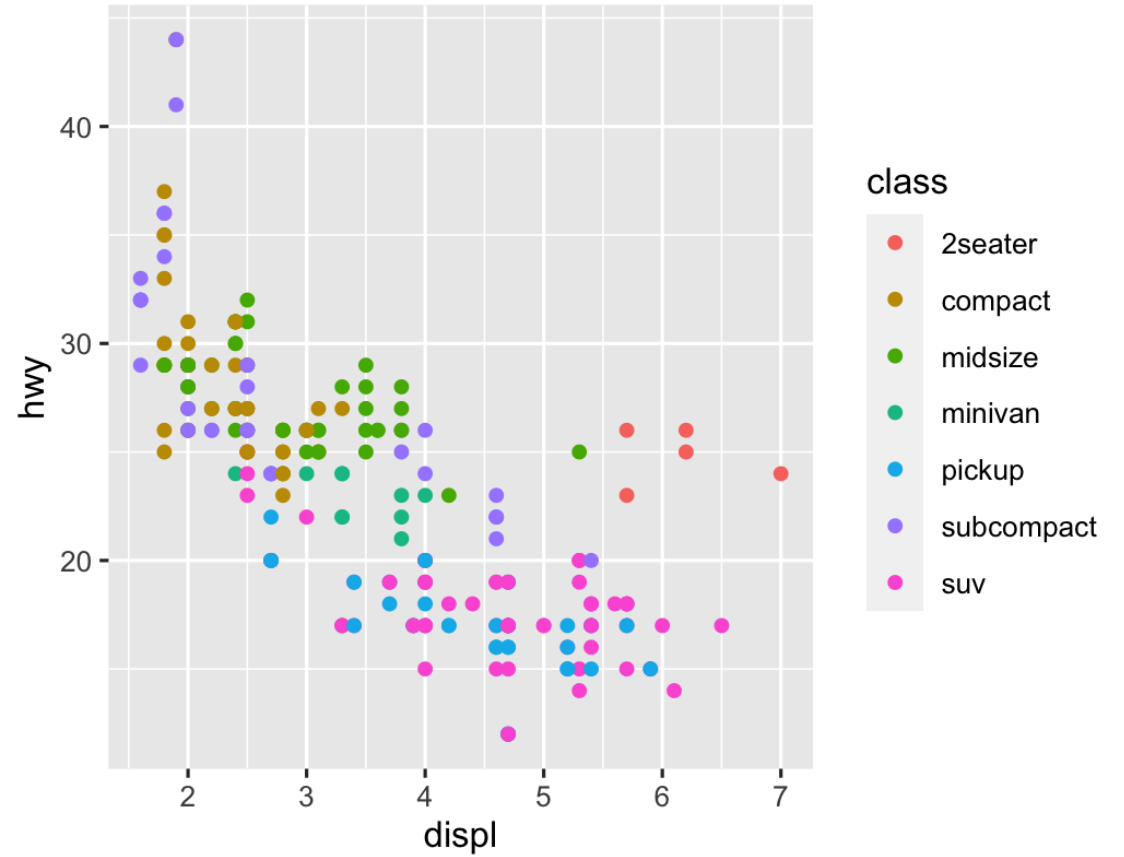
Do different things happen when you map aesthetics to discrete and continuous variables?

What happens when you use more than one aesthetic?

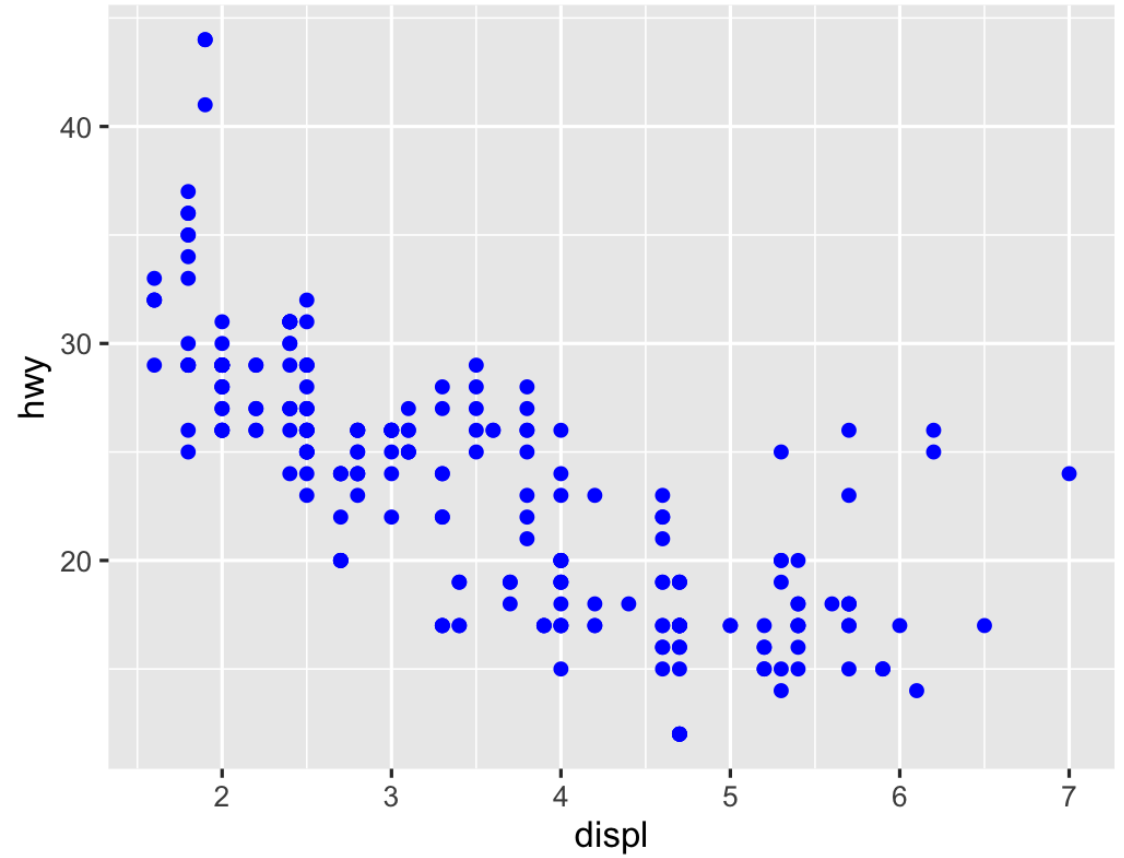
How would you make this plot?



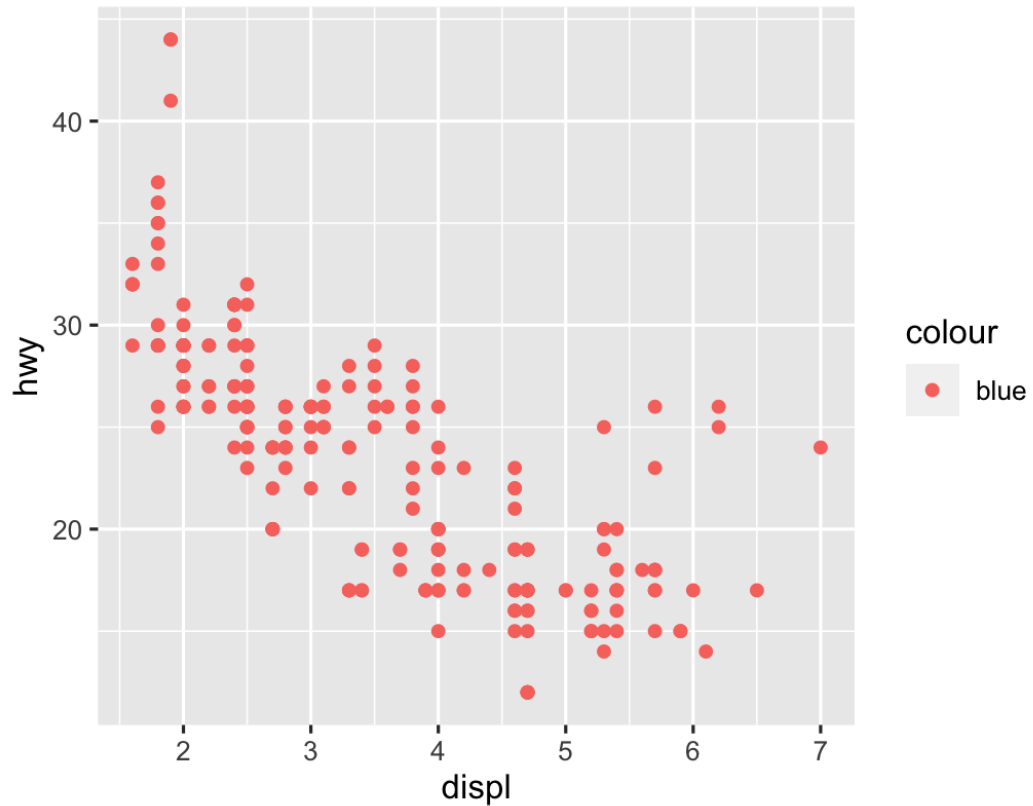
```
ggplot(mpg) +  
  geom_point(aes(x = displ, y = hwy,  
                 color = class))
```



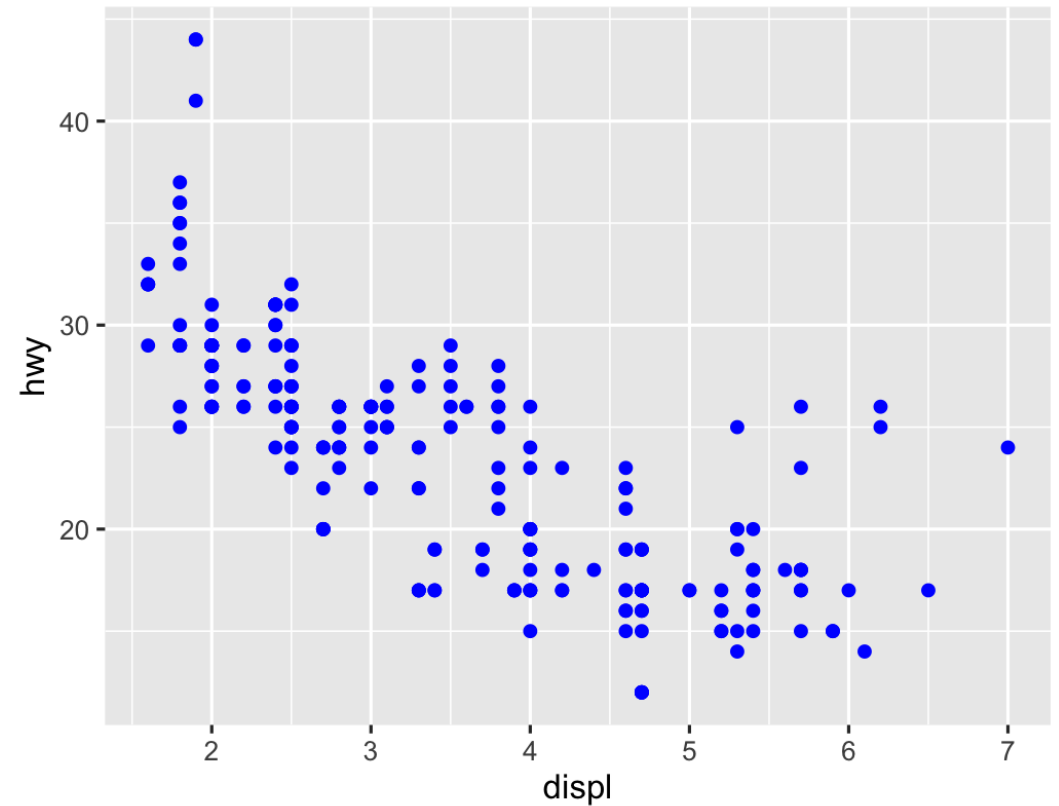
```
ggplot(mpg) +  
  geom_point(aes(x = displ, y = hwy),  
             color = "blue")
```



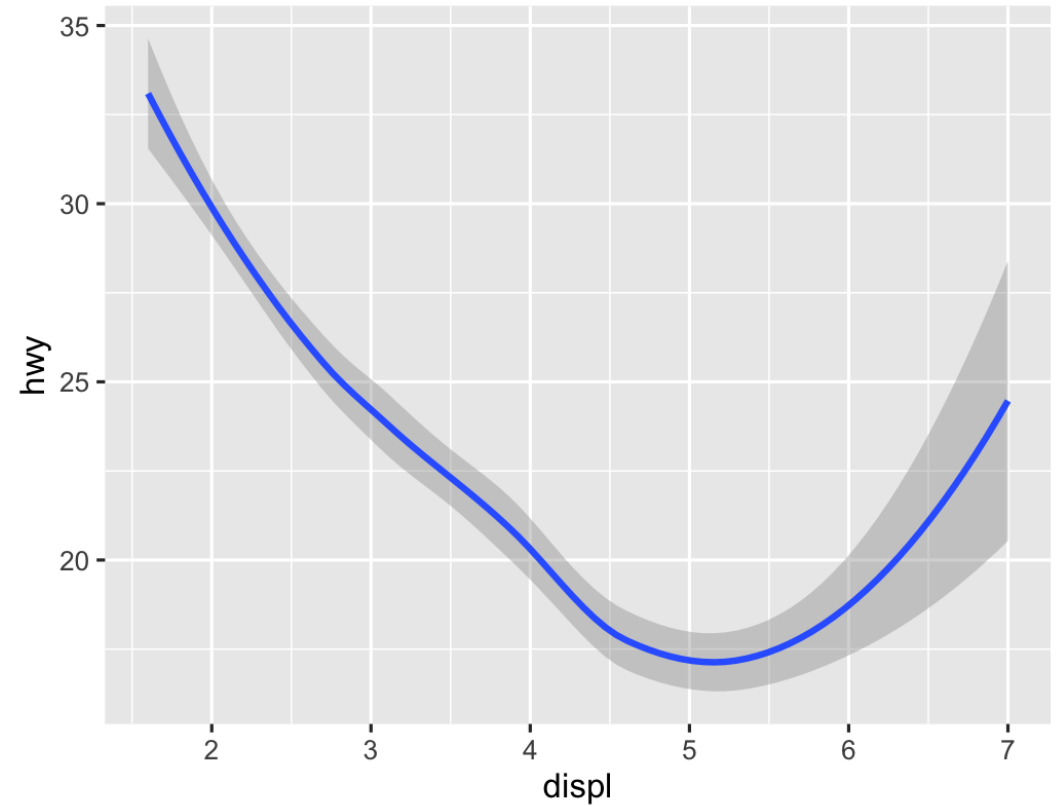
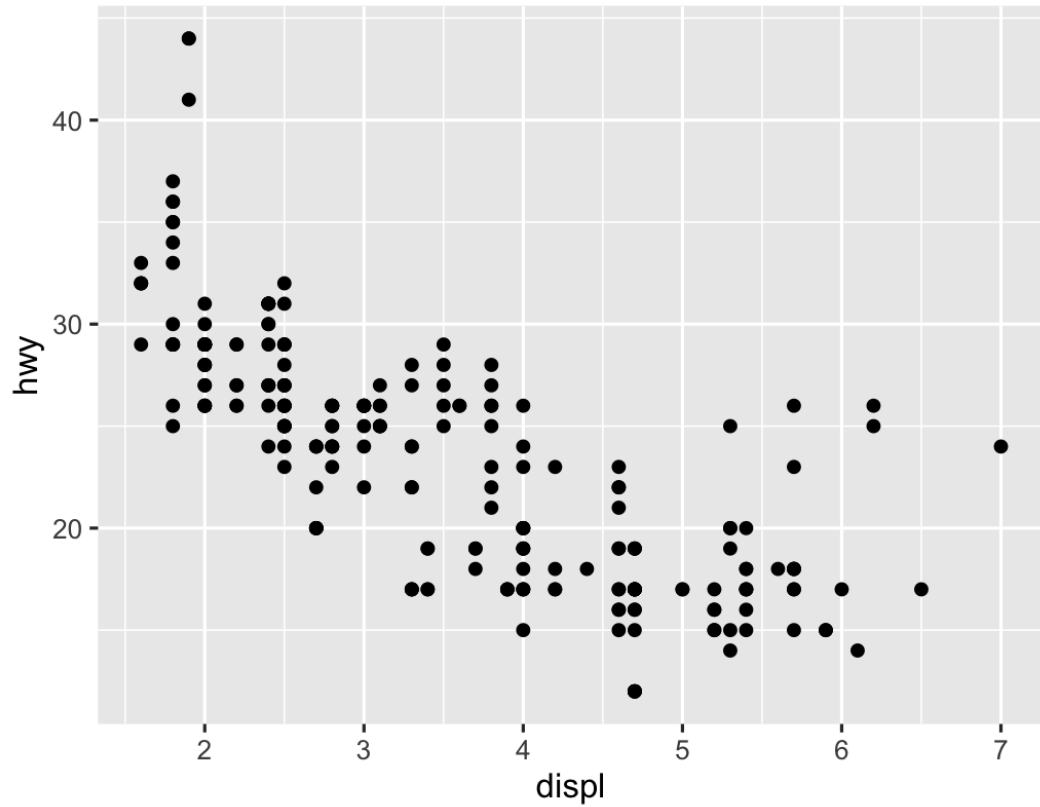

```
ggplot(mpg) +  
  geom_point(aes(x = displ, y = hwy,  
                 color = "blue"))
```



```
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
	<code>geom_col()</code>	Bar charts
<i>text</i>	<code>geom_text()</code>	Text
	<code>geom_point()</code>	Points
	<code>geom_boxplot()</code>	Boxplots
	<code>geom_sf()</code>	Maps

Possible geoms

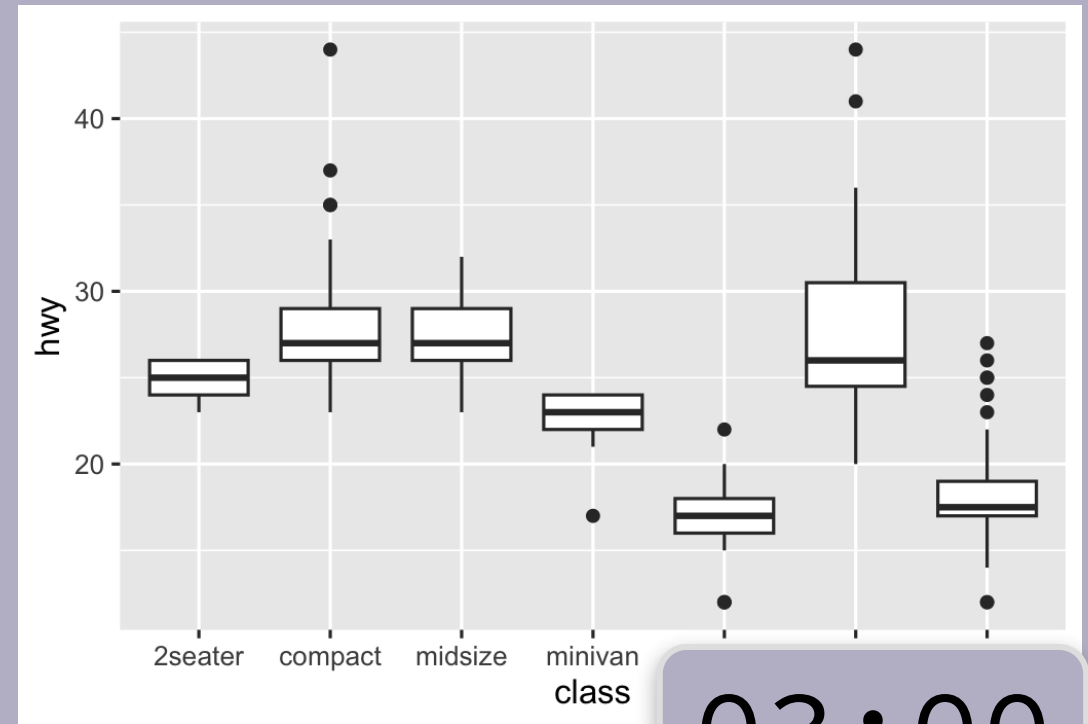
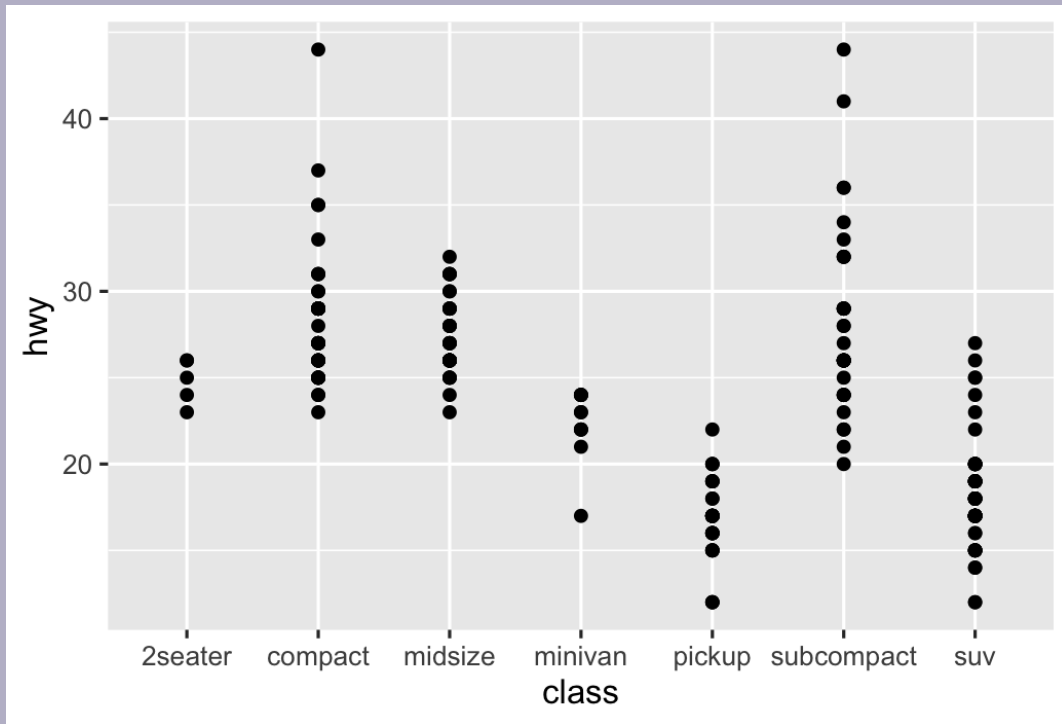
There are dozens of possible geoms!

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

Also see the ggplot cheatsheet

Your turn #3

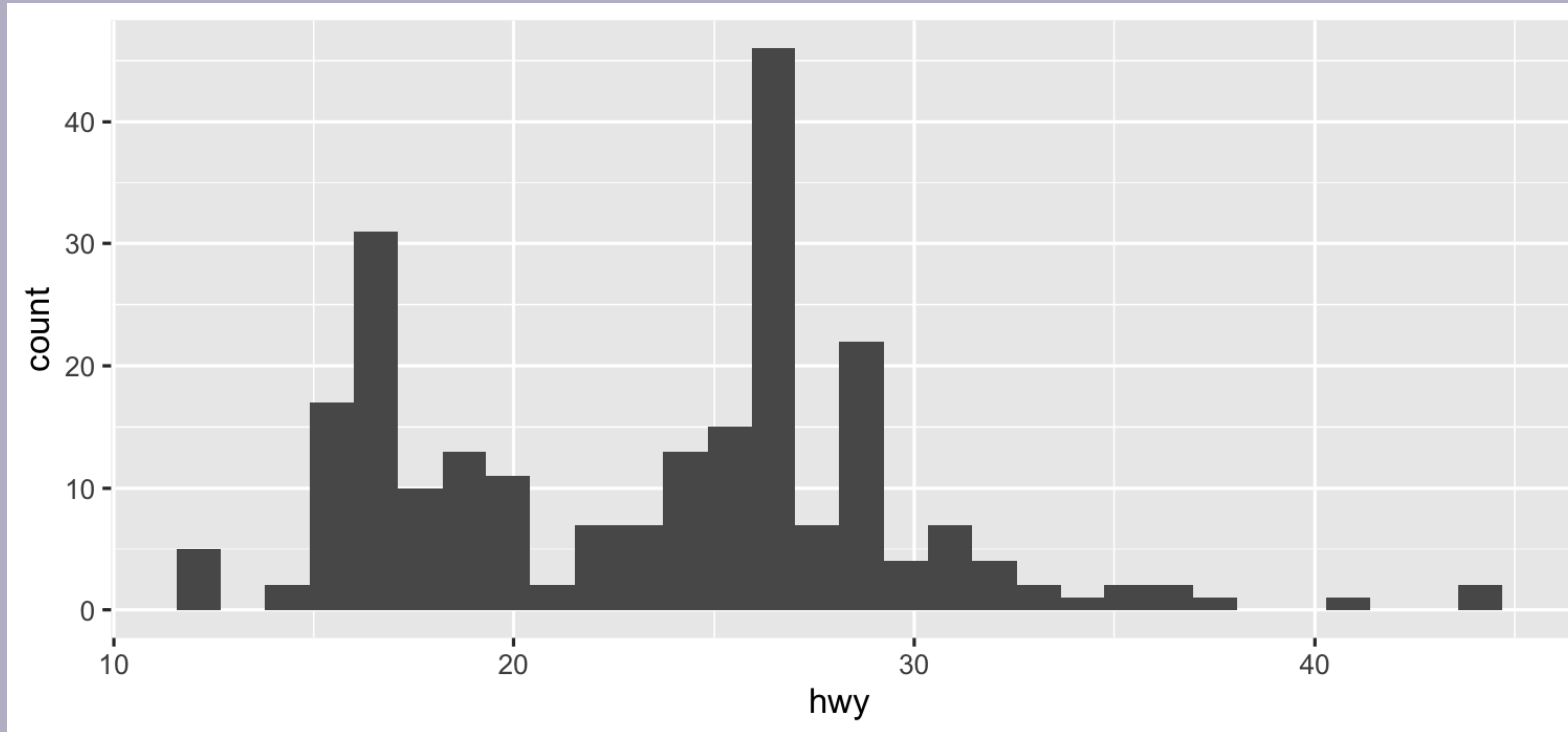
Replace this scatterplot with boxplots. Use the cheatsheet.



03:00

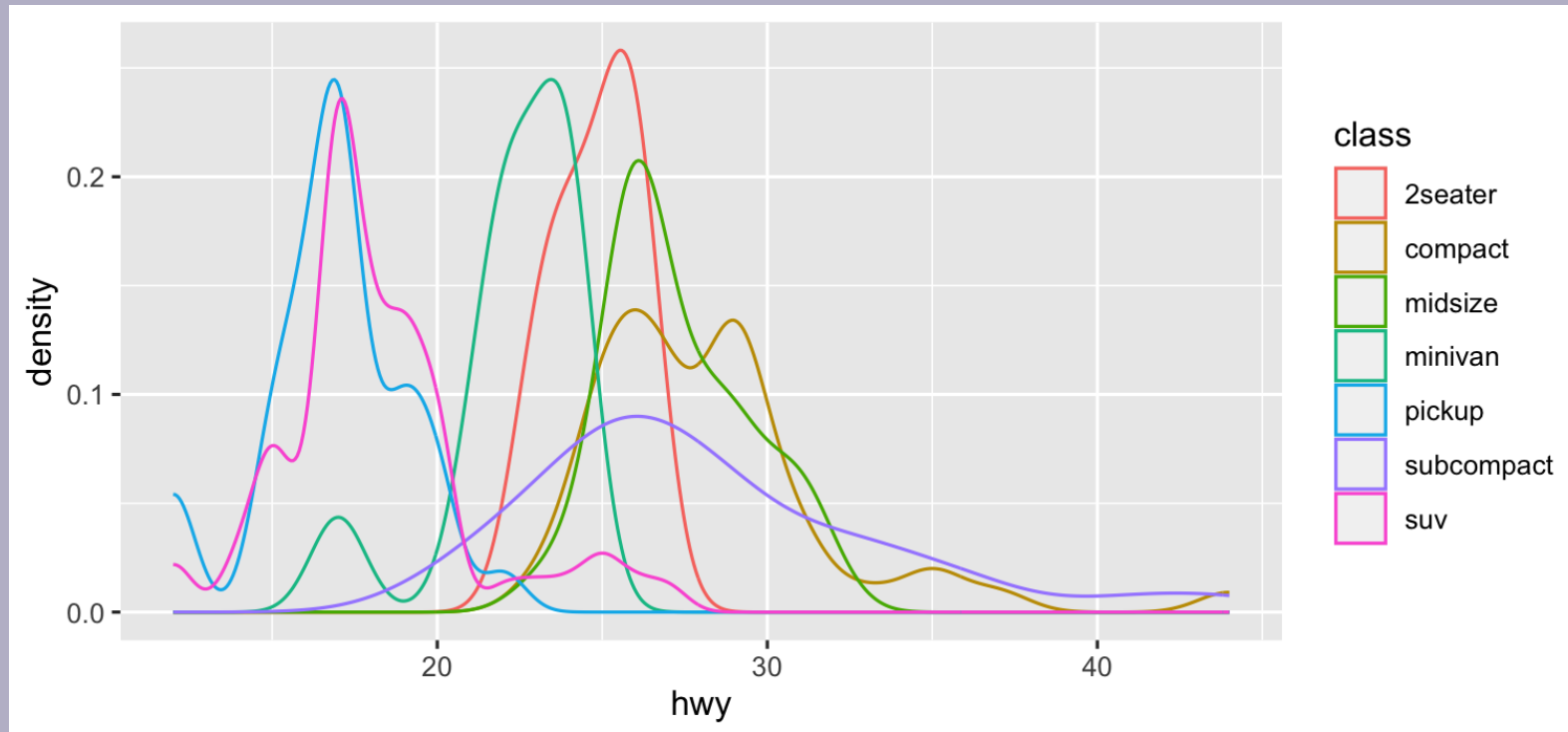
Your turn #4

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

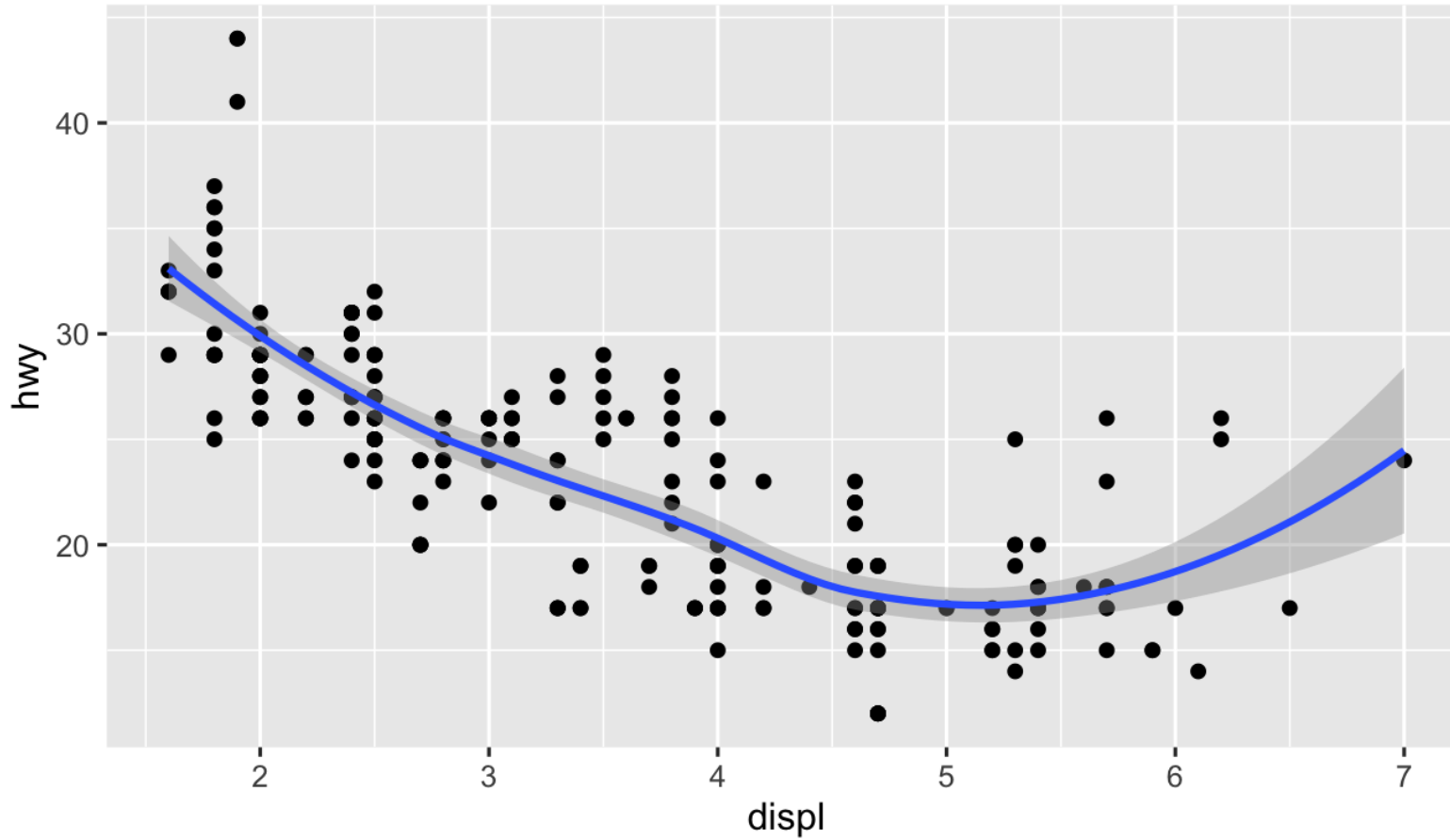


Your turn #5

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.

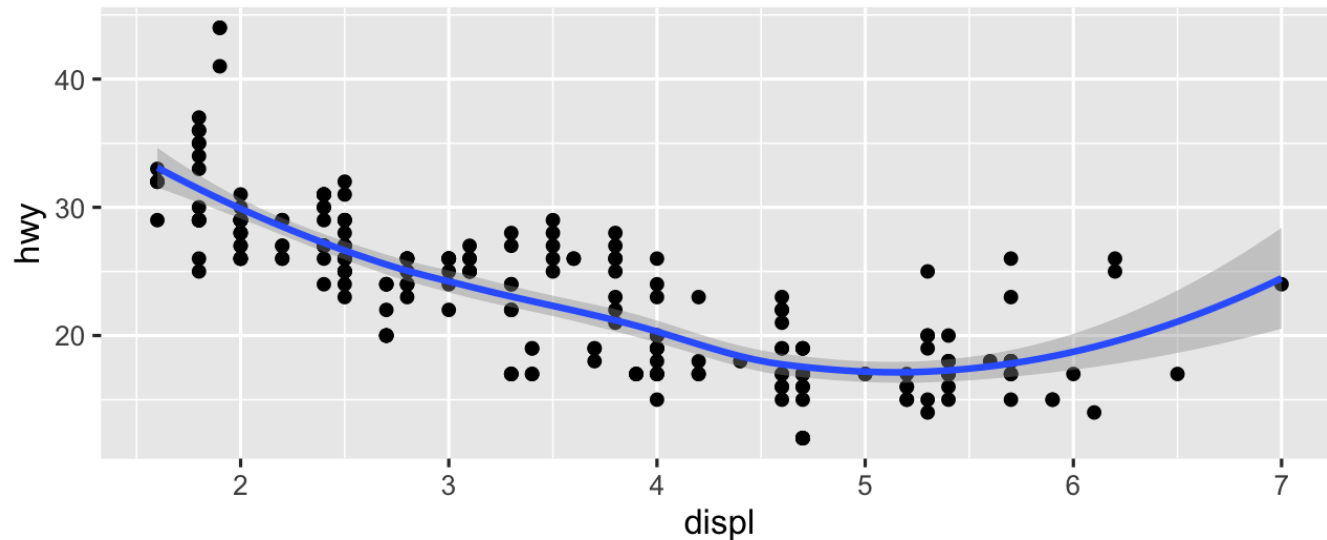
```
ggplot(data = mpg) +  
  geom_point(mapping = aes(x = displ, y = hwy)) +  
  geom_smooth(mapping = aes(x = displ, y = hwy))
```

02:00

Global vs. local

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

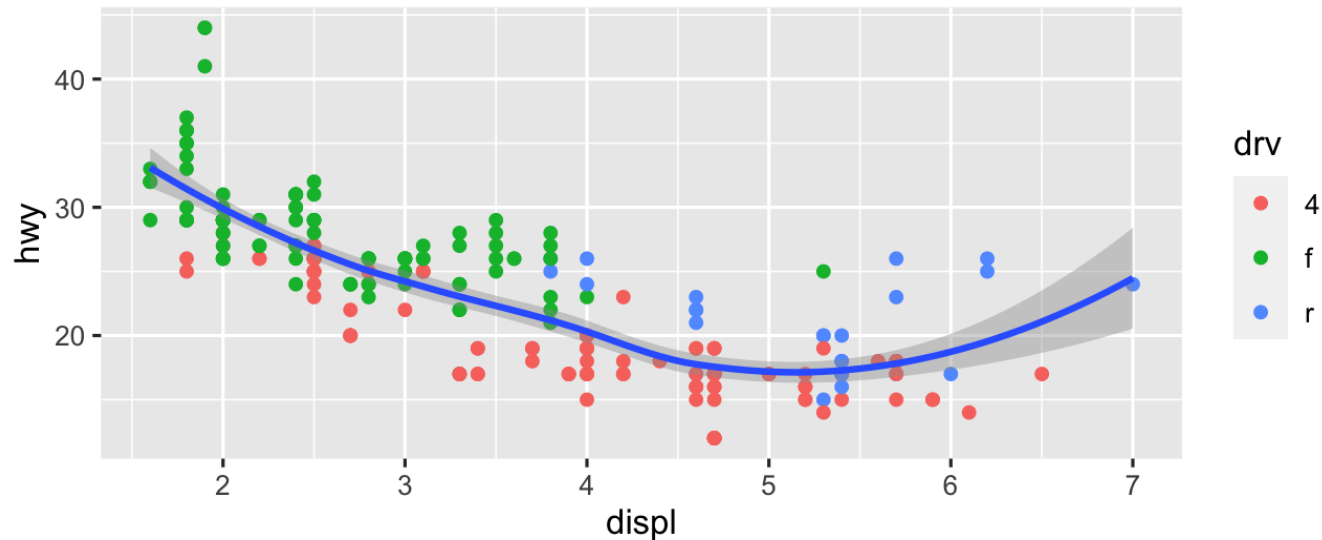
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point() +  
  geom_smooth()
```



Global vs. local

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

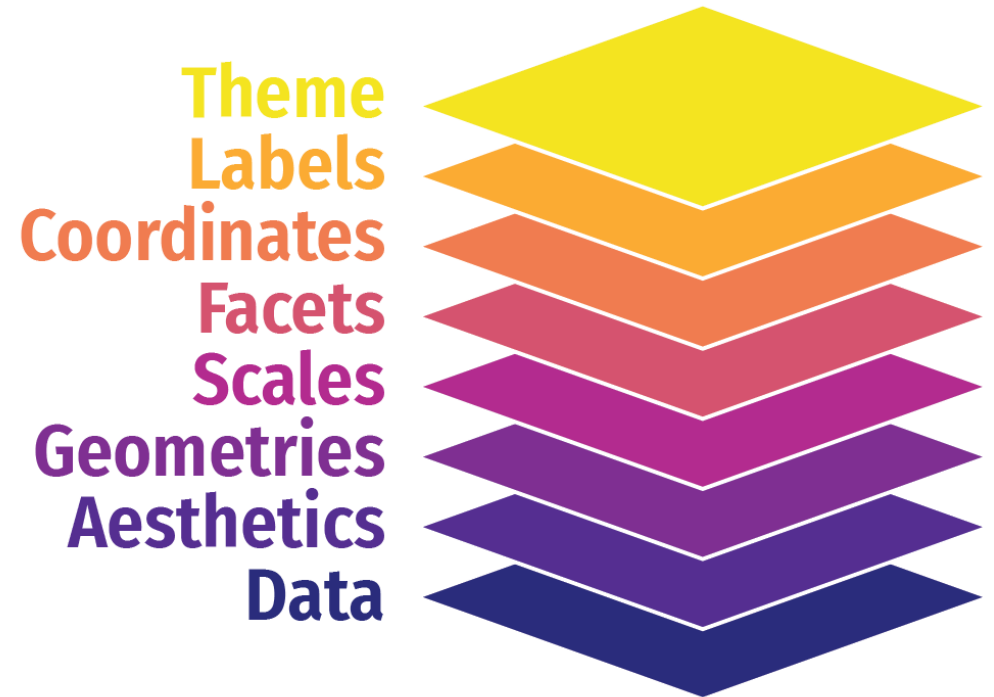
```
ggplot(data = mpg, mapping = aes(x = displ, y = hwy)) +  
  geom_point(mapping = aes(color = drv)) +  
  geom_smooth()
```



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

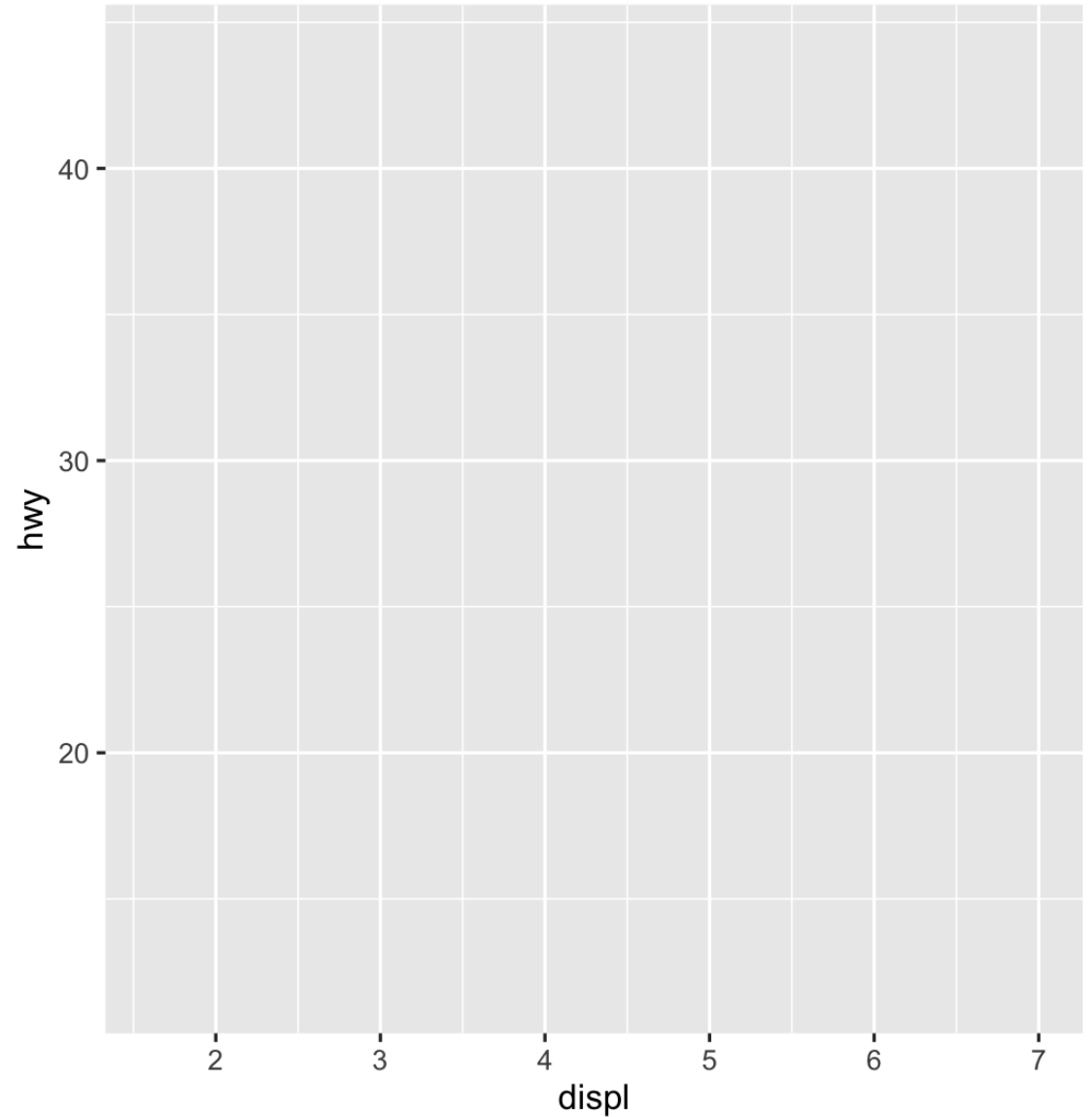


Putting it all together

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

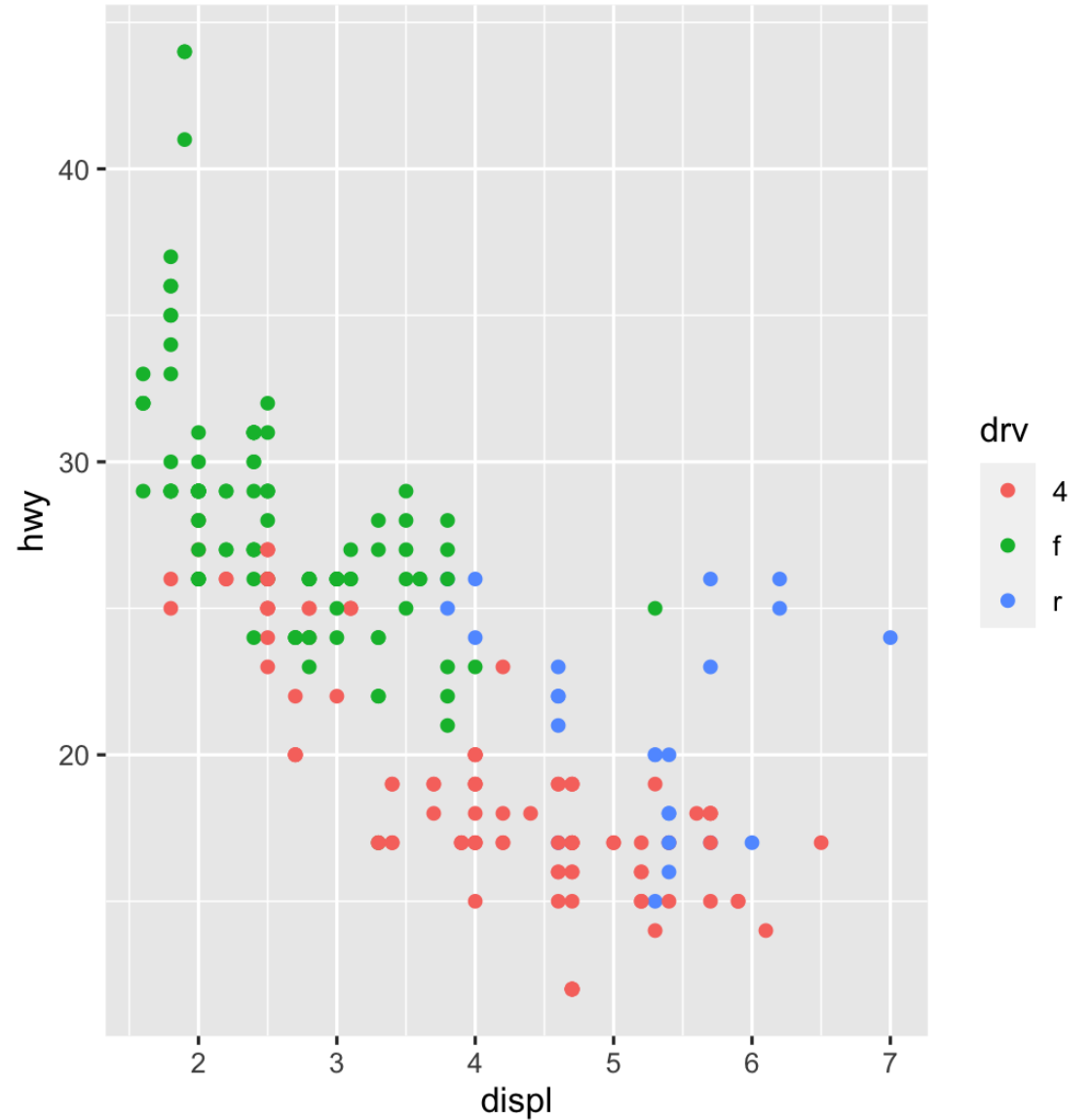
Start with data and aesthetics

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



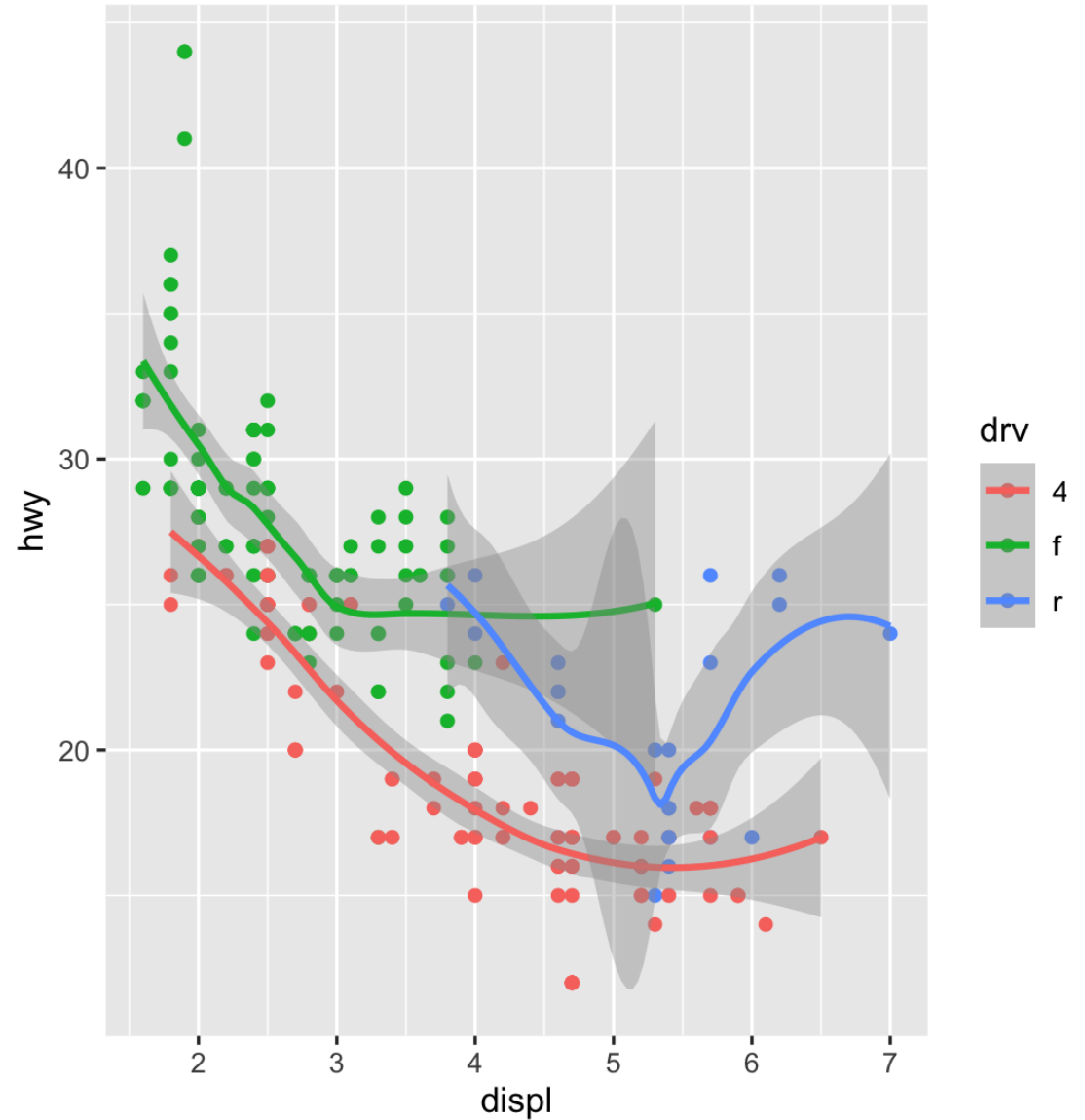
Add a point geom

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



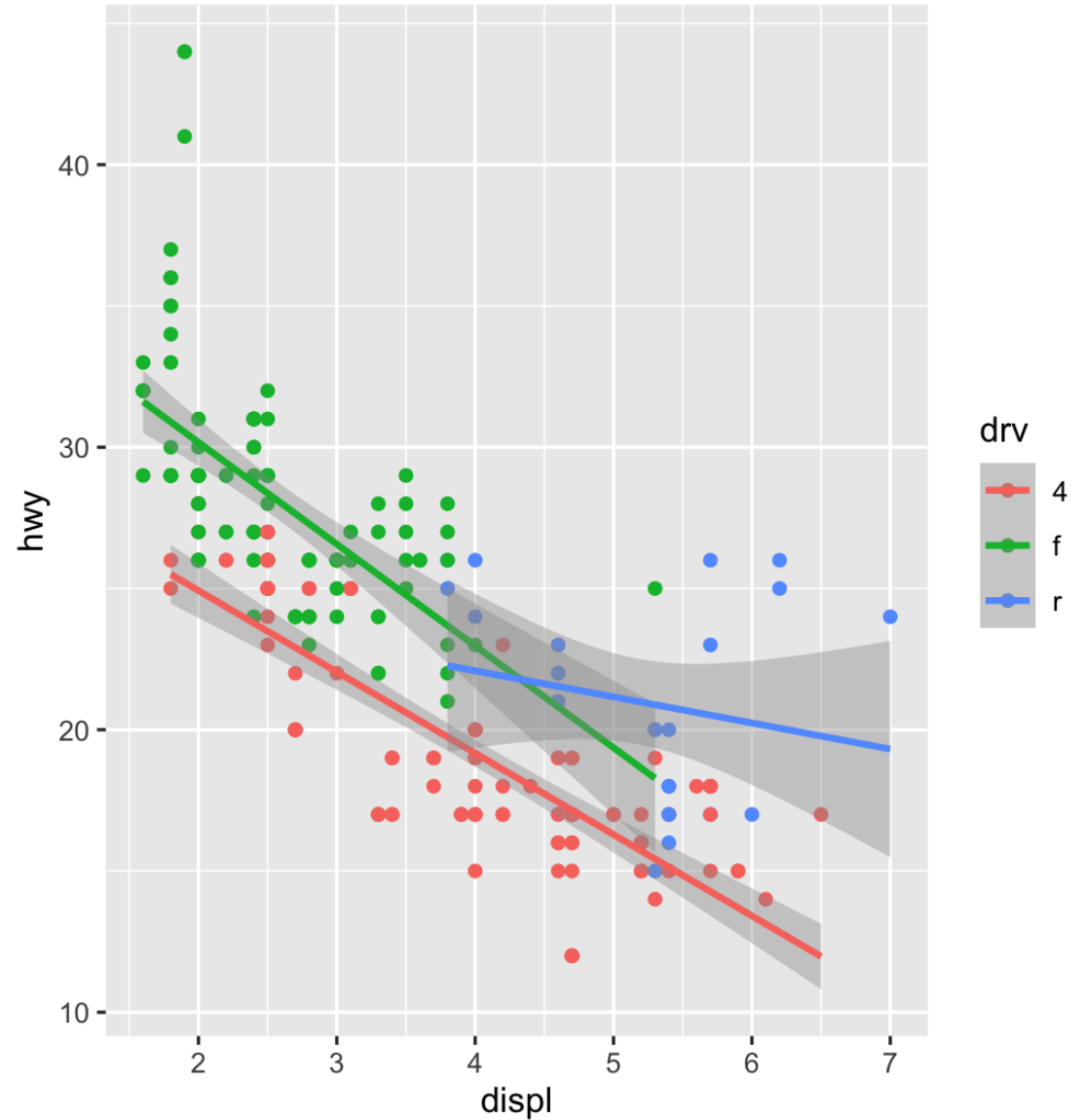
Add a smooth geom

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



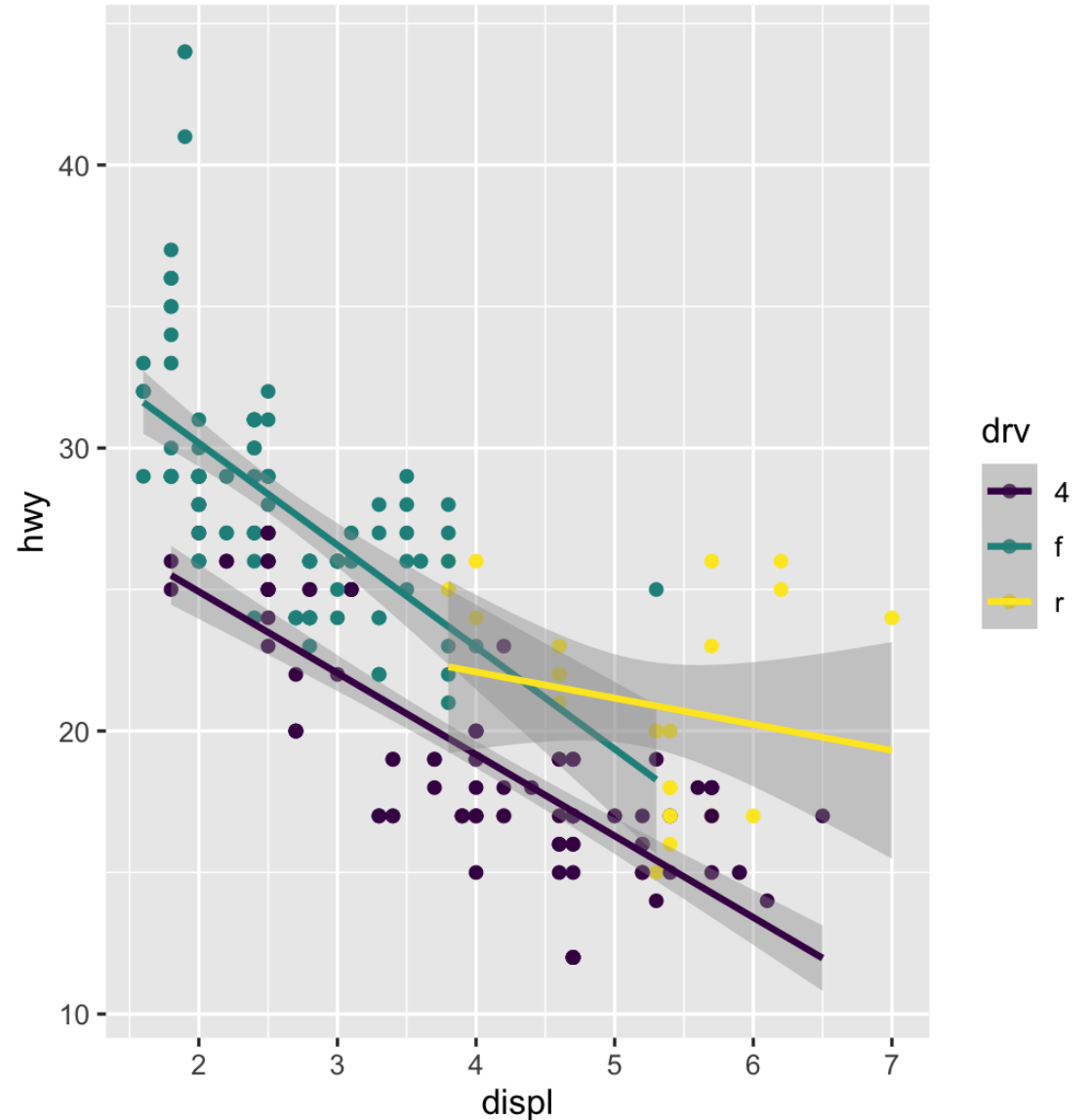
Make it straight

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



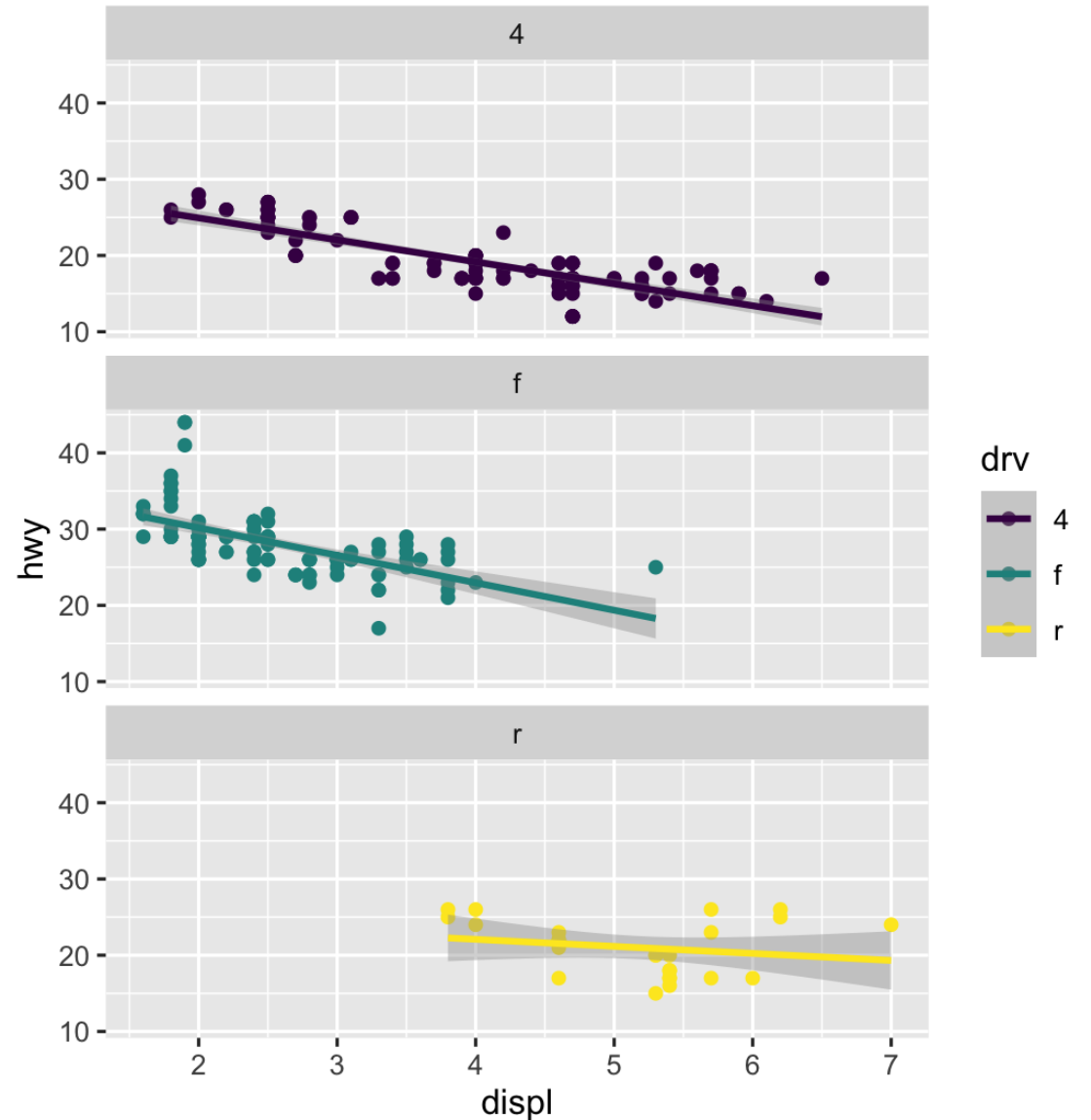
Use a viridis color scale

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



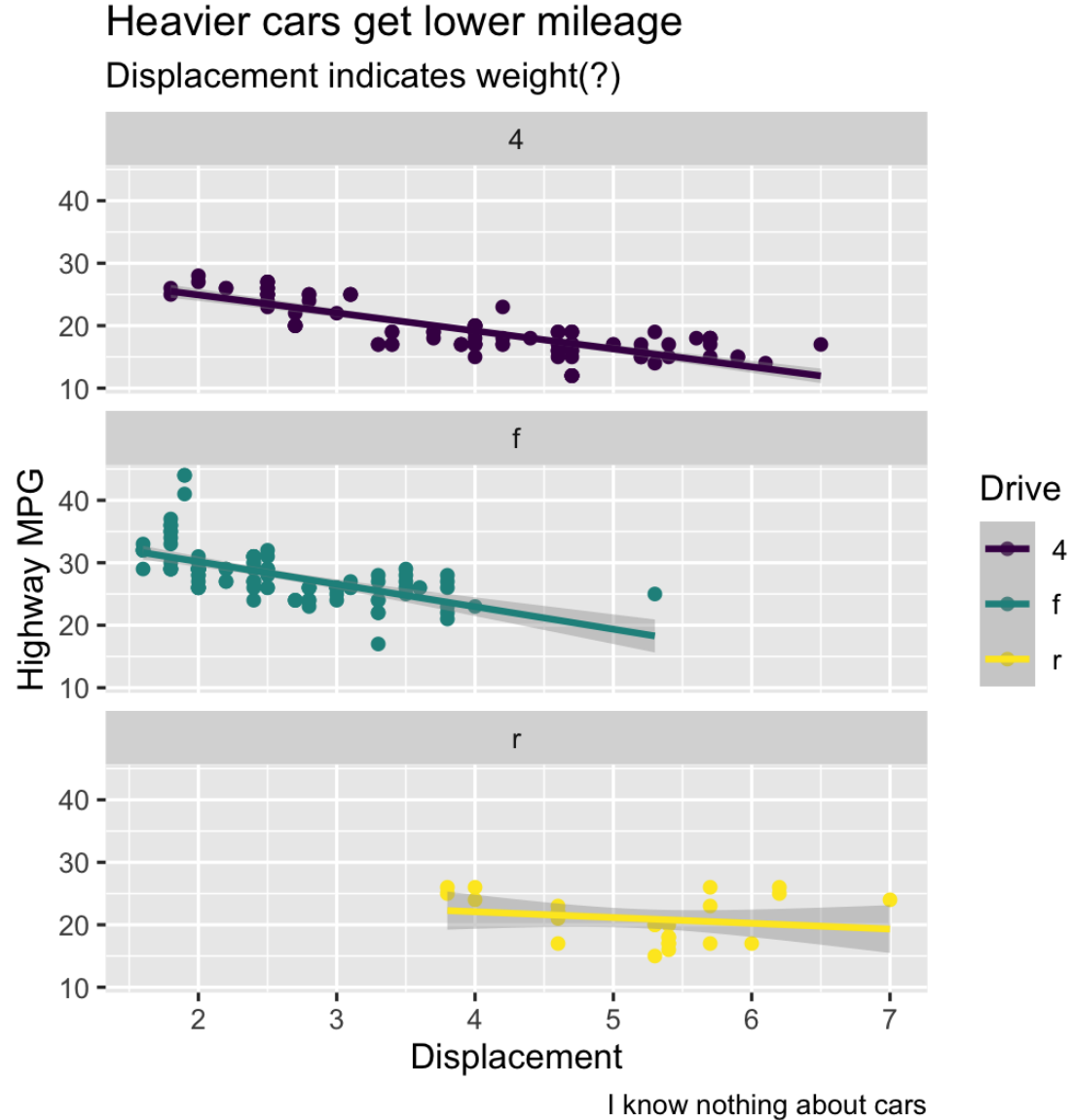
Facet by drive

```
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)
```



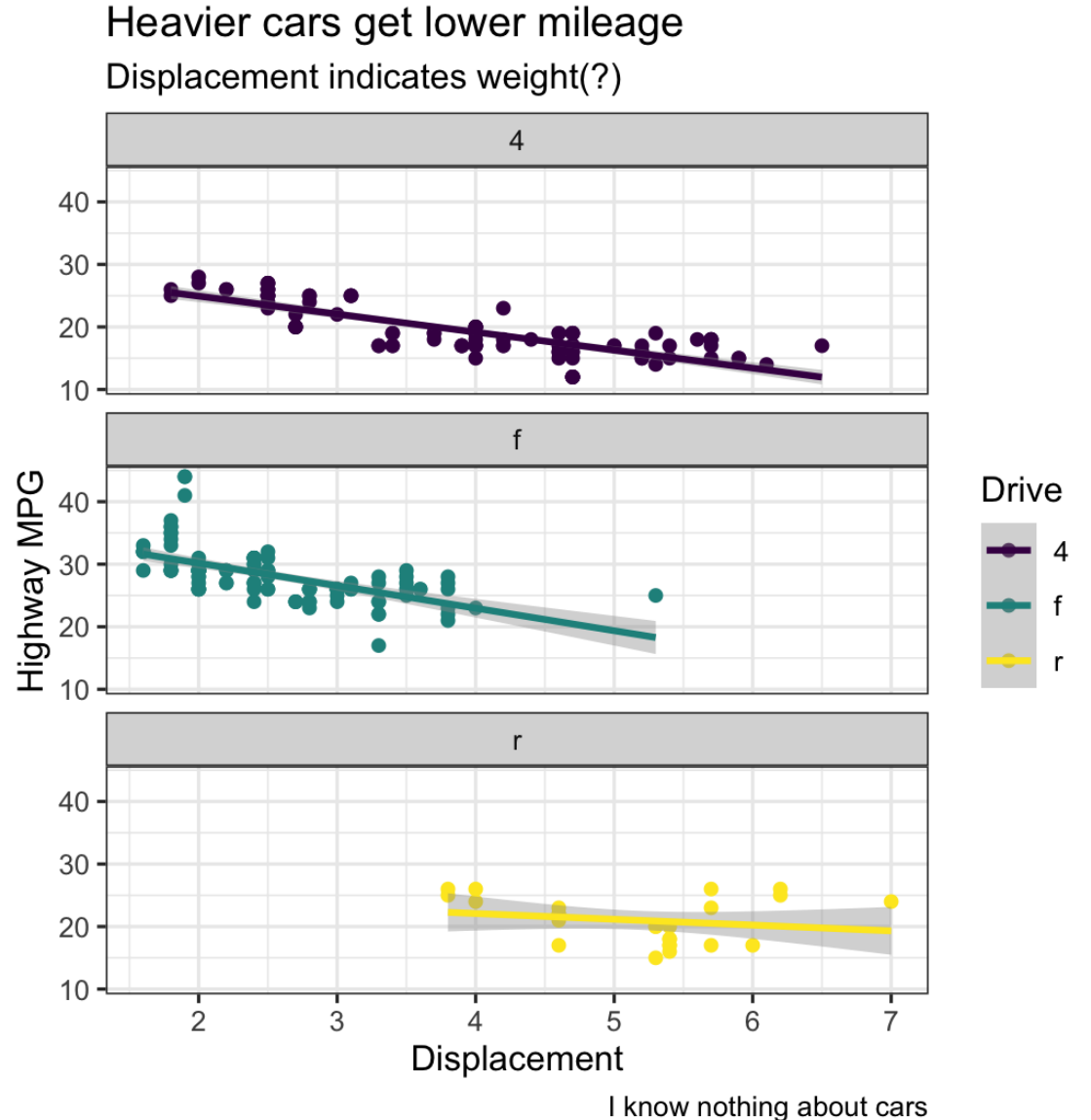
Add labels

```
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 mileage",
       subtitle = "Displacement indicates weight",
       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 mileage",
       subtitle = "Displacement indicates weight",
       caption = "I know nothing about cars")
  theme_bw()
```

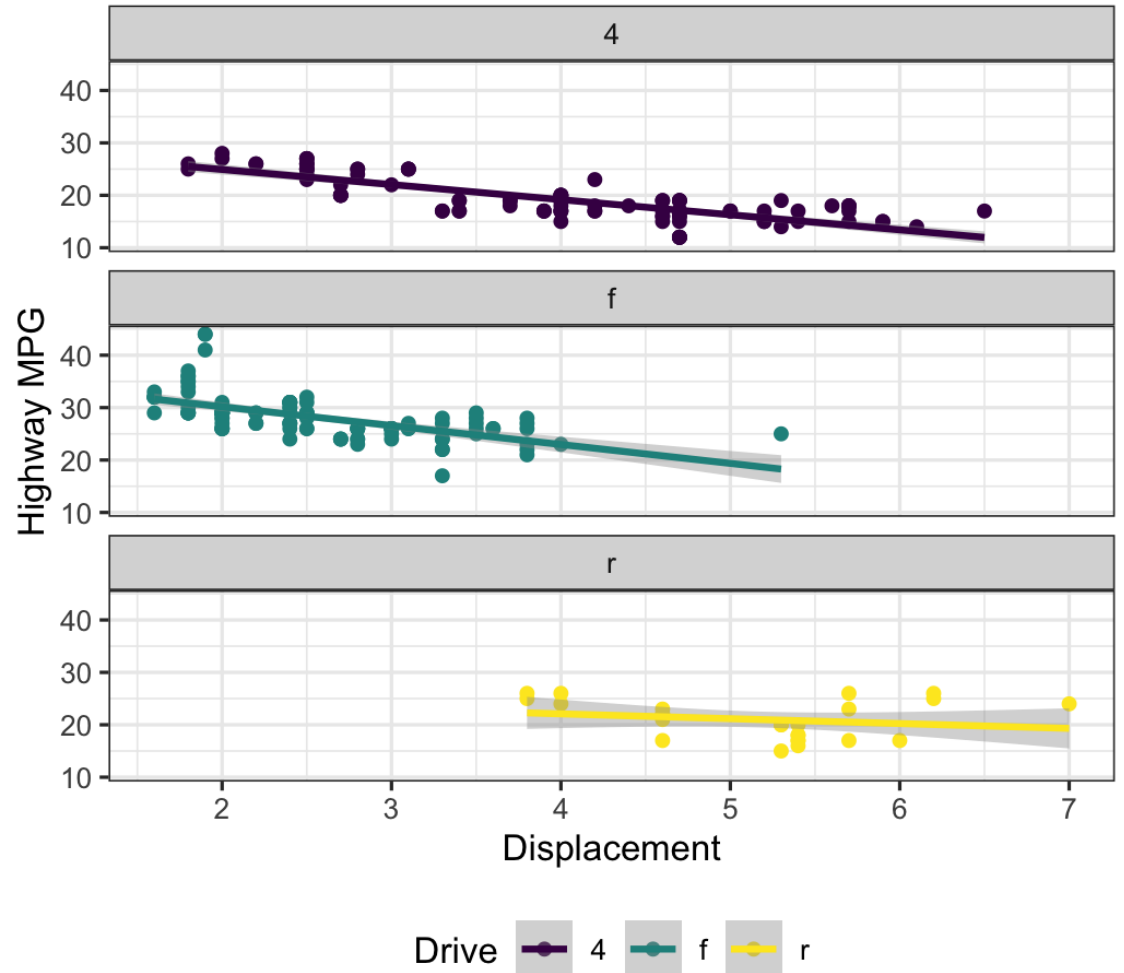


Modify the theme

```
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 mileage",
       subtitle = "Displacement indicates weight",
       caption = "I know nothing about cars") +
  theme_bw() +
  theme(legend.position = "bottom",
       plot.title = element_text(face = "bold"))
```

Heavier cars get lower mileage

Displacement indicates weight(?)



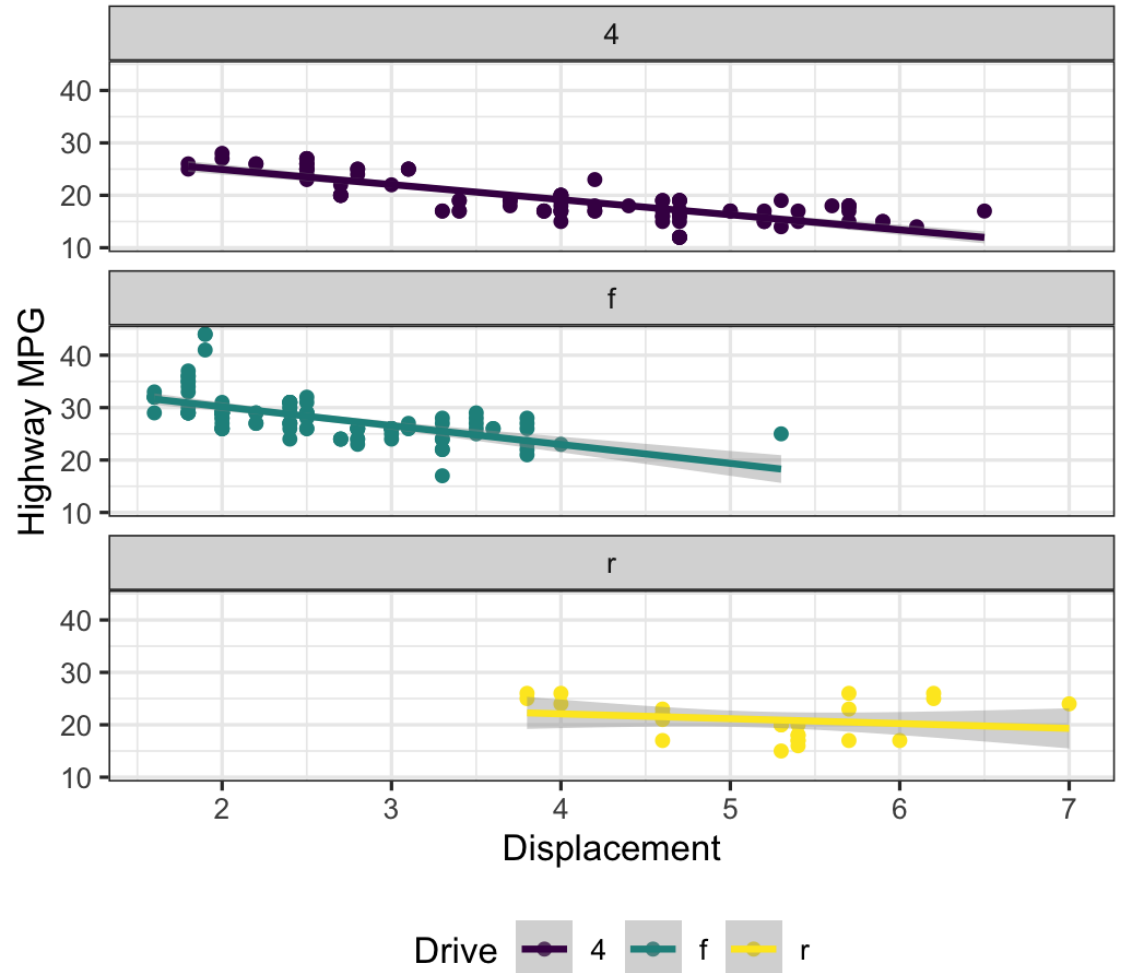
I know nothing about cars

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 mileage",
       subtitle = "Displacement indicates weight",
       caption = "I know nothing about cars") +
  theme_bw() +
  theme(legend.position = "bottom",
       plot.title = element_text(face = "bold"))
```

Heavier cars get lower mileage

Displacement indicates weight(?)



I know nothing about cars

Next up

**Transforming and
manipulating data with dplyr**