

TODAY'S TOPICS

- Why data visualization
- Visual encoding
- An introduction to ggplot2: grammar of graphics

WHAT IS DATA VISUALIZATION?

- Visual representation of information to help people make sense of complex phenomena through data.
- Designed to enable exploration, analysis, or communication.

WHAT MAKES A
VISUALIZATION EFFECTIVE?

Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813.

Précisé 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 desser la carte ont été puisés dans les ouvrages de M. M. Chiers, de Séguir, 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 Mogilow et qui rejoignent Orsha en Witebsk, avaient toujours marché avec l'armée.

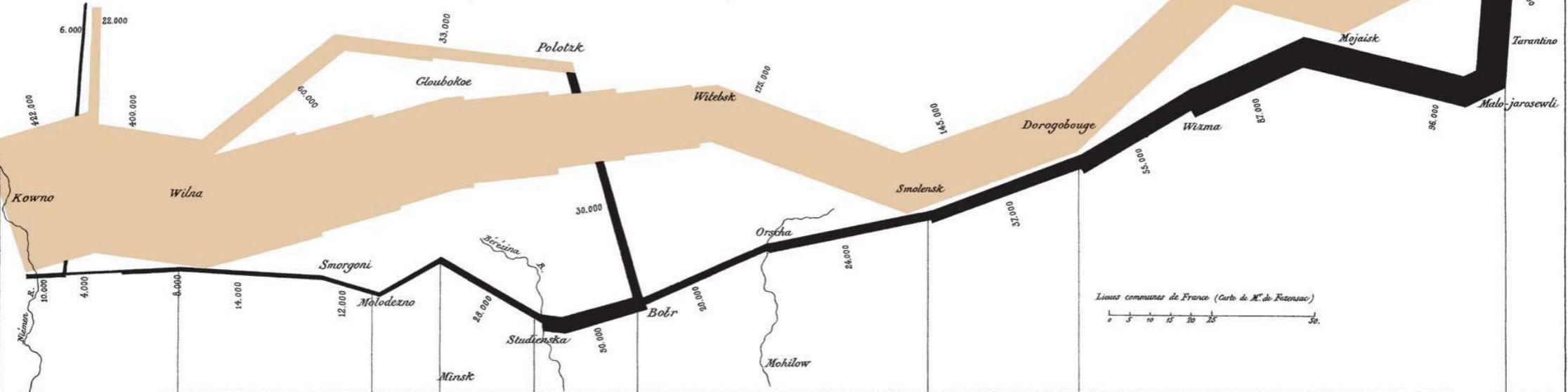
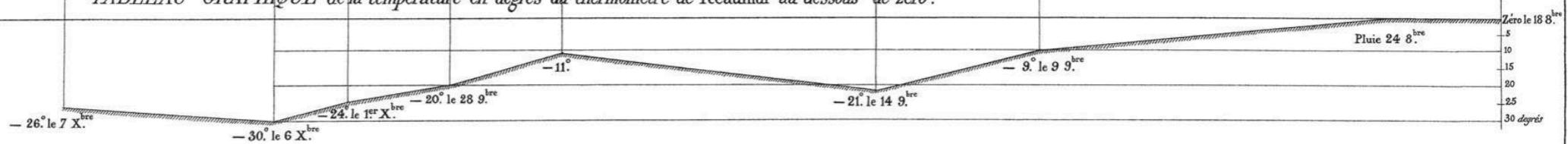


TABLEAU GRAPHIQUE de la température en degrés du thermomètre de Réaumur au dessous de zéro.

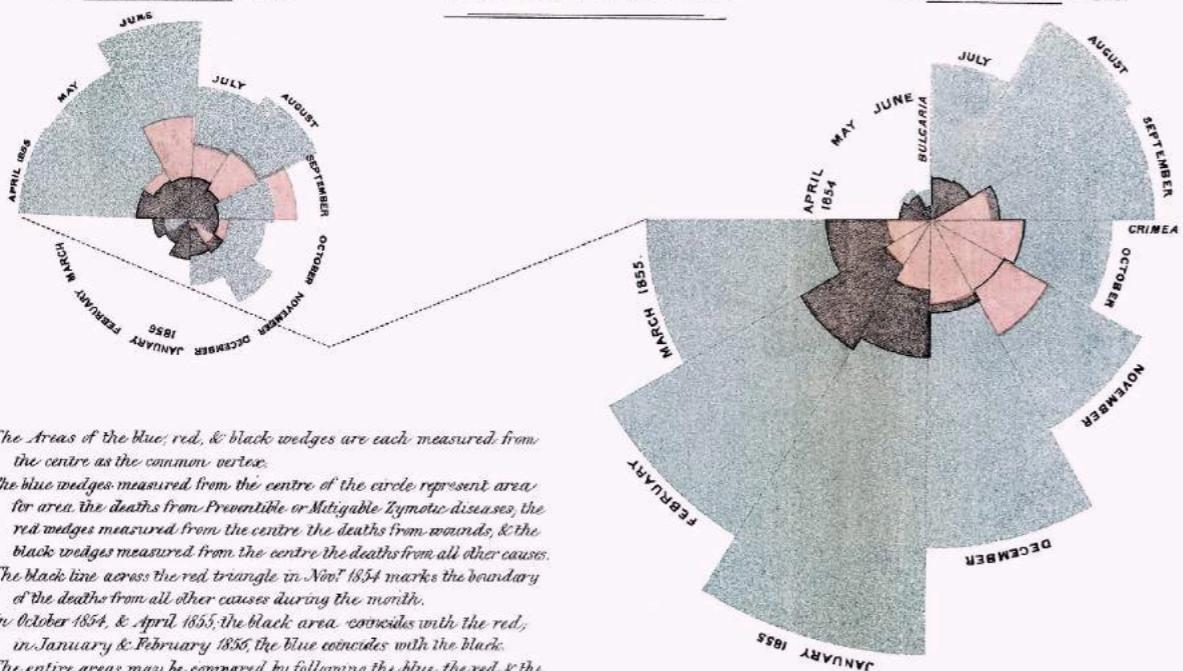
Les cosaques passent au galop
le Niémen gelé.



Autog. par Regnier, 8. Pas. S^e Marie St. Gain à Paris.

Imp. Lith. Regnier et Dourdet.

DIAGRAM OF THE CAUSES OF MORTALITY
IN THE ARMY IN THE EAST.



The areas of the blue, red, & black wedges are each measured from the centre as the common vertex.

The blue wedges measured from the centre of the circle represent area for area the deaths from Preventive or Mitigable Zymotic diseases, the red wedges measured from the centre the deaths from wounds, & the black wedges measured from the centre the deaths from all other causes.

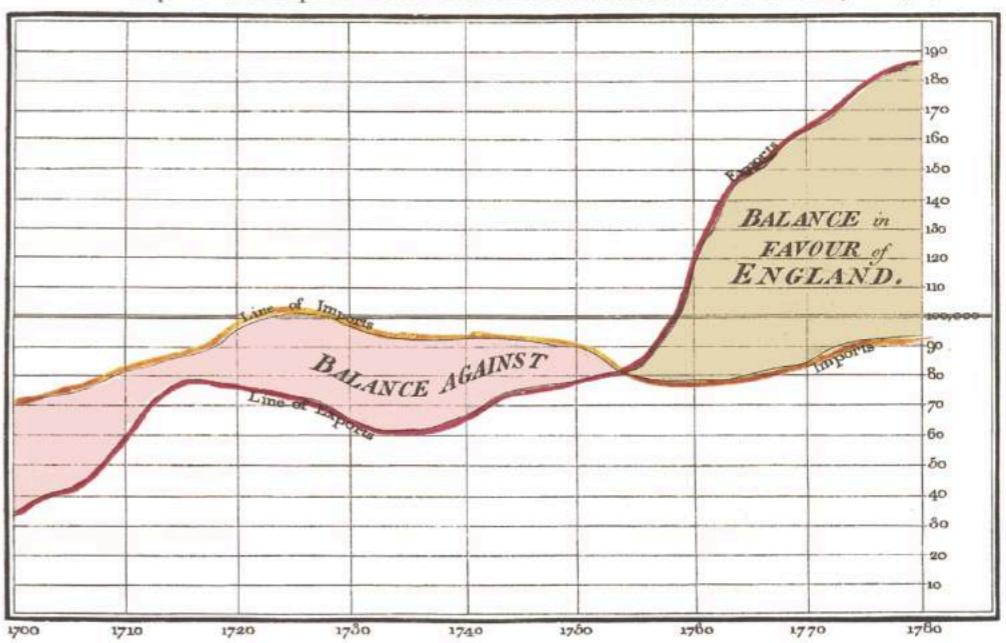
The black line across the red triangle in Nov^r 1854 marks the boundary of the deaths from all other causes during the month.

In October 1854, & April 1855, the black area coincides with the red;

in January & February 1856, the blue coincides with the black.

The entire areas may be compared by following the blue, the red & the black lines enclosing them.

Exports and Imports to and from DENMARK & NORWAY from 1700 to 1780.



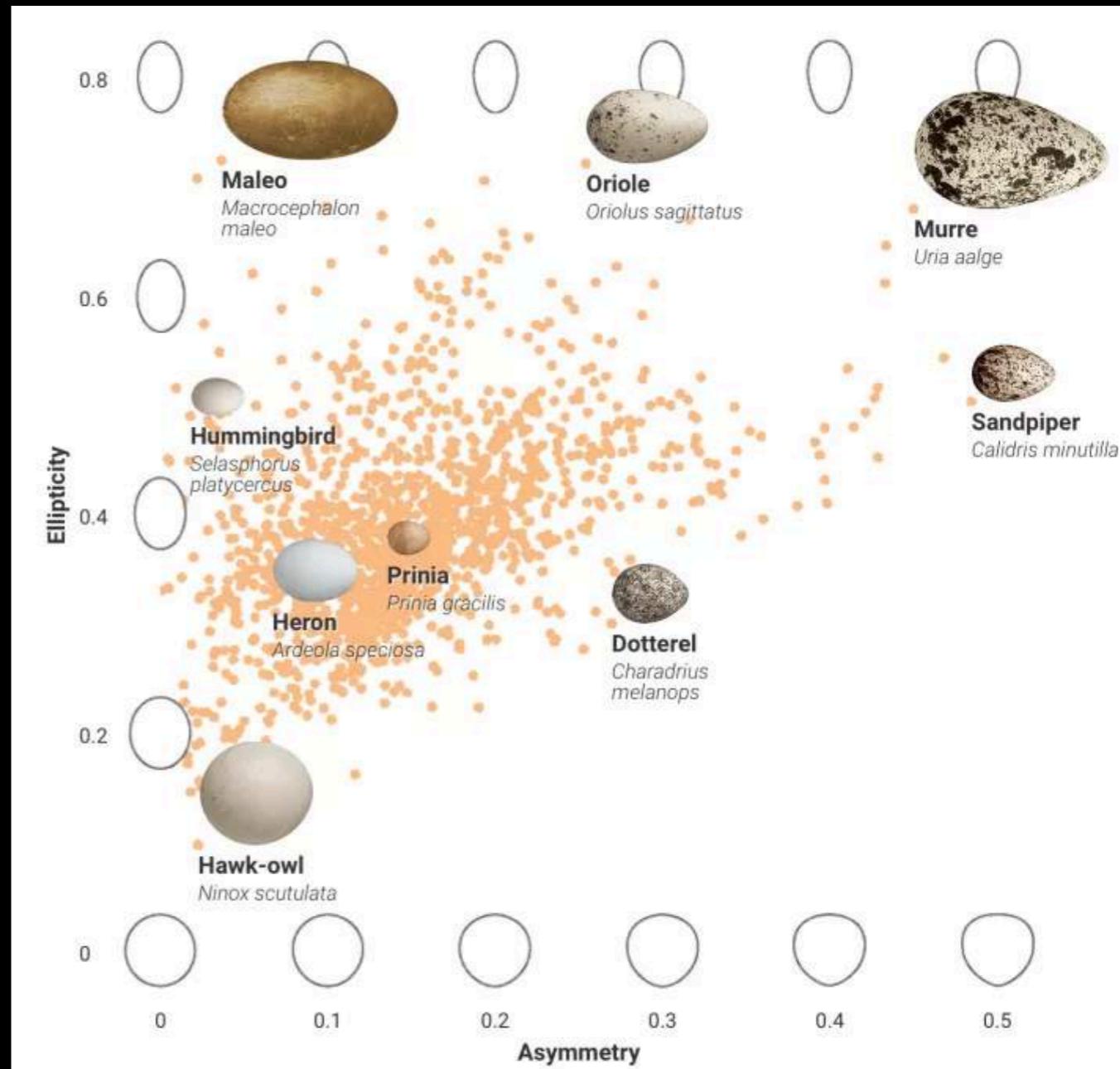
The Bottom line is divided into Years, the Right hand line into £10,000 each.
Published as the Act directs, 1st May 1868, by W^m Playfair
No. 100 Strand, London.

WHAT MAKES A VISUALIZATION MEMORABLE?

- *Great data visualization tells a great story*
 - Convince us of something, compel us to action, enlighten us with new information, or force us to question our own preconceptions
 - Great data visualization reaches reaches people in a way that words or numbers can't

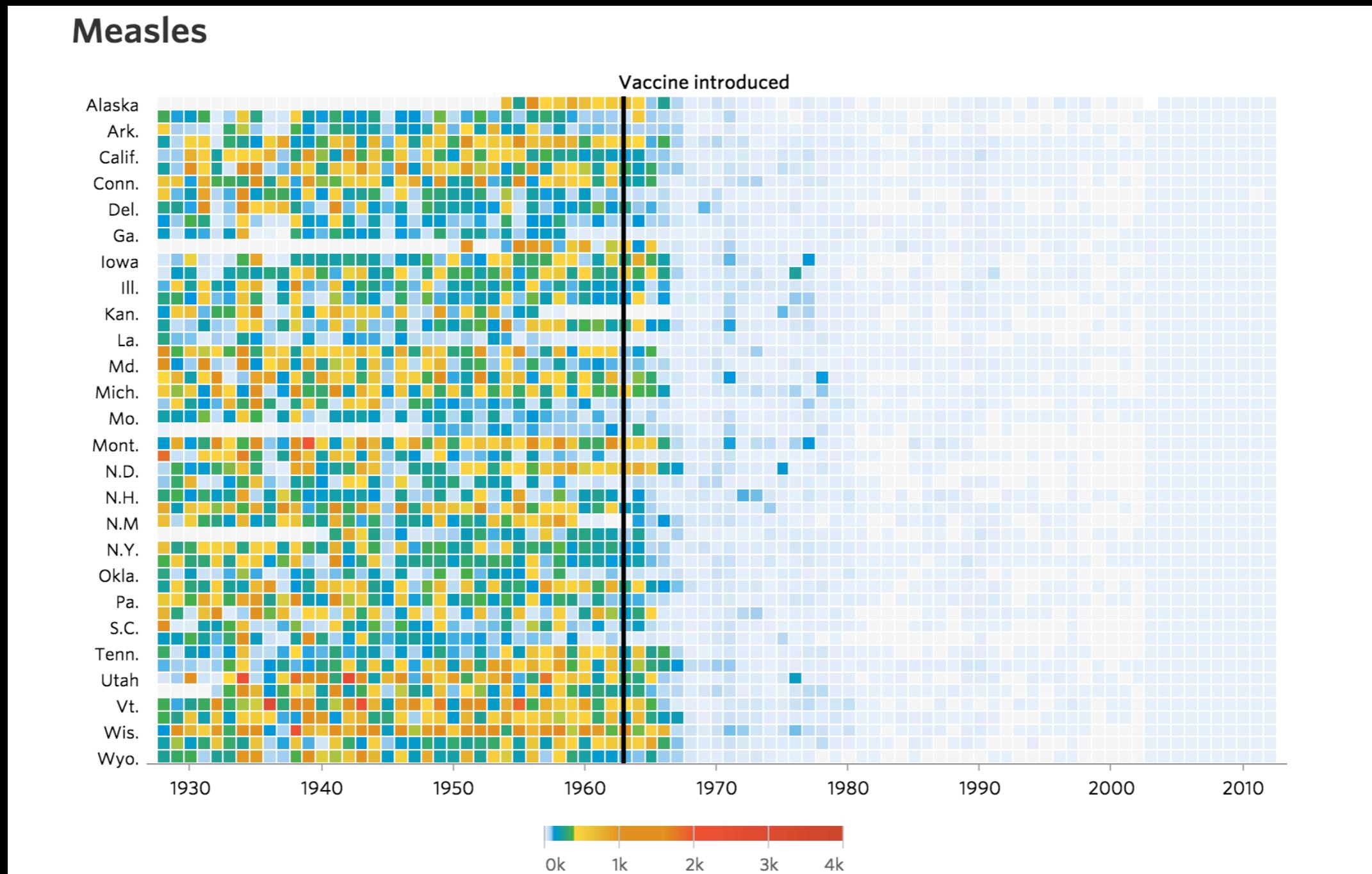
WHY DISPLAY DATA?

1. Initial examination and exploratory analysis
 - Check data (distribution, outliers, impossible values)
 - See data in context
 - Find patterns
 - Discover new questions



WHY DISPLAY DATA?

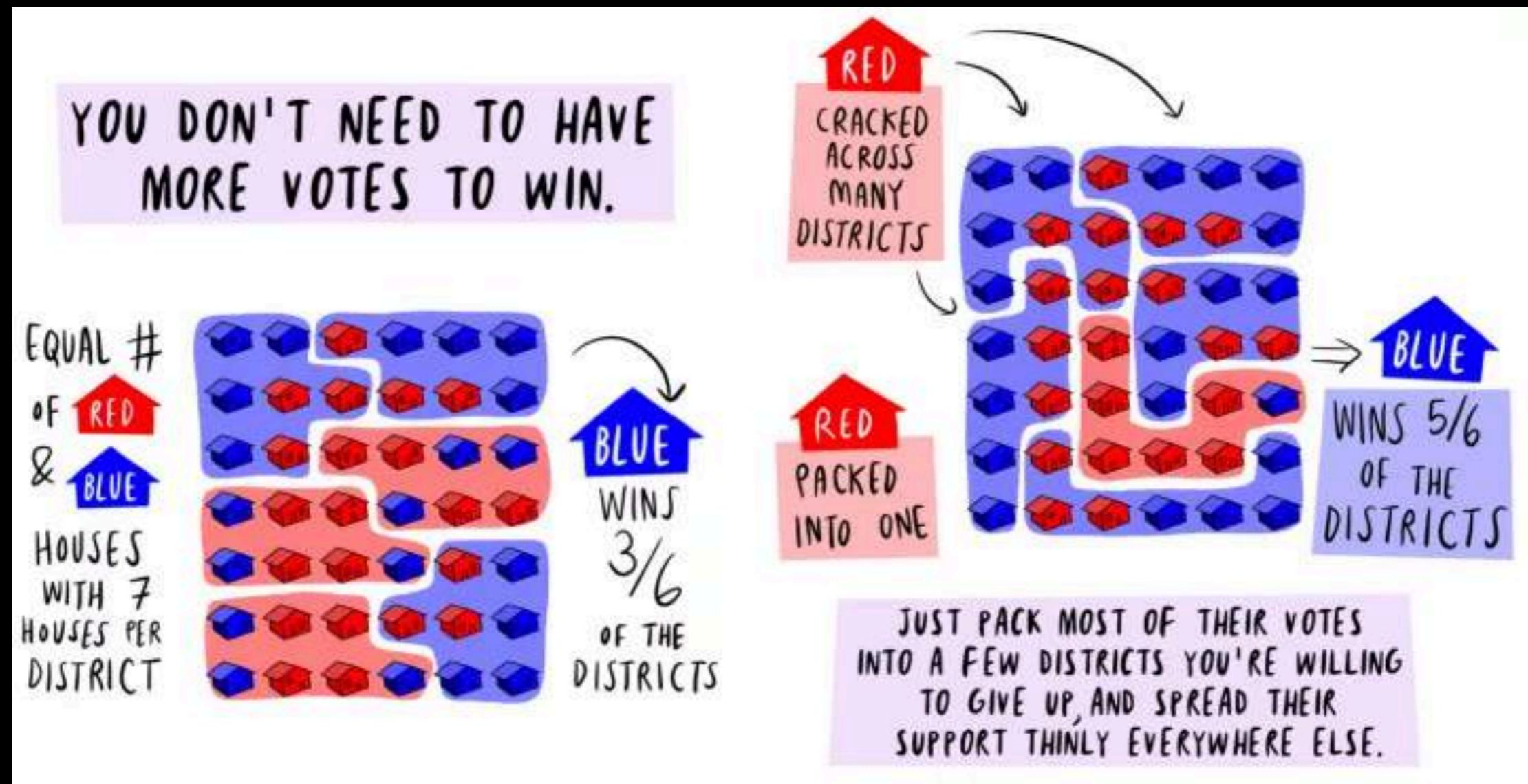
2. Answer questions & make decisions



WHY DISPLAY DATA?

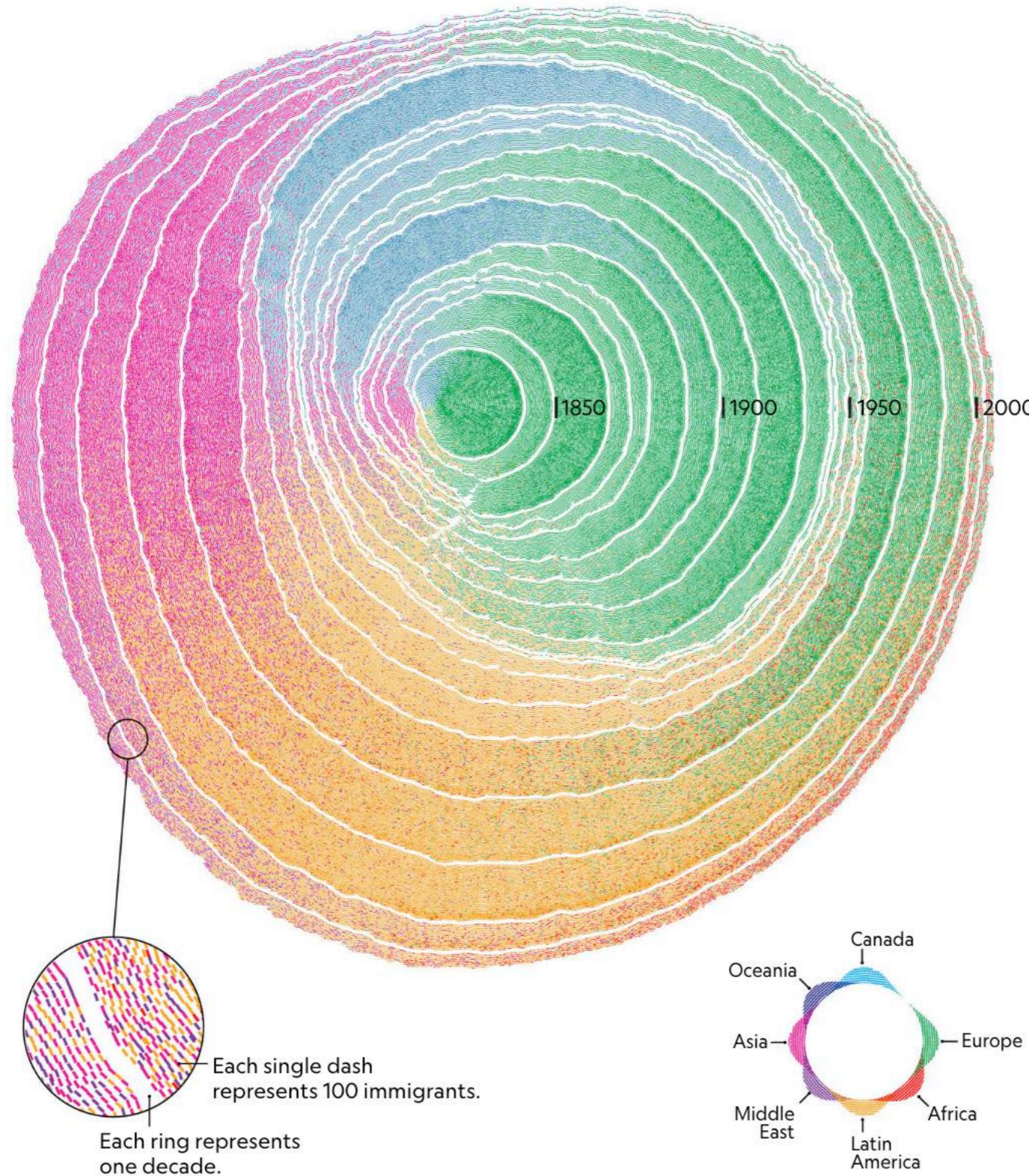
3. Communicate meaning to others

- Present argument, tell a story, inspire, archive



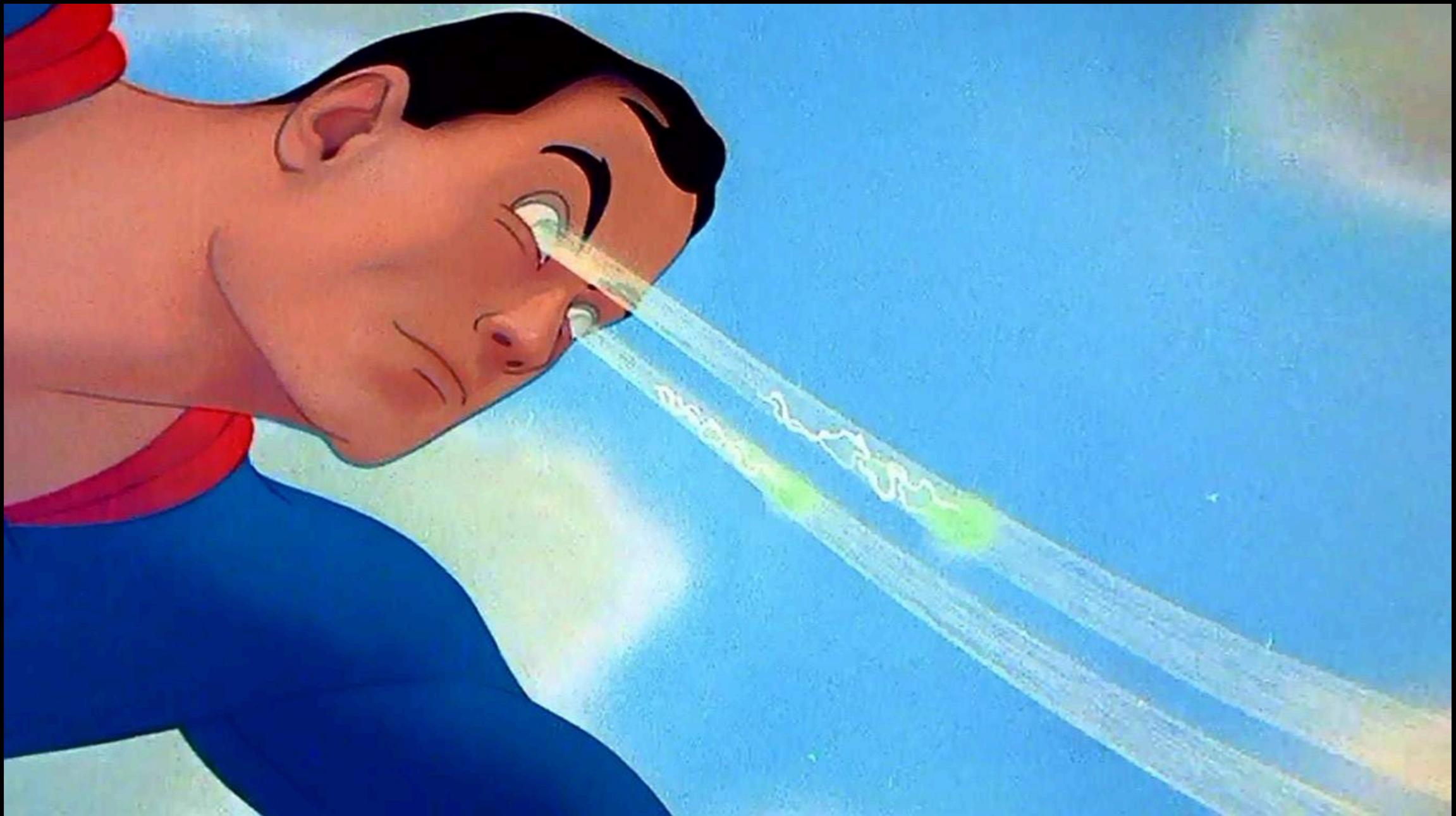
U.S. IMMIGRATION BY ORIGIN AT BIRTH, 1830-2015

<https://www.nationalgeographic.com/culture/2018/07/graphic-united-states-immigration-origins-rings-tree-culture/>



WHY WE VISUALIZE DATA

WHY WE VISUALIZE DATA



Human visual reasoning is like a superpower

WHY WE VISUALIZE DATA

- Information shaped as a graphic functions as a cognitive aid
- You can take something extremely complex, or something concealed in the data, and make it understandable and salient

CORE PRINCIPLES OF GOOD GRAPHICS

- They are **honest** representations of the data
- They show the right amount of data
- They attract readers' attention
- They don't frustrate readers

Some degree of
subjectivity &
judgement

Fig. 1.1 in Claus Wilke's *Data Visualization*

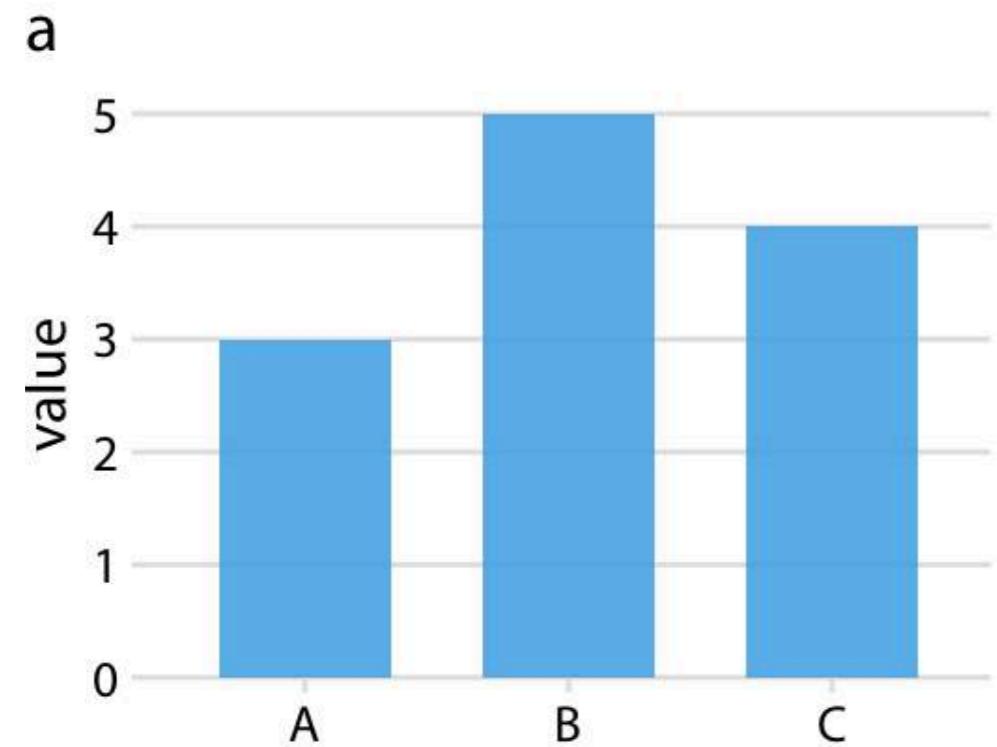


Fig. 1.1 in Claus Wilke's *Data Visualization*

Aesthetic problems or "bad taste"

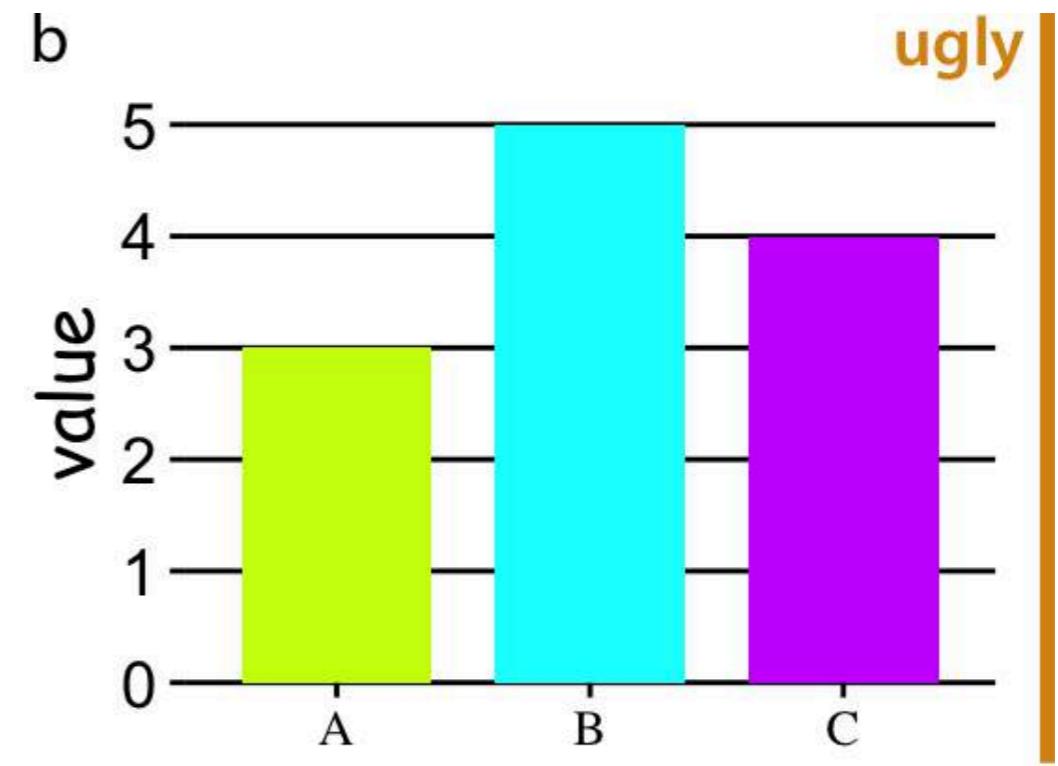
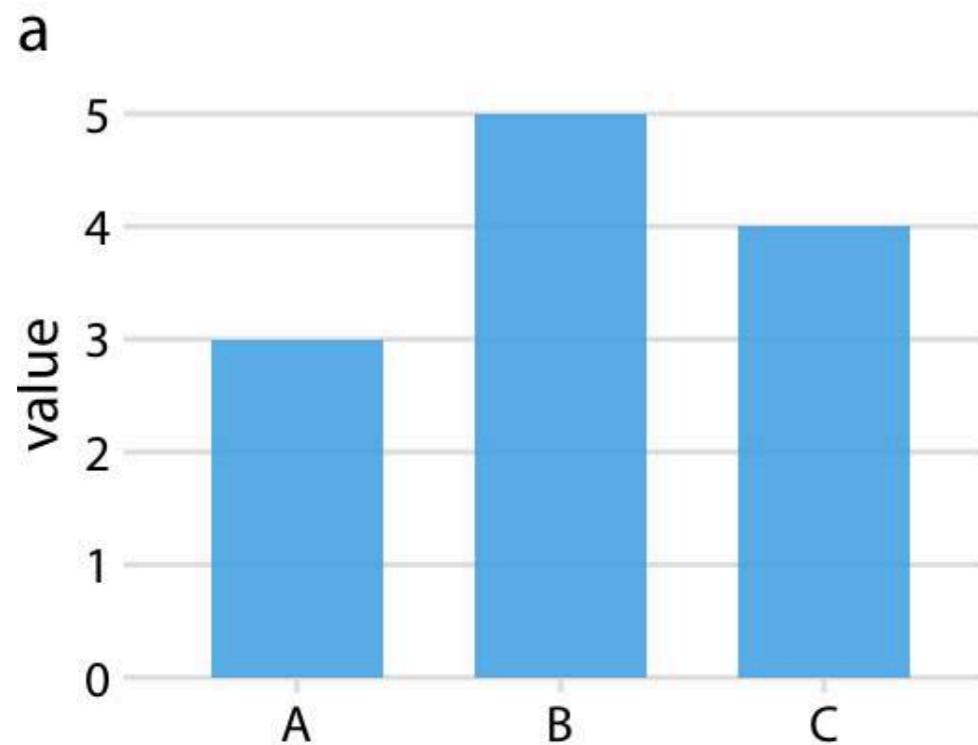
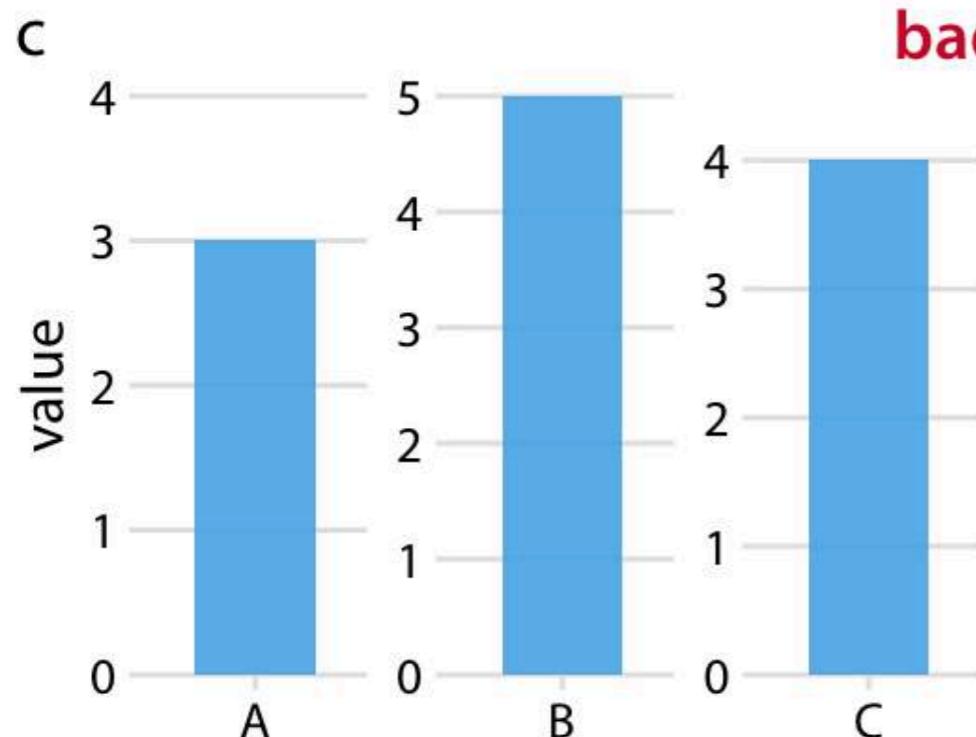
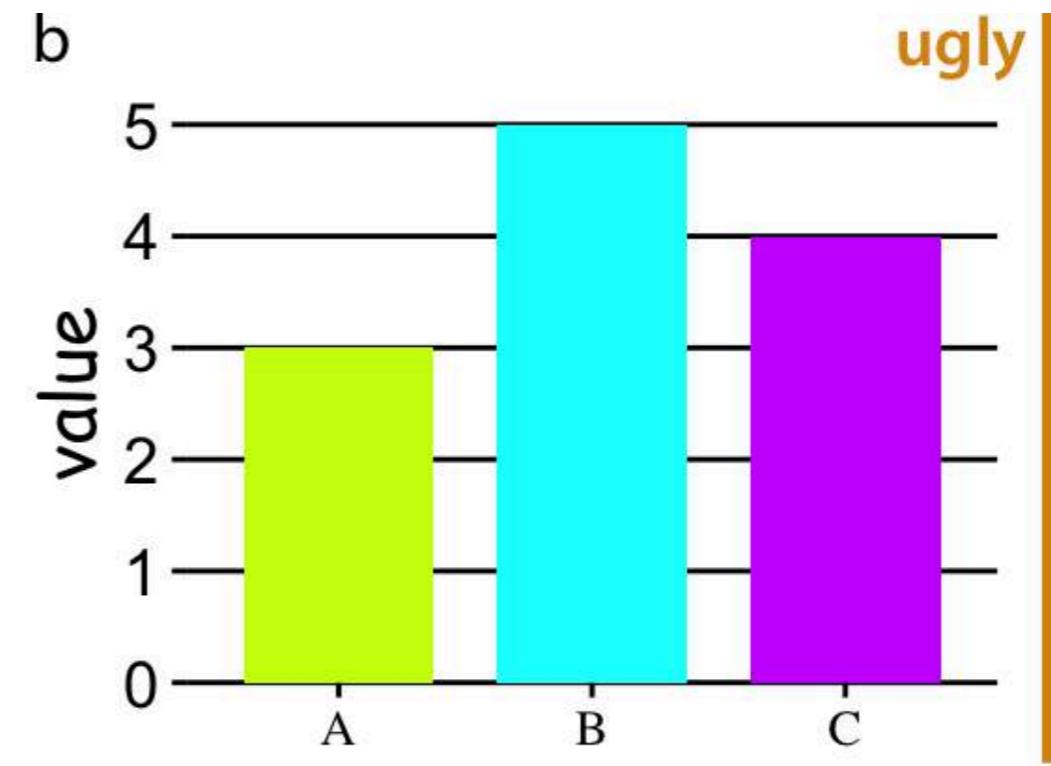
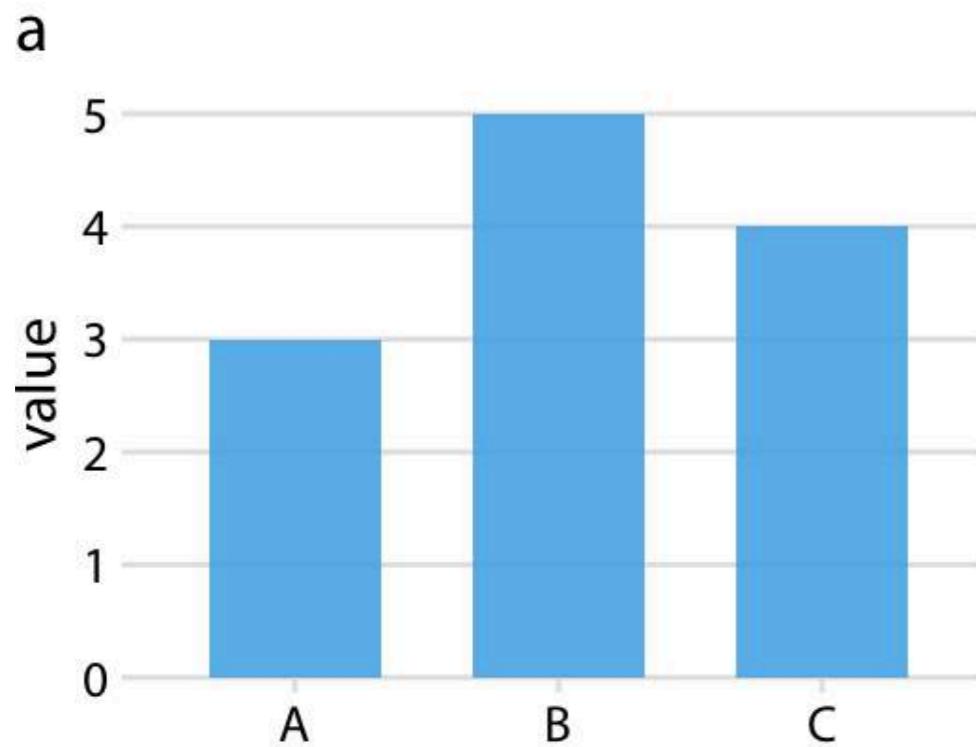


Fig. 1.1 in Claus Wilke's *Data Visualization*

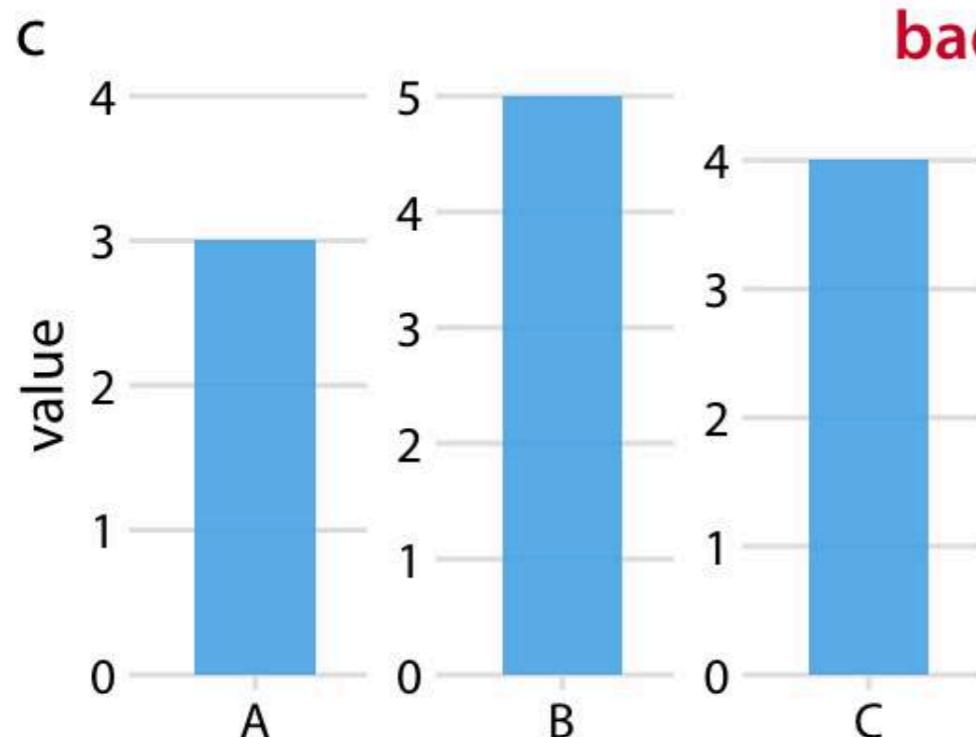
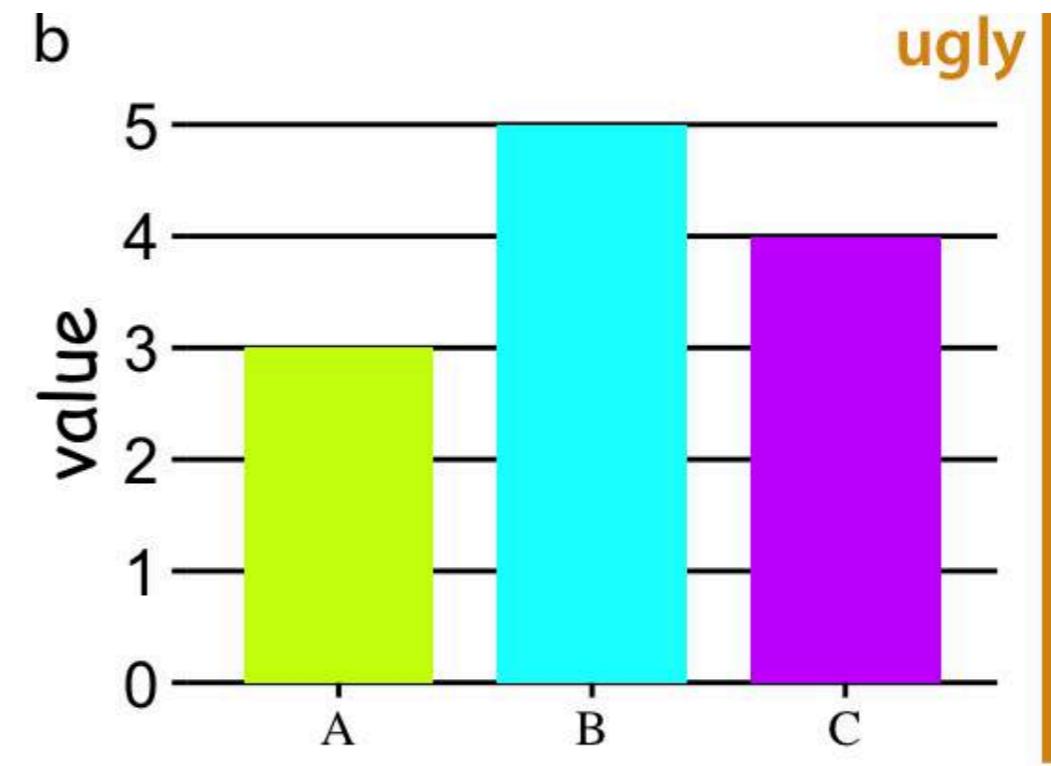
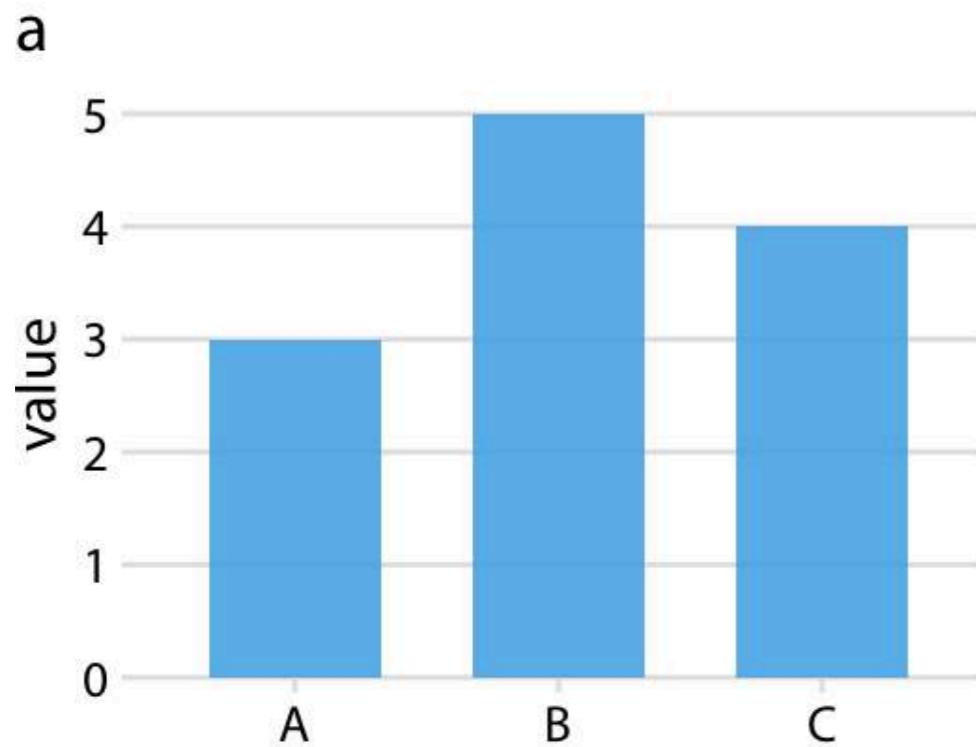
Aesthetic problems or "bad taste"



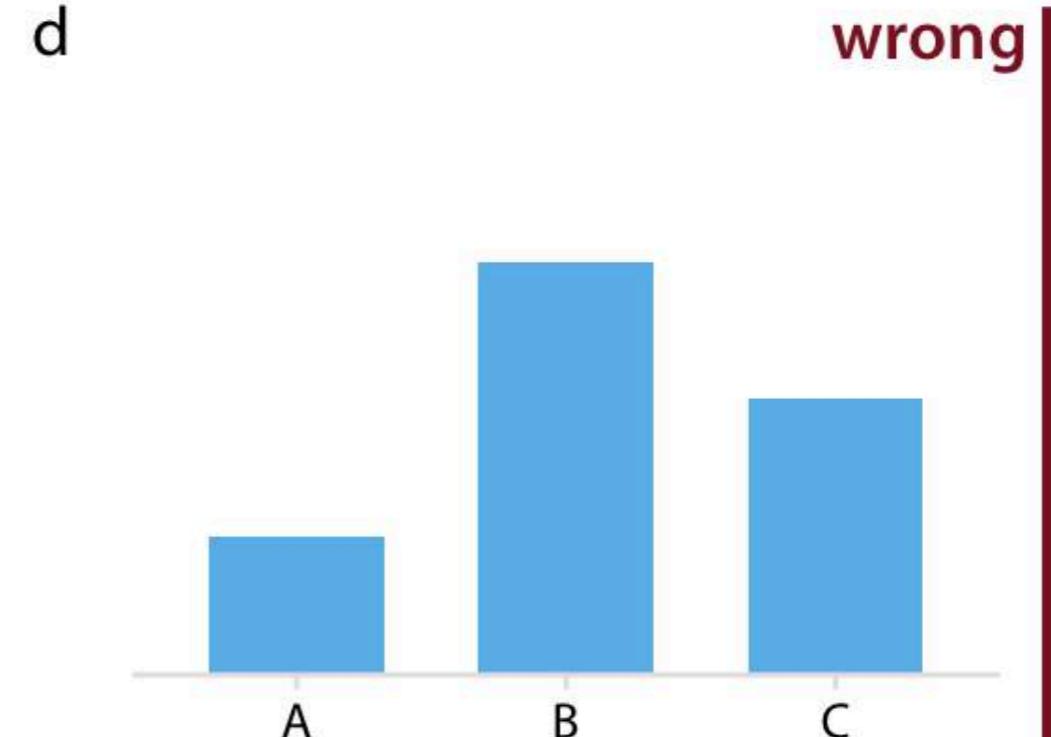
Perceptual problems

Fig. 1.1 in Claus Wilke's *Data Visualization*

Aesthetic problems or "bad taste"



Perceptual problems



Numerical problems or "bad data"

YOUR TURN

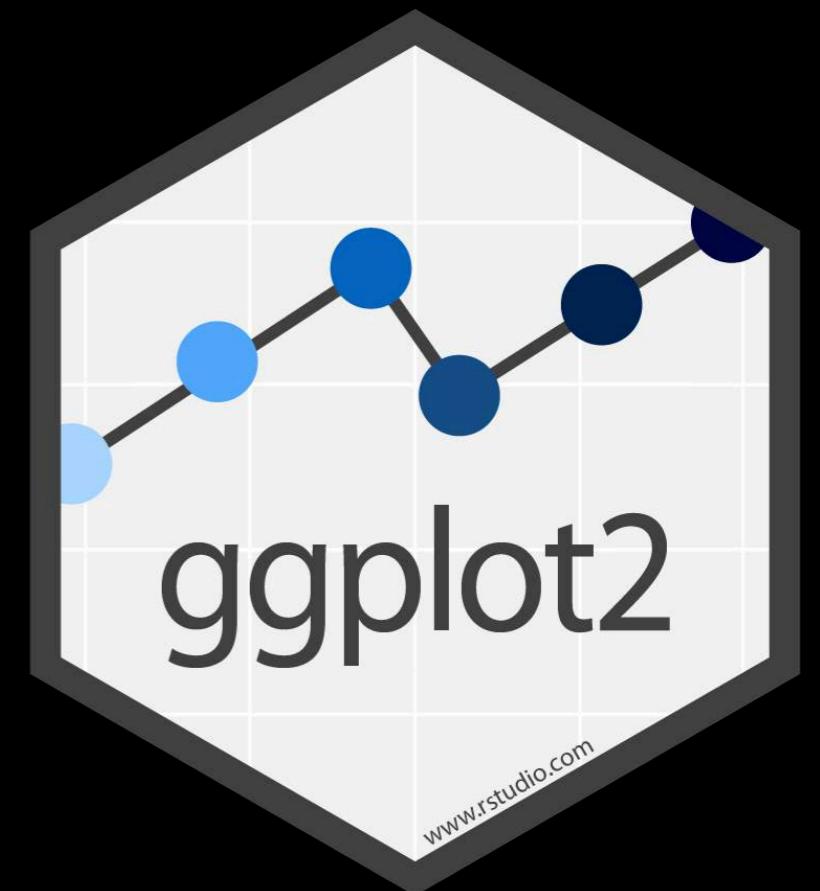


- Let's tidy up your "ANT 6973 Activities" project
- In the base directory, create a new folder and move all of last week's R scripts and R markdown files into the folder
- Create a new folder for this week's files.
- This is a *suggestion* for organization. You should fine-tune it to find what works best for you.

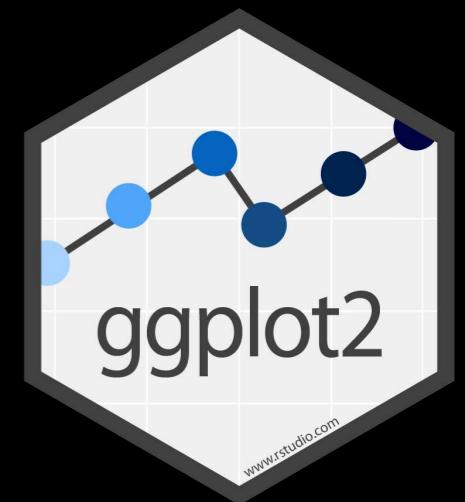


GGPLOT2

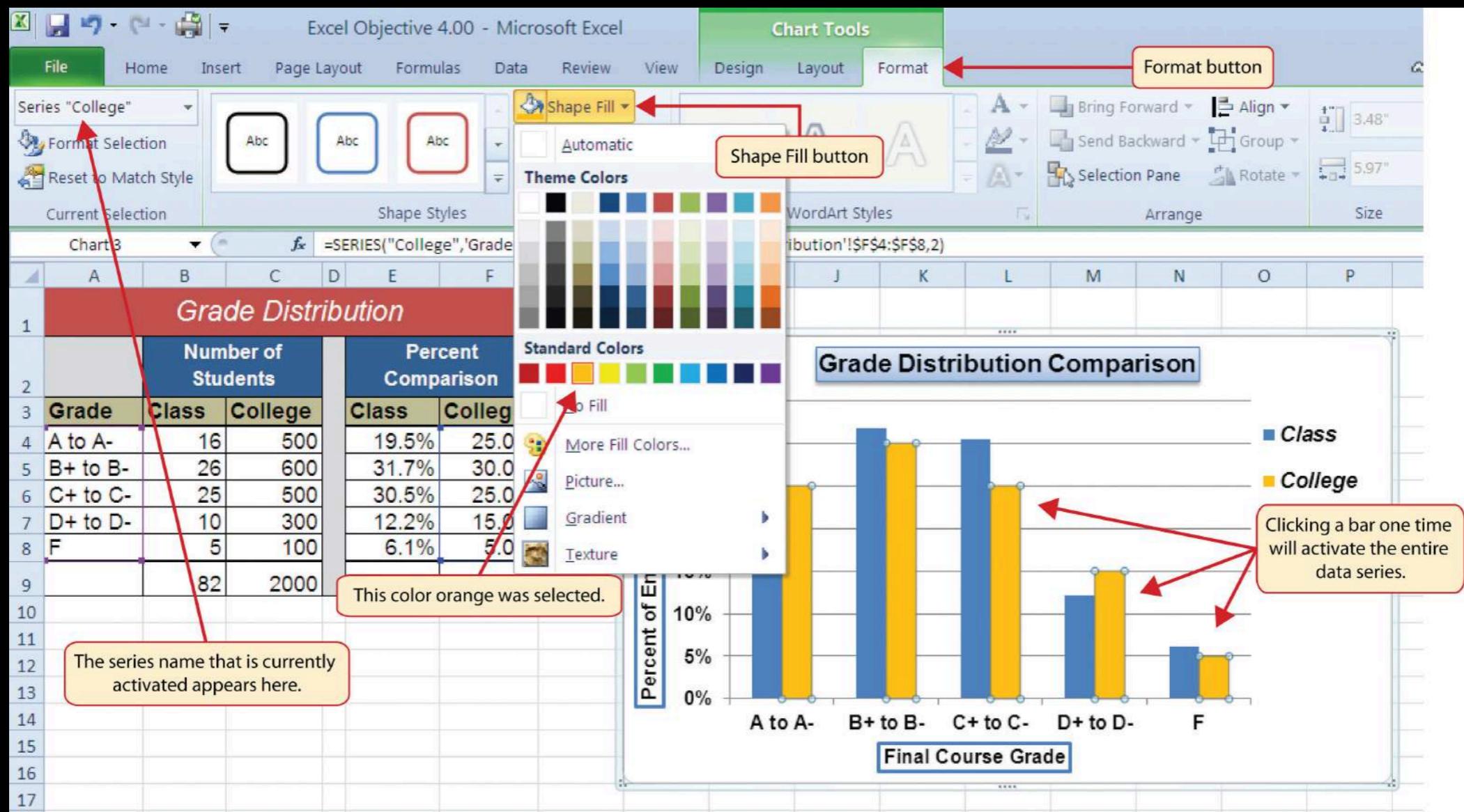
A GRAMMAR OF GRAPHICS



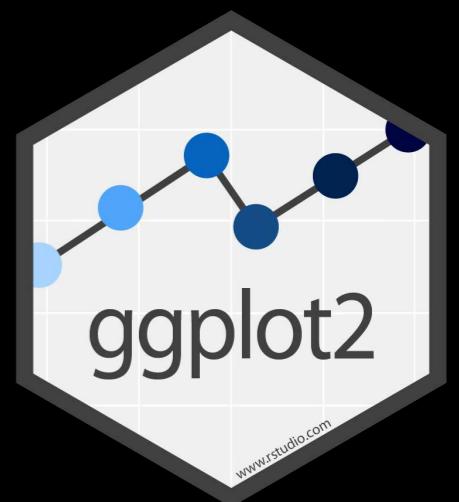
GRAMMAR OF GRAPHICS



- Traditional plotting: you *are* a painter
 - Manually place/modify graphical elements

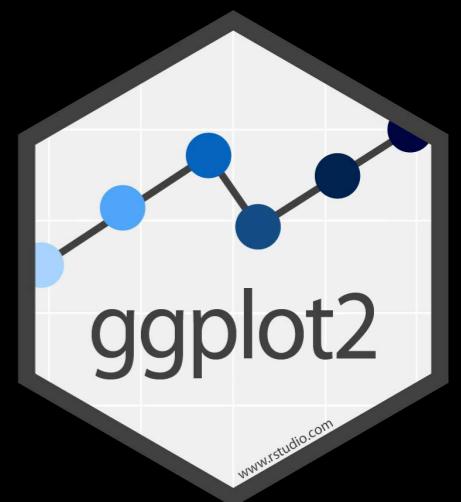


GRAMMAR OF GRAPHICS



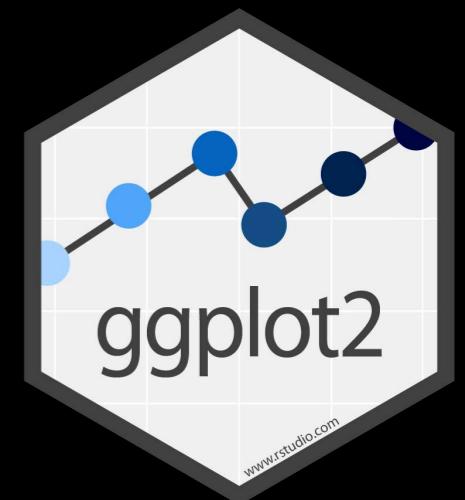
- Traditional plotting: you *are* a painter
 - Manually place/modify graphical elements

GRAMMAR OF GRAPHICS

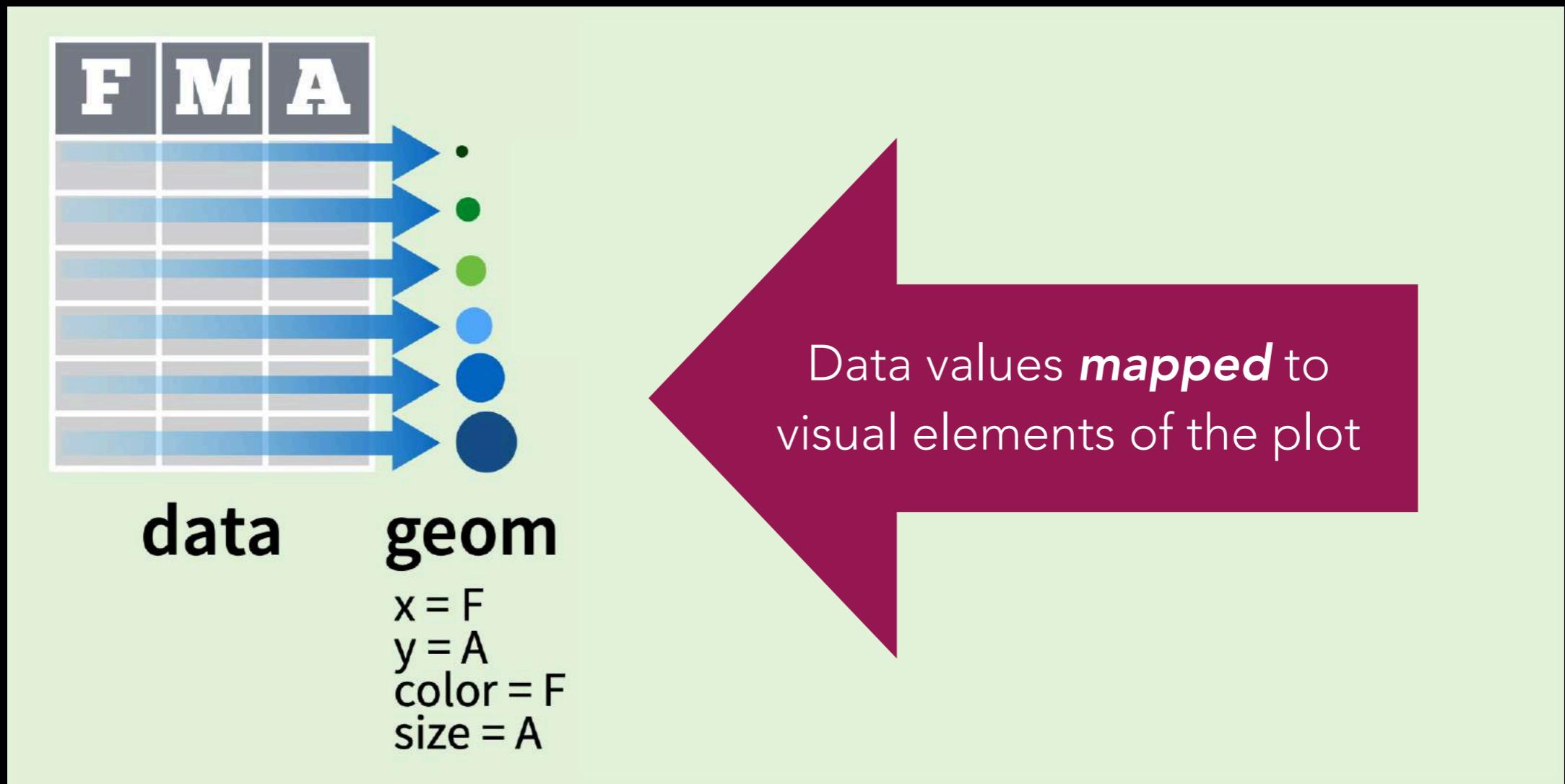


- ggplot2: you *employ* a painter
 - Describe conceptually how to visualize the data

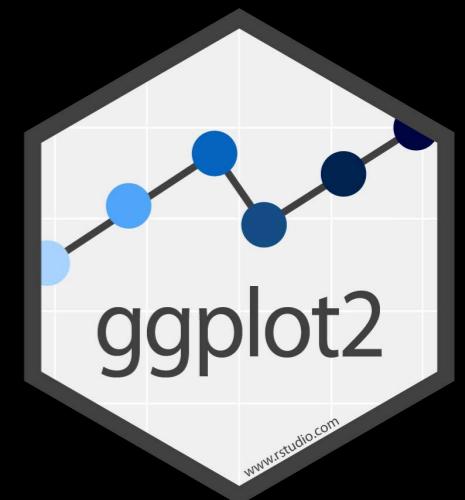
GRAMMAR OF GRAPHICS



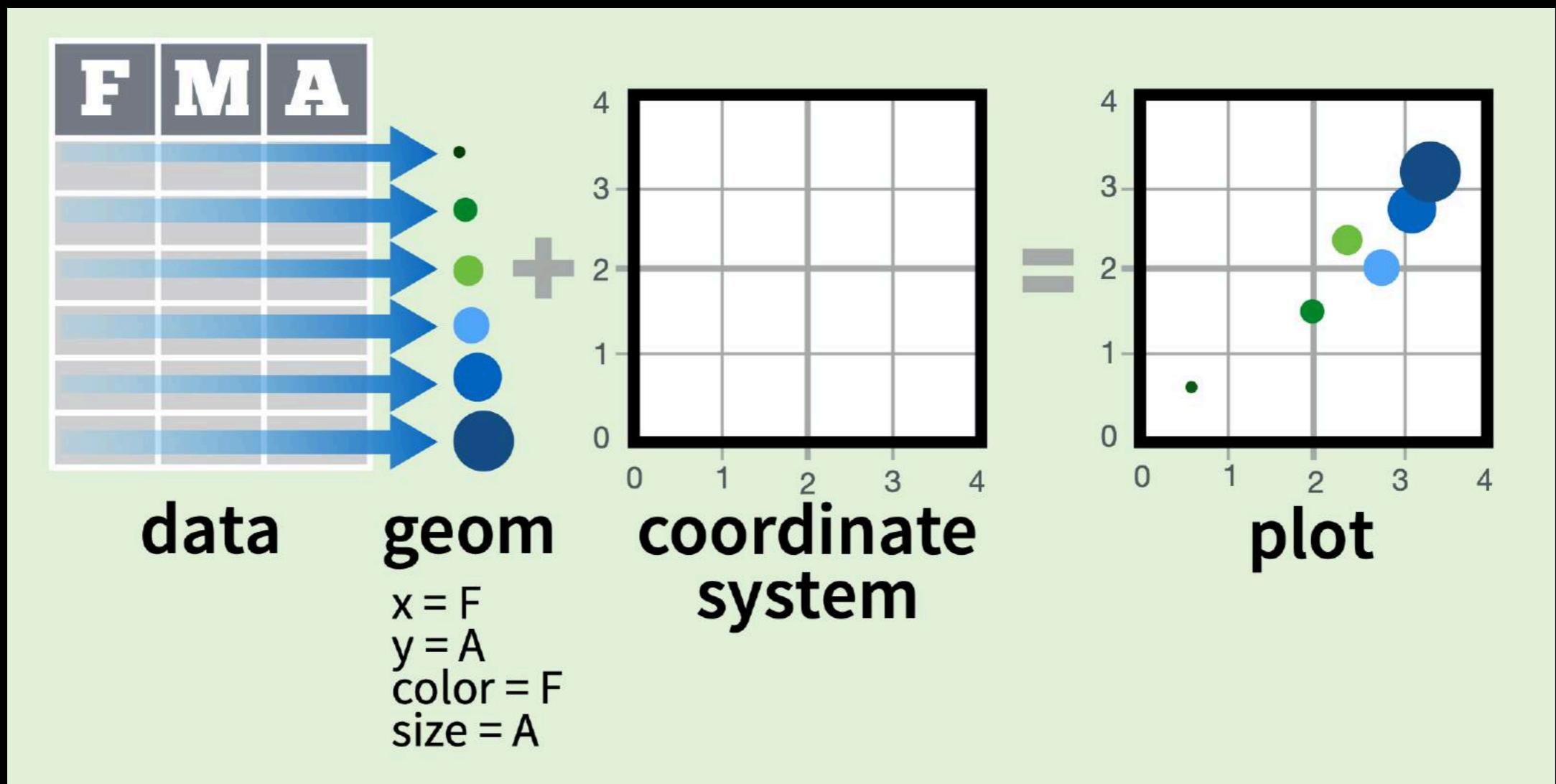
- ggplot2: you *employ* a painter
 - Describe conceptually how to visualize the data



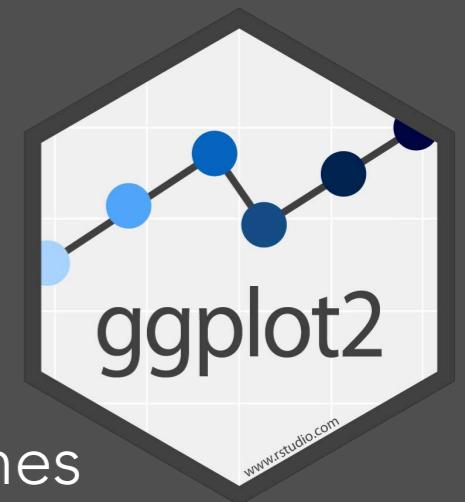
GRAMMAR OF GRAPHICS



- ggplot2: you *employ* a painter
 - Describe conceptually how to visualize the data



YOUR TURN



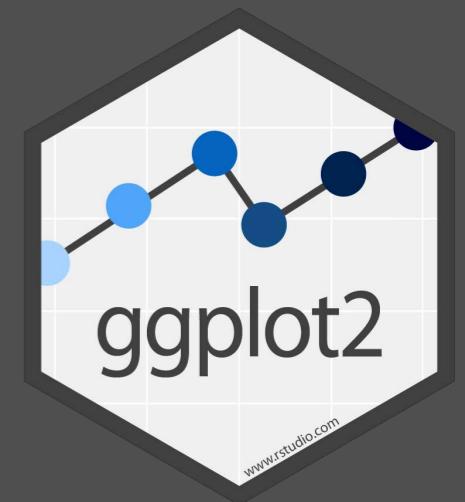
- Let's create a plot in ggplot2 using the `storms` dataset that comes with `tidyverse`

name	year	month	day	hour	lat	long	status	category	wind	pressure	ts_diameter	hu_diameter
Amy	1975	6	27	0	27.5	-79.0	tropical depression	-1	25	1013	NA	NA
Amy	1975	6	27	6	28.5	-79.0	tropical depression	-1	25	1013	NA	NA
Amy	1975	6	27	12	29.5	-79.0	tropical depression	-1	25	1013	NA	NA
Amy	1975	6	27	18	30.5	-79.0	tropical depression	-1	25	1013	NA	NA
Amy	1975	6	28	0	31.5	-78.8	tropical depression	-1	25	1012	NA	NA
Amy	1975	6	28	6	32.4	-78.7	tropical depression	-1	25	1012	NA	NA

- First, confer with your neighbors:
 - What relationship do you expect between wind speed and pressure?
 - Is storm status determined by wind speed or air pressure? (no Googling!)

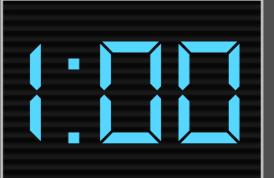


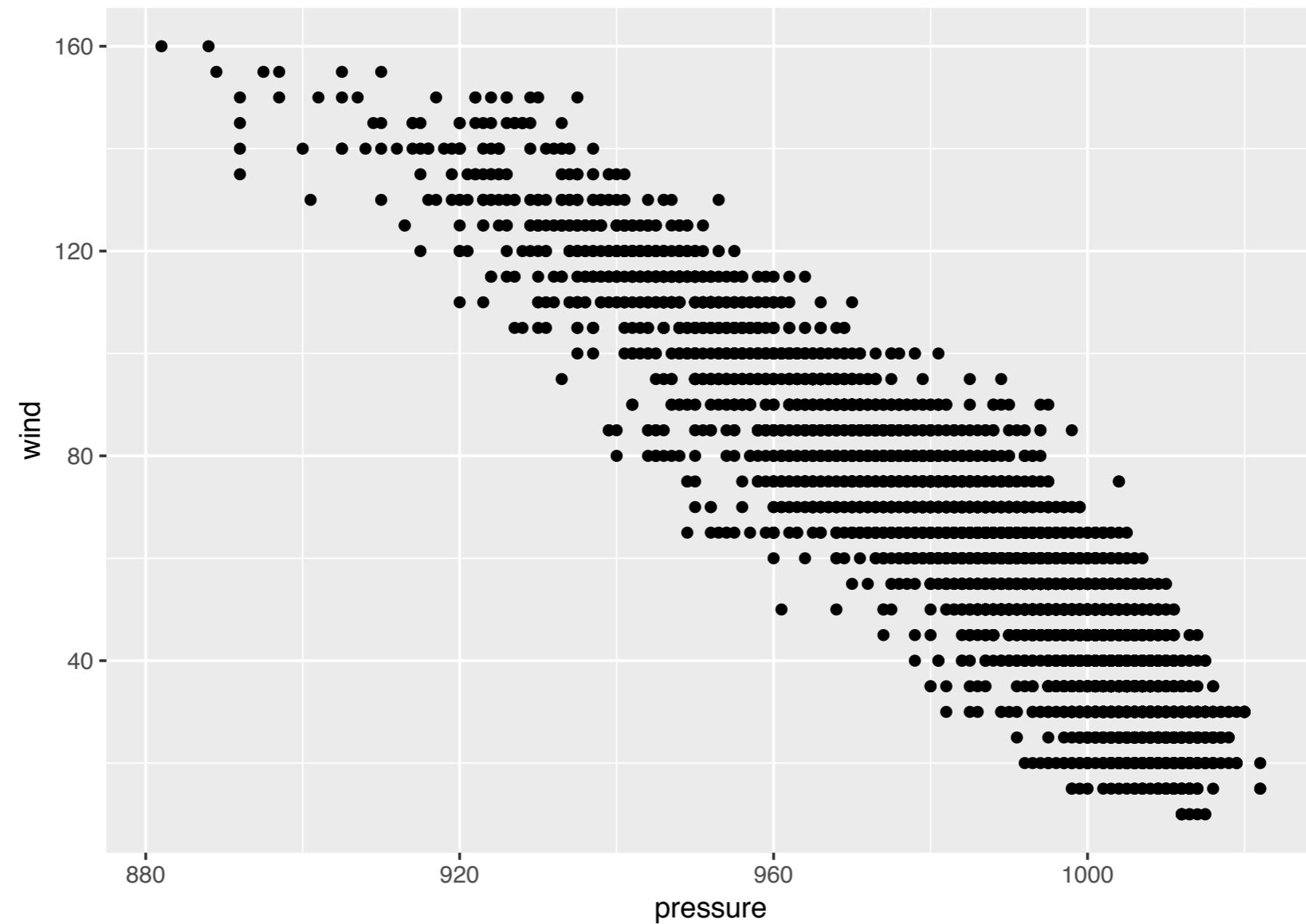
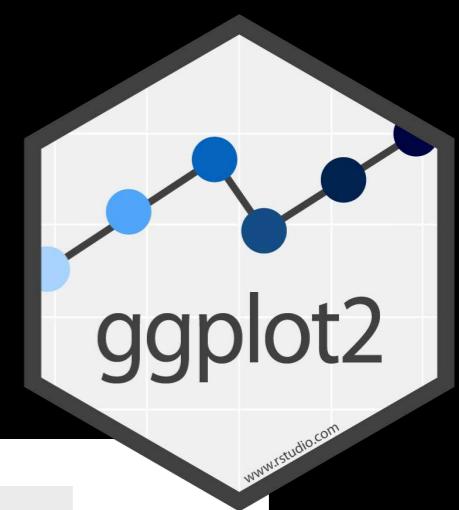
YOUR TURN



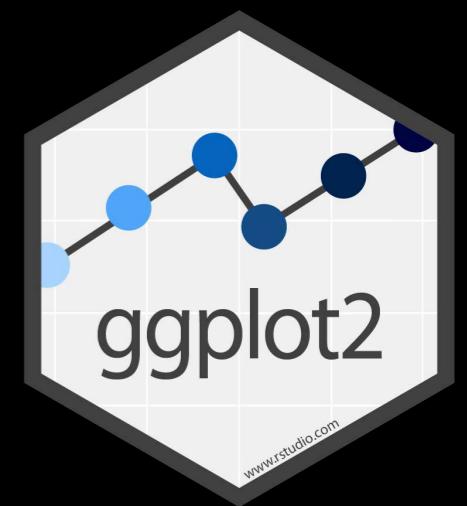
- Create a new R markdown and clear out the extra stuff
- Create a new R chunk and load the `tidyverse` package.
- Run this code to make a graph. Pay strict attention to spelling, capitalization, and parentheses!

```
ggplot(data = storms) +  
  geom_point(mapping = aes(x = pressure, y = wind))
```





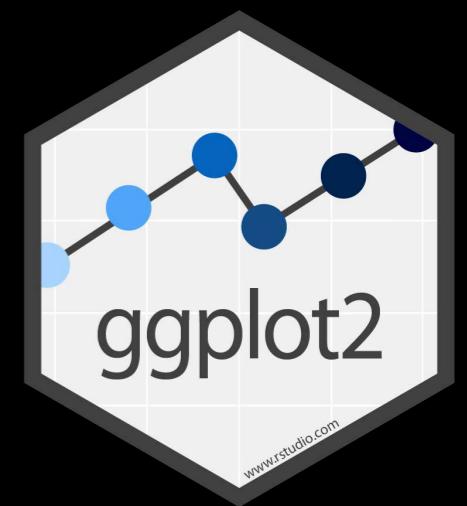
```
ggplot(data = storms) +  
  geom_point(mapping = aes(x = pressure, y = wind))
```



1. "Initialize" a plot with `ggplot()`
2. Add layers with `geom_` functions

Important tip: always put the `+` at the end of a line, never at the start

```
ggplot(data = storms) +  
  geom_point(mapping = aes(x = pressure, y = wind))
```



initialize

data

+ before new line

```
ggplot(data = storms) +  
  geom_point(mapping = aes(x = pressure, y = wind))
```

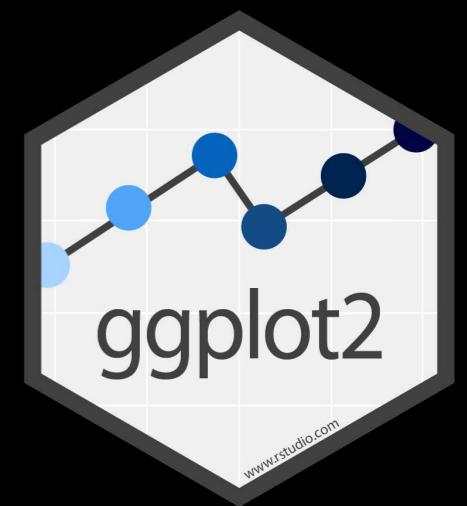
type of layer

aes()

x variable

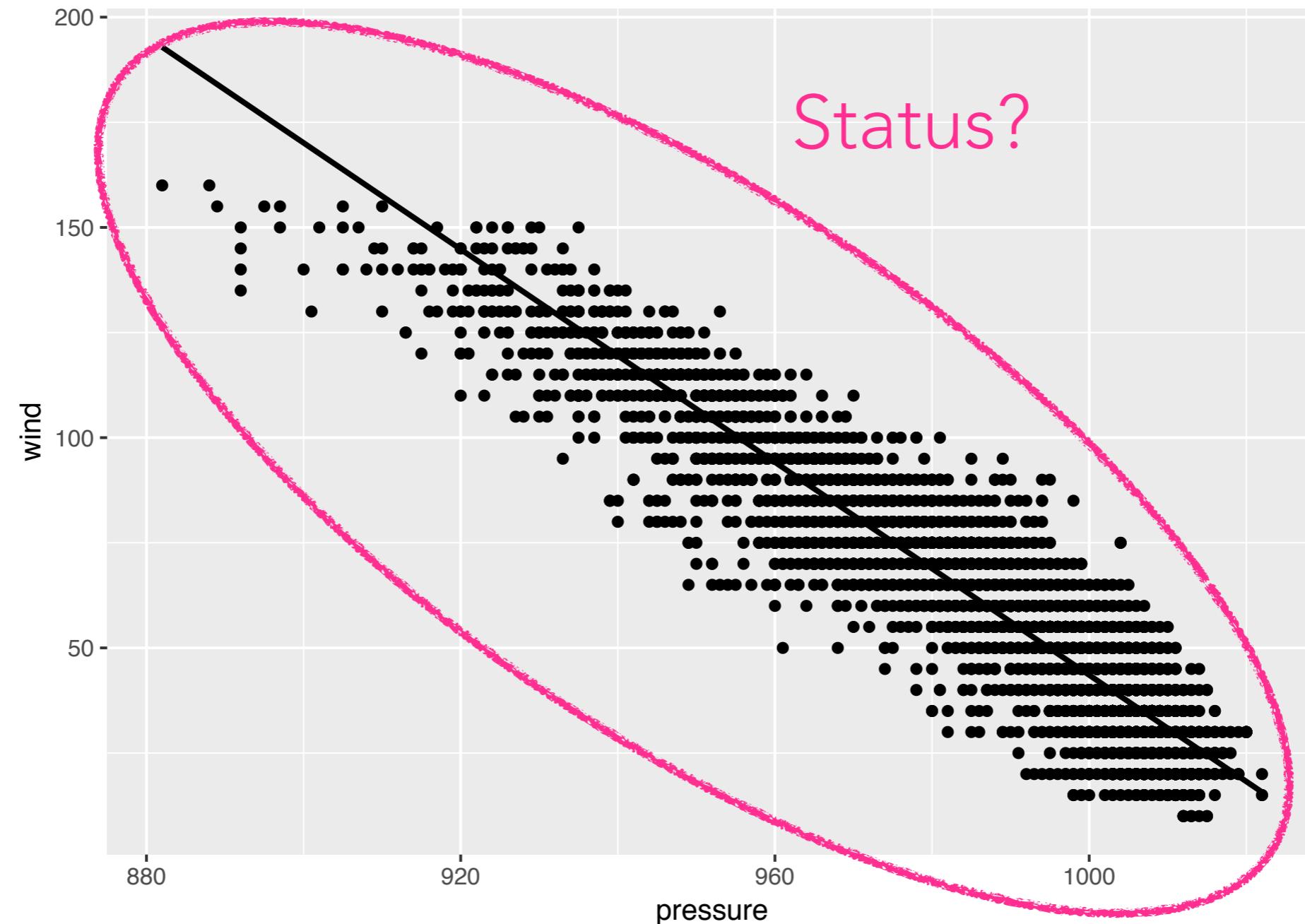
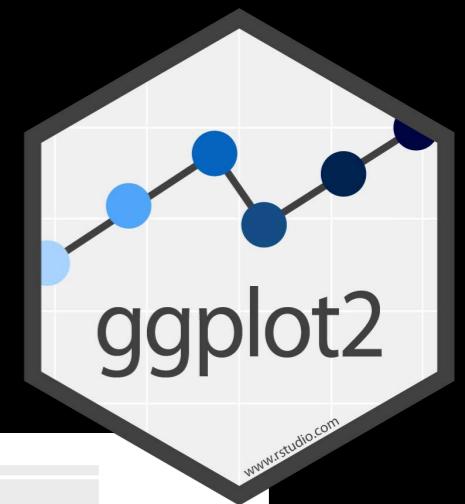
y variable

A TEMPLATE



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

```
ggplot(data = storms) +  
  geom_point(mapping = aes(x = pressure, y = wind))
```



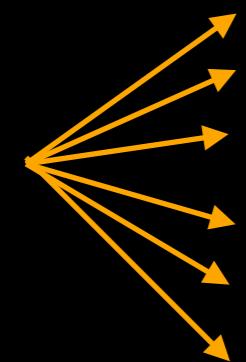
```
ggplot(data = storms) +  
  geom_point(mapping = aes(x = pressure, y = wind))
```

VISUAL ENCODING

VISUAL ENCODING

- Every visualization can be described as a set of mappings:
 - From data *items* to visual *marks*.
 - From data *variables* to visual *channels*.

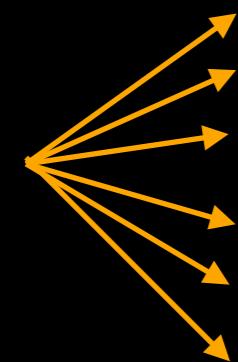
Items
(cases)



name	year	month	day	hour	lat	long	status	category	wind	pressure	ts_diameter	hu_diameter
Amy	1975	6	27	0	27.5	-79.0	tropical depression	-1	25	1013	NA	NA
Amy	1975	6	27	6	28.5	-79.0	tropical depression	-1	25	1013	NA	NA
Amy	1975	6	27	12	29.5	-79.0	tropical depression	-1	25	1013	NA	NA
Amy	1975	6	27	18	30.5	-79.0	tropical depression	-1	25	1013	NA	NA
Amy	1975	6	28	0	31.5	-78.8	tropical depression	-1	25	1012	NA	NA
Amy	1975	6	28	6	32.4	-78.7	tropical depression	-1	25	1012	NA	NA

Variables (attributes)

Items
(cases)



name	year	month	day	hour	lat	long	status	category	wind	pressure	ts_diameter	hu_diameter
Amy	1975	6	27	0	27.5	-79.0	tropical depression	-1	25	1013	NA	NA
Amy	1975	6	27	6	28.5	-79.0	tropical depression	-1	25	1013	NA	NA
Amy	1975	6	27	12	29.5	-79.0	tropical depression	-1	25	1013	NA	NA
Amy	1975	6	27	18	30.5	-79.0	tropical depression	-1	25	1013	NA	NA
Amy	1975	6	28	0	31.5	-78.8	tropical depression	-1	25	1012	NA	NA
Amy	1975	6	28	6	32.4	-78.7	tropical depression	-1	25	1012	NA	NA

- **Marks** are the basic visual objects/units that represent data objects visually.
- **Visual channels** are visual variables we can use to represent characteristics of these objects.

Variables (attributes)

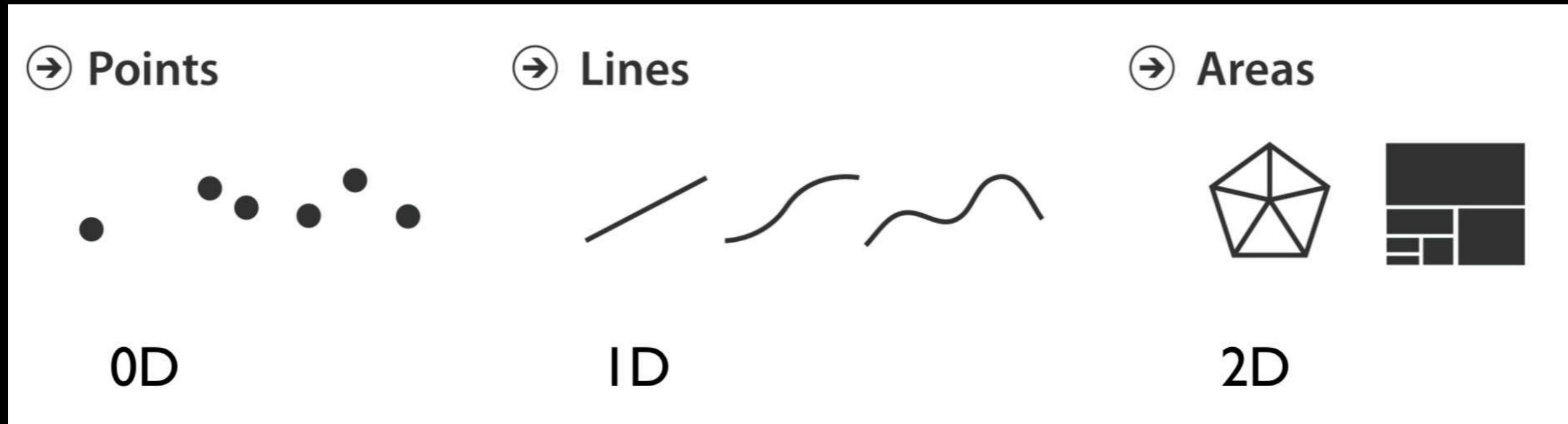
Items
(cases)

MAPPING

name	year	month	day	hour	lat	long	category	wind	pressure	ts_diameter	hu_diameter
Amy	1975	6	27	0	27.5	-79.0	tropical depression	-1	25	1013	NA
Amy	1975	6	27	6	28.5	-79.0	tropical depression	-1	25	1013	NA
Amy	1975	6	27	12	29.0	-79.0	tropical depression	-1	25	1013	NA
Amy	1975	6	27	18	30.0	-79.0	tropical depression	-1	25	1013	NA
Amy	1975	6	28	0	31.5	-78.8	tropical depression	-1	25	1012	NA
Amy	1975	6	28	6	32.4	-78.7	tropical depression	-1	25	1012	NA

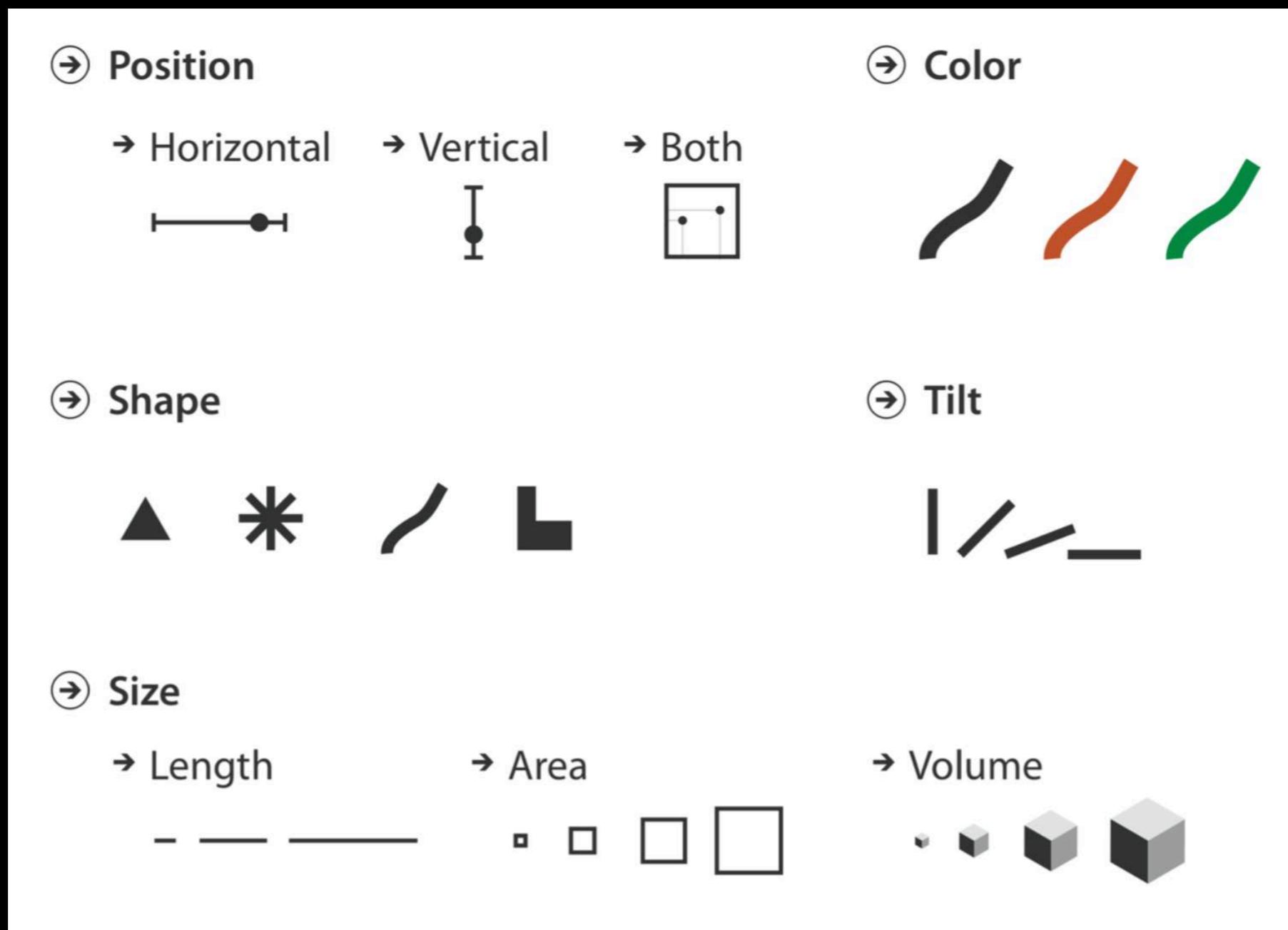
- **Marks** are the basic visual objects/units that represent data objects visually.
- **Visual channels** are visual variables we can use to represent characteristics of these objects.

MARKS: BASIC GEOMETRIC ELEMENTS

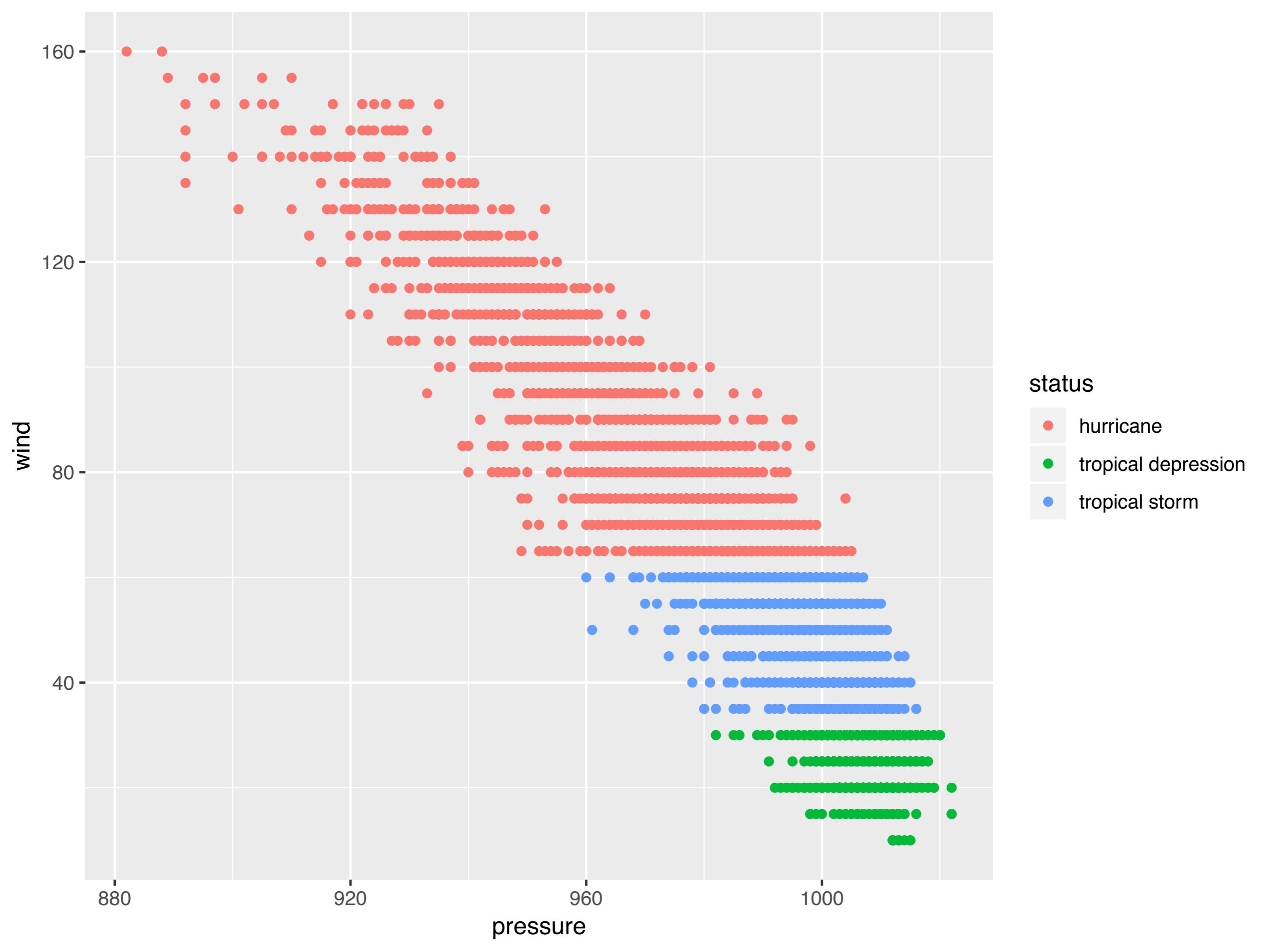


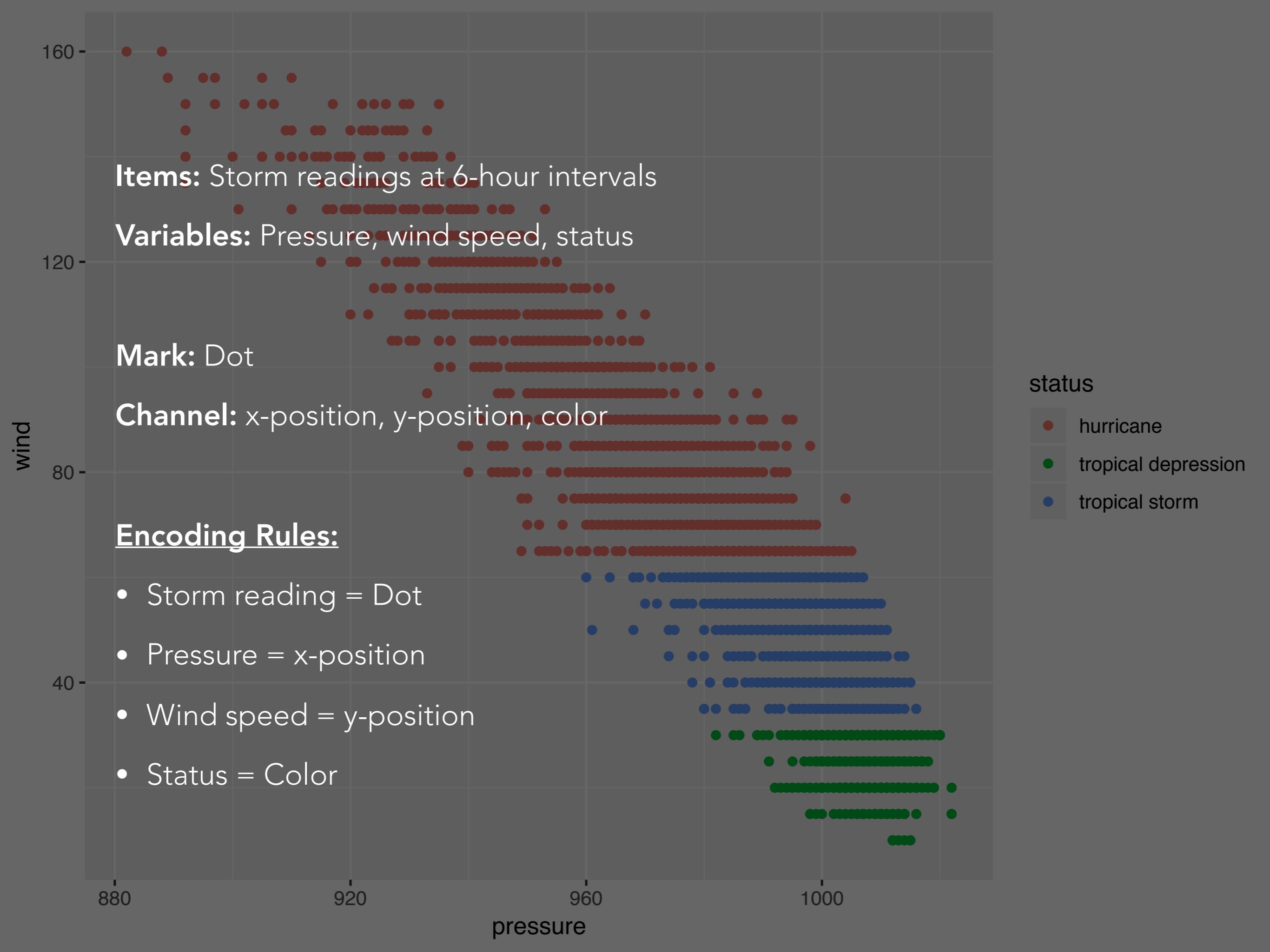
- These are “geoms” in ggplot2’s jargon

CHANNELS: VISUAL VARIABLES

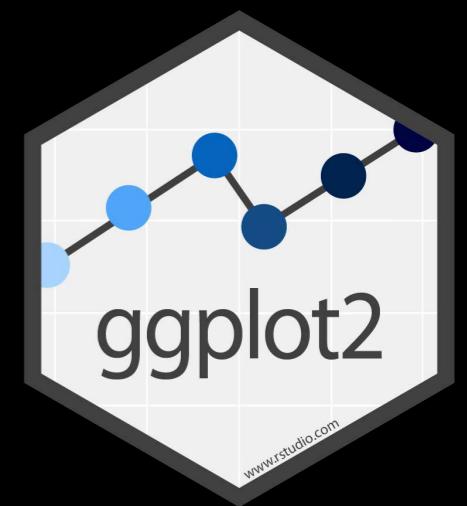


- These are “aesthetics” in ggplot2’s jargon

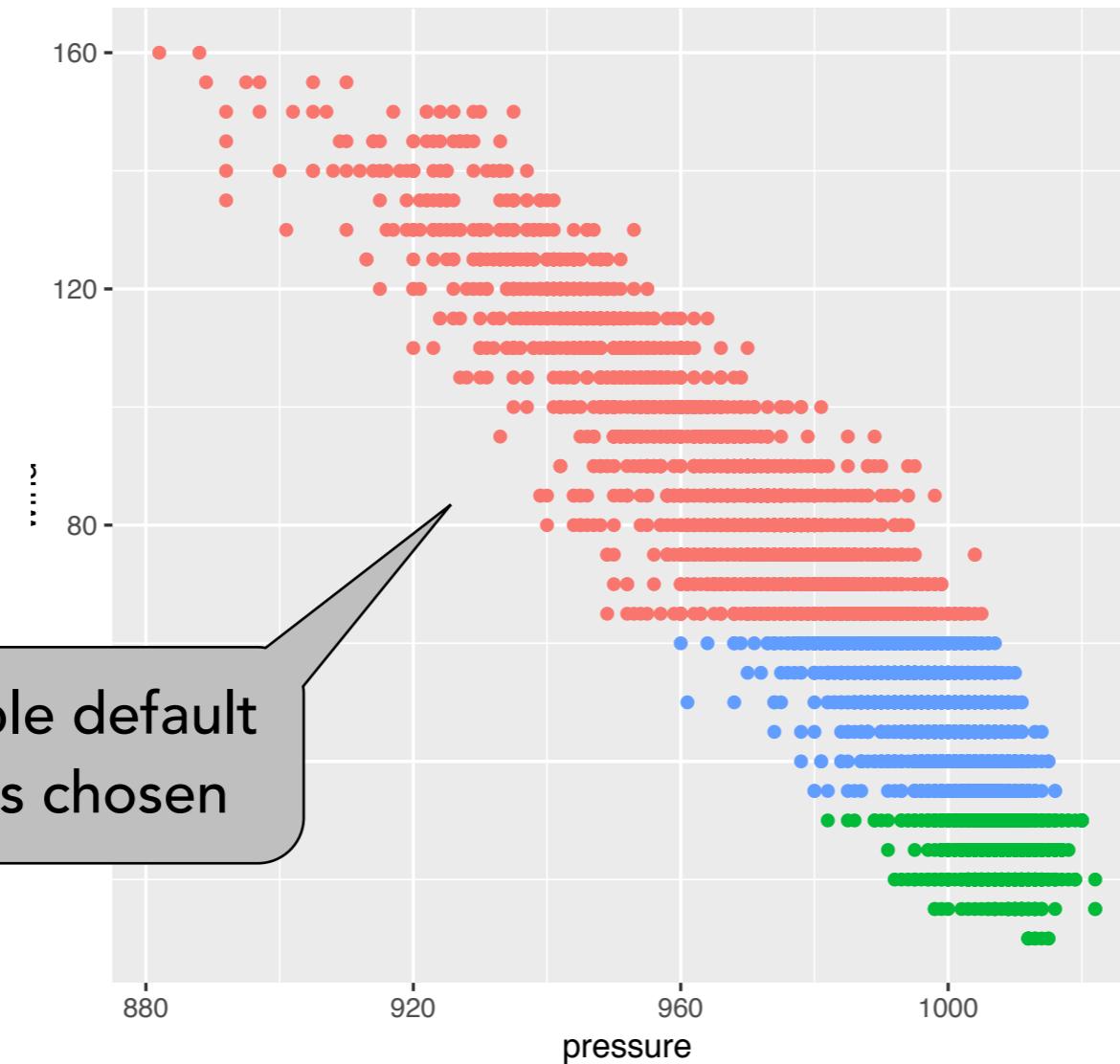




AESTHETICS



```
ggplot(data = storms) +  
  geom_point(mapping = aes(x = pressure, y = wind, color = status))
```



Sensible default
colors chosen

Channel
(aesthetic
property)

Variable to
map to the
channel

status
● hurricane
● tropical depression
● tropical storm

Legend added
automatically

To make a graph

[template]

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



To make a graph

mpg	cyl	disp	hp
21.0	6	160.0	2
21.0	6	160.0	2
22.8	4	108.0	1
21.4	6	258.0	2
18.7	8	360.0	3
18.1	6	225.0	2
14.3	8	360.0	5
24.4	4	146.7	1
22.8	4	140.8	1
19.2	6	167.6	2
17.8	6	167.6	2
16.4	8	275.8	3
17.3	8	275.8	3
15.2	8	275.8	3
10.4	8	472.0	4
10.4	8	460.0	4
14.7	8	440.0	4
32.4	4	78.7	1
30.4	4	75.7	1
33.9	4	71.1	1

1. Pick a **data** set

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

data



To make a graph

mpg	cyl	disp	hp	graph
21.0	6	160.0	2	●
21.0	6	160.0	2	●
22.8	4	108.0	1	●
21.4	6	258.0	2	●
18.7	8	360.0	3	●
18.1	6	225.0	2	●
14.3	8	360.0	5	●
24.4	4	146.7	1	●
22.8	4	140.8	1	●
19.2	6	167.6	2	●
17.8	6	167.6	2	●
16.4	8	275.8	3	●
17.3	8	275.8	3	●
15.2	8	275.8	3	●
10.4	8	472.0	4	●
10.4	8	460.0	4	●
14.7	8	440.0	4	●
32.4	4	78.7	1	●
30.4	4	75.7	1	●
33.9	4	71.1	1	●

data geom

1. Pick a **data** set

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

2. Choose a **geom**
to display cases



To make a graph

mappings

mpg	cyl	disp	hp	fill
21.0	6	160.0	2	●
21.0	6	160.0	2	●
22.8	4	108.0	1	●
21.4	6	258.0	2	●
18.7	8	360.0	3	●
18.1	6	225.0	2	●
14.3	8	360.0	5	●
24.4	4	146.7	1	●
22.8	4	140.8	1	●
19.2	6	167.6	2	●
17.8	6	167.6	2	●
16.4	8	275.8	3	●
17.3	8	275.8	3	●
15.2	8	275.8	3	●
10.4	8	472.0	4	●
10.4	8	460.0	4	●
14.7	8	440.0	4	●
32.4	4	78.7	1	●
30.4	4	75.7	1	●
33.9	4	71.1	1	●

data

geom

1. Pick a **data** set

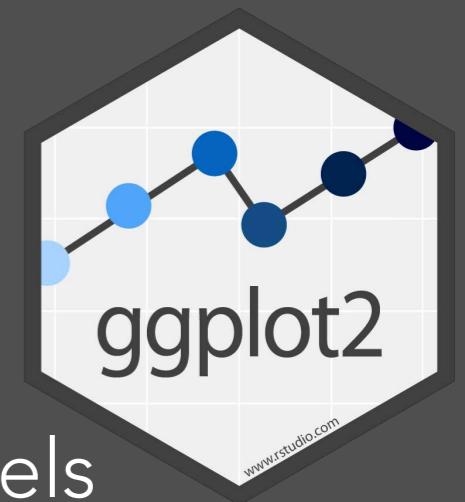
```
ggplot(data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

2. Choose a **geom**
to display cases

3. **Map** aesthetic
properties to
variables



YOUR TURN



- In your R script, experiment with different visual channels by mapping color, shape, size, and alpha to storm status.
- What happens when you map the same variable to multiple aesthetics?
 - Hint: `color = status`, `shape = status`.
- Try mapping color to the variable category rather than status. What changes and why?
- What happens if you map color to two different variables (both status and category)?



TYPES OF CHANNELS

→ **Magnitude Channels: Ordered Attributes**

Position on common scale



Position on unaligned scale



Length (1D size)



Tilt/angle



Area (2D size)



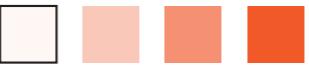
Depth (3D position)



Color luminance



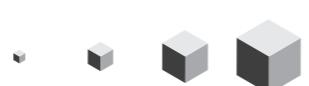
Color saturation



Curvature



Volume (3D size)



→ **Identity Channels: Categorical Attributes**

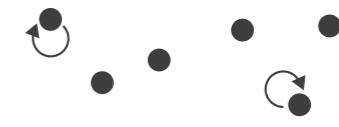
Spatial region



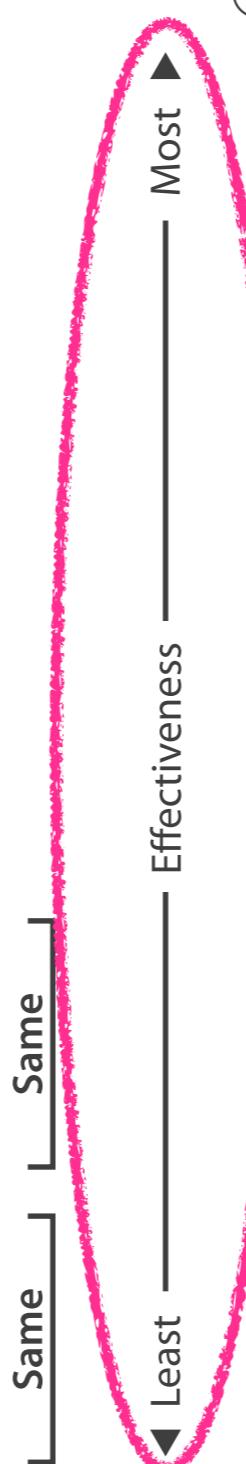
Color hue



Motion

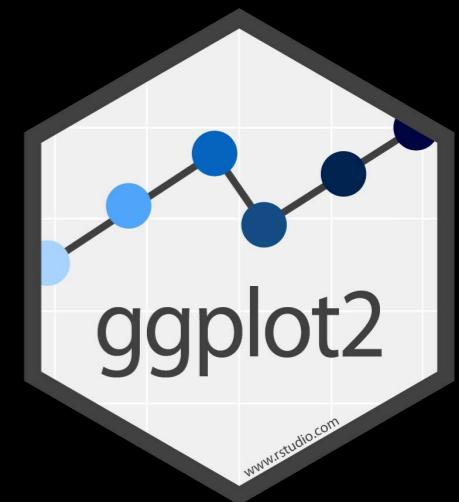


Shape



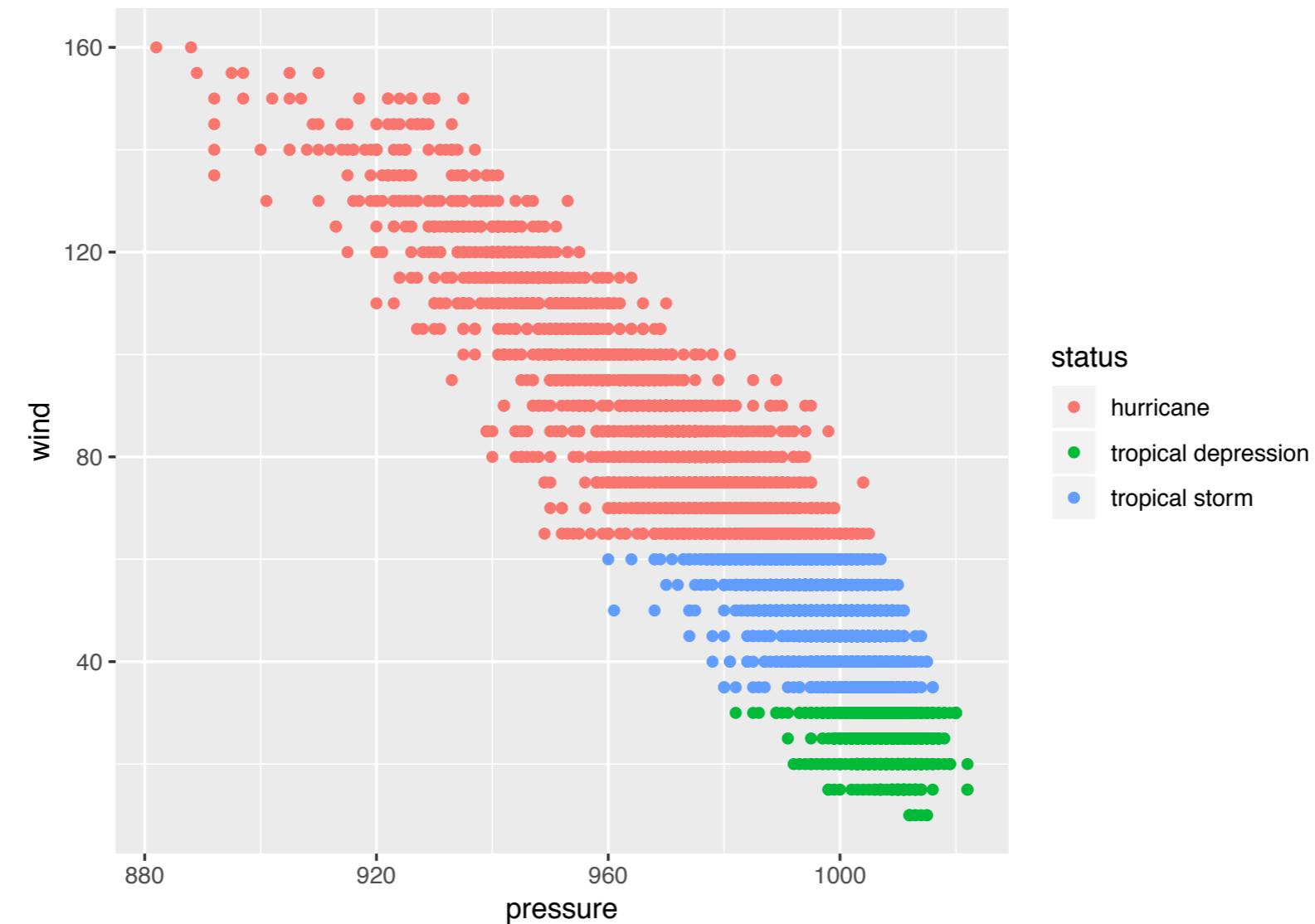
MAPPING VS SETTING AESTHETIC PROPERTIES

LET'S GO BACK TO THIS PLOT...

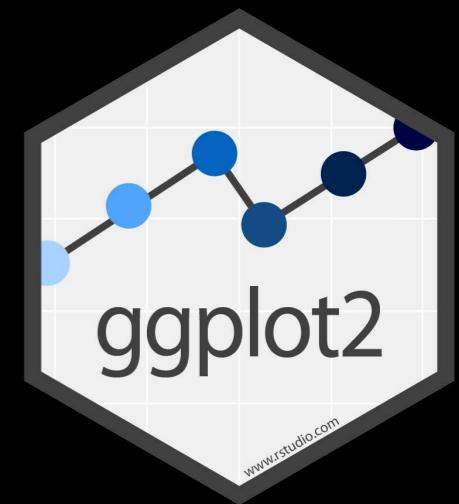


First, a time-saver...

```
ggplot(data = storms) +  
  geom_point(mapping = aes(x = pressure, y = wind, color = status))
```

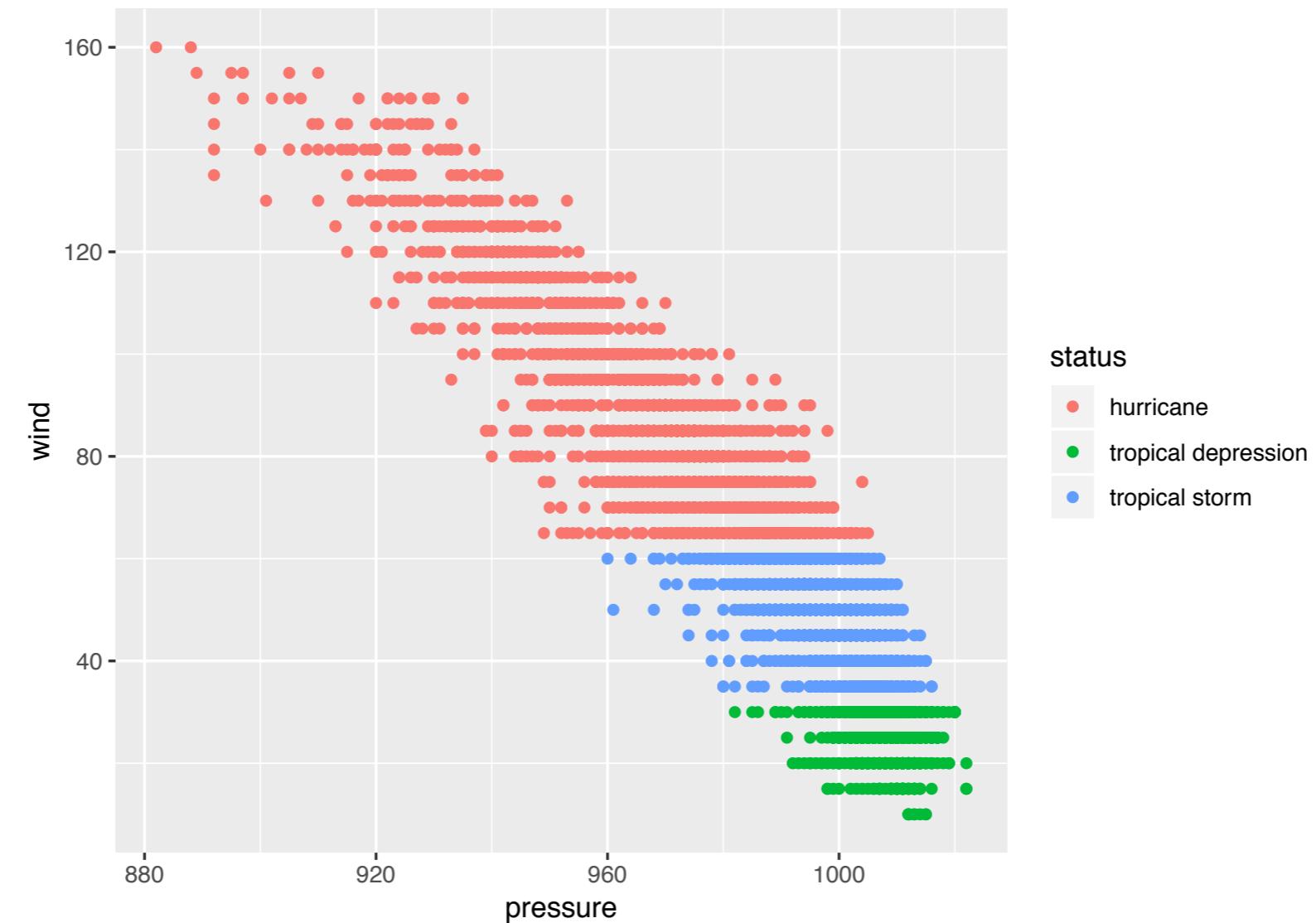


LET'S GO BACK TO THIS PLOT...

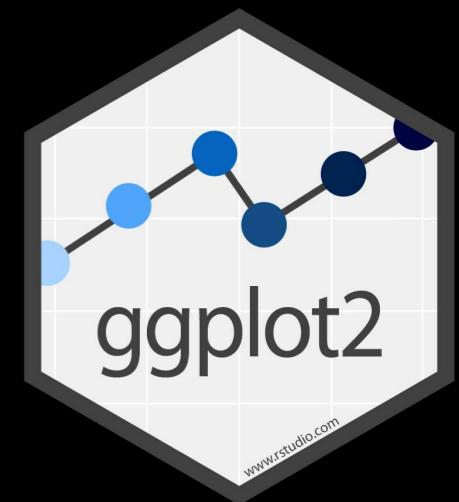


...these can be omitted

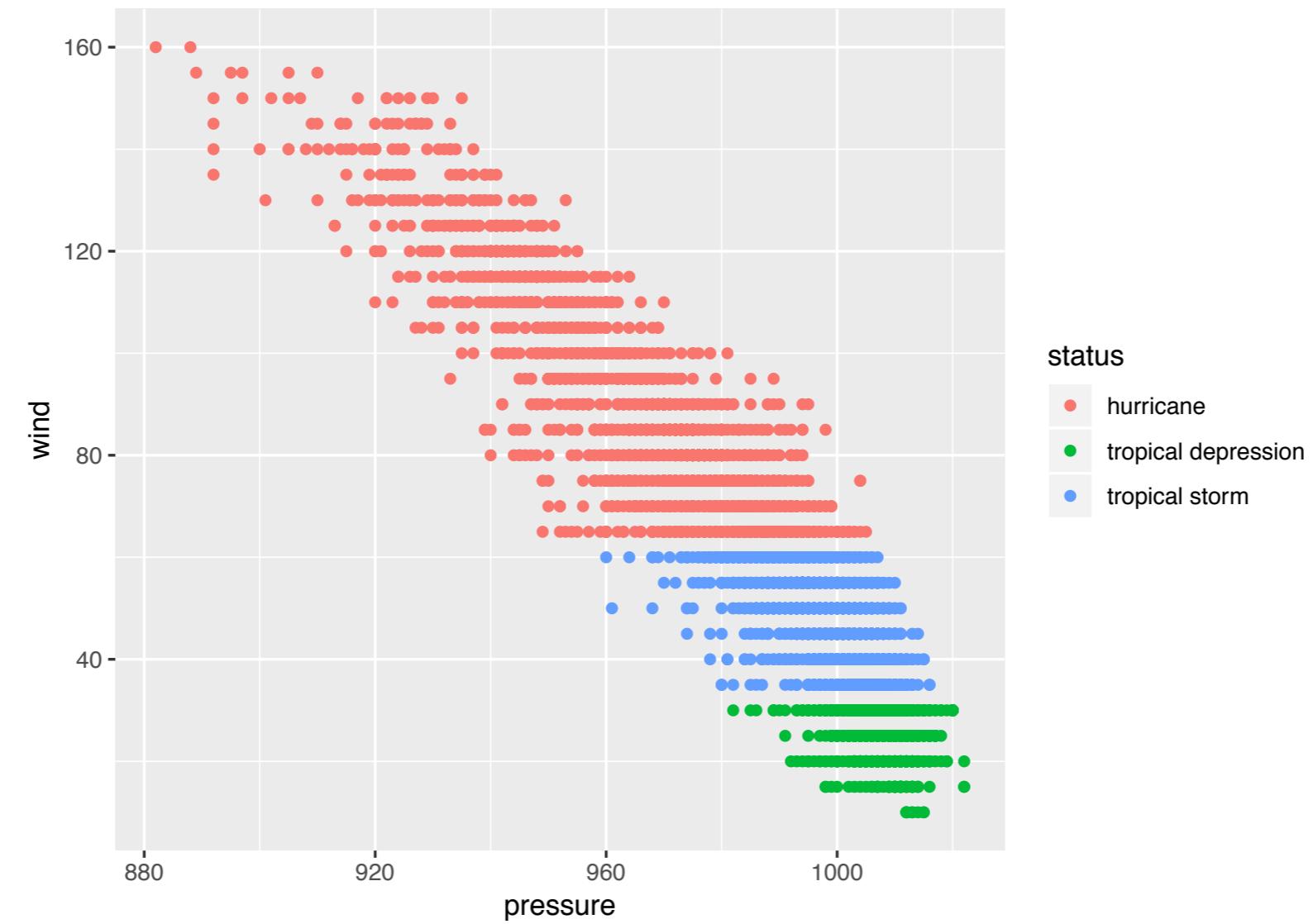
```
ggplot(data = storms) +  
  geom_point(mapping = aes(x = pressure, y = wind, color = status))
```



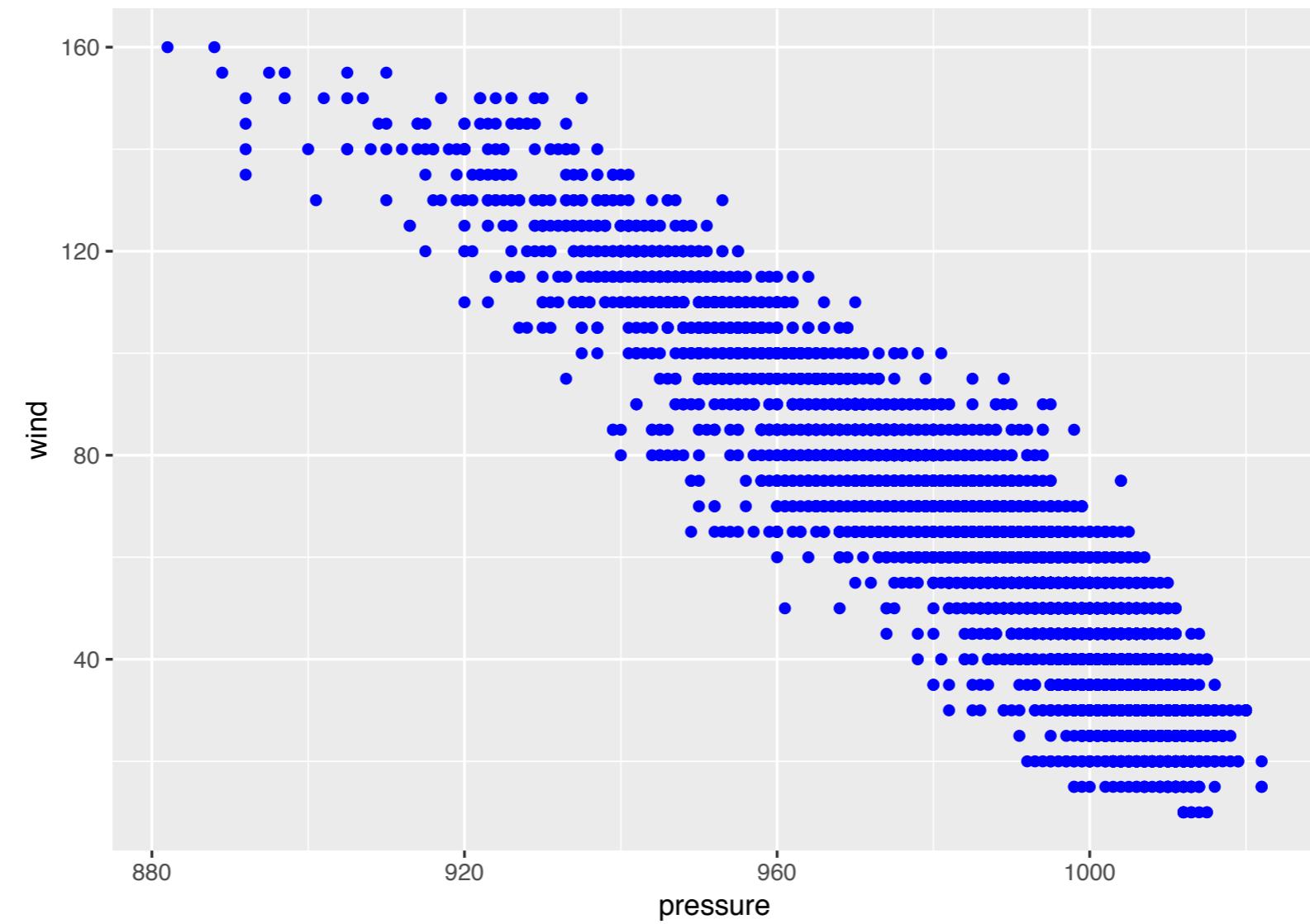
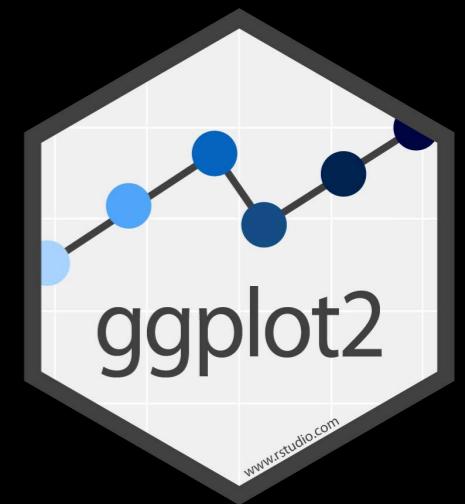
LET'S GO BACK TO THIS
PLOT...



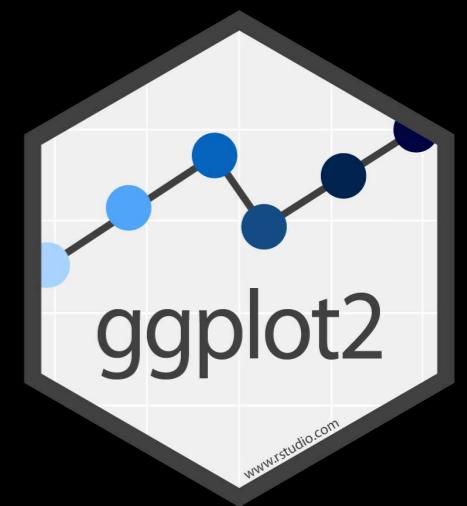
```
ggplot(storms) +  
  geom_point(aes(x = pressure, y = wind, color = status))
```



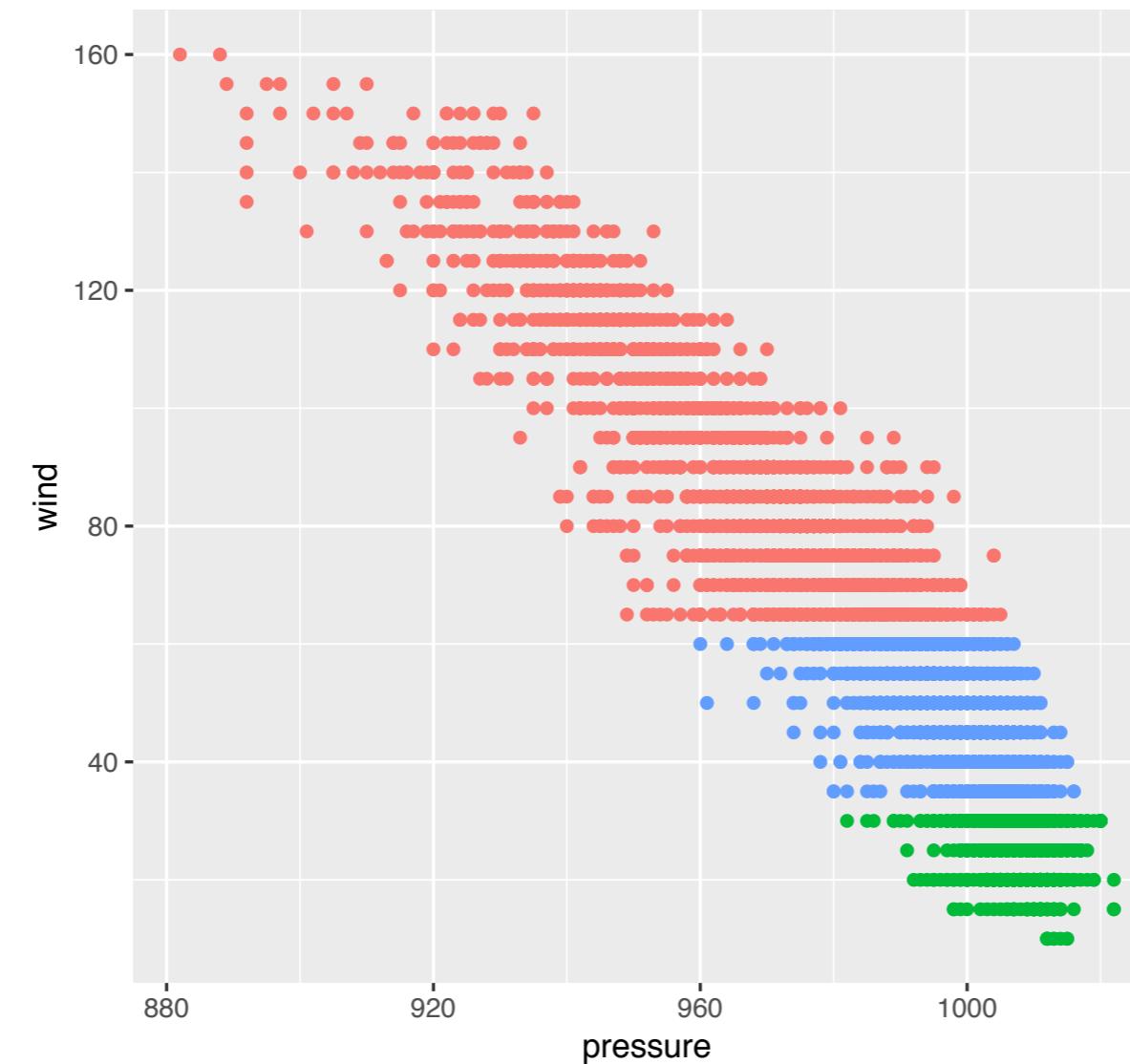
HOW WOULD YOU MAKE THIS PLOT?



AESTHETICS



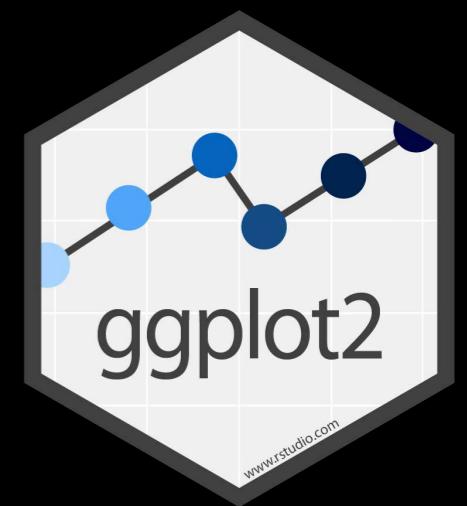
```
ggplot(storms) +  
  geom_point(aes(x = pressure, y = wind, color = status))
```



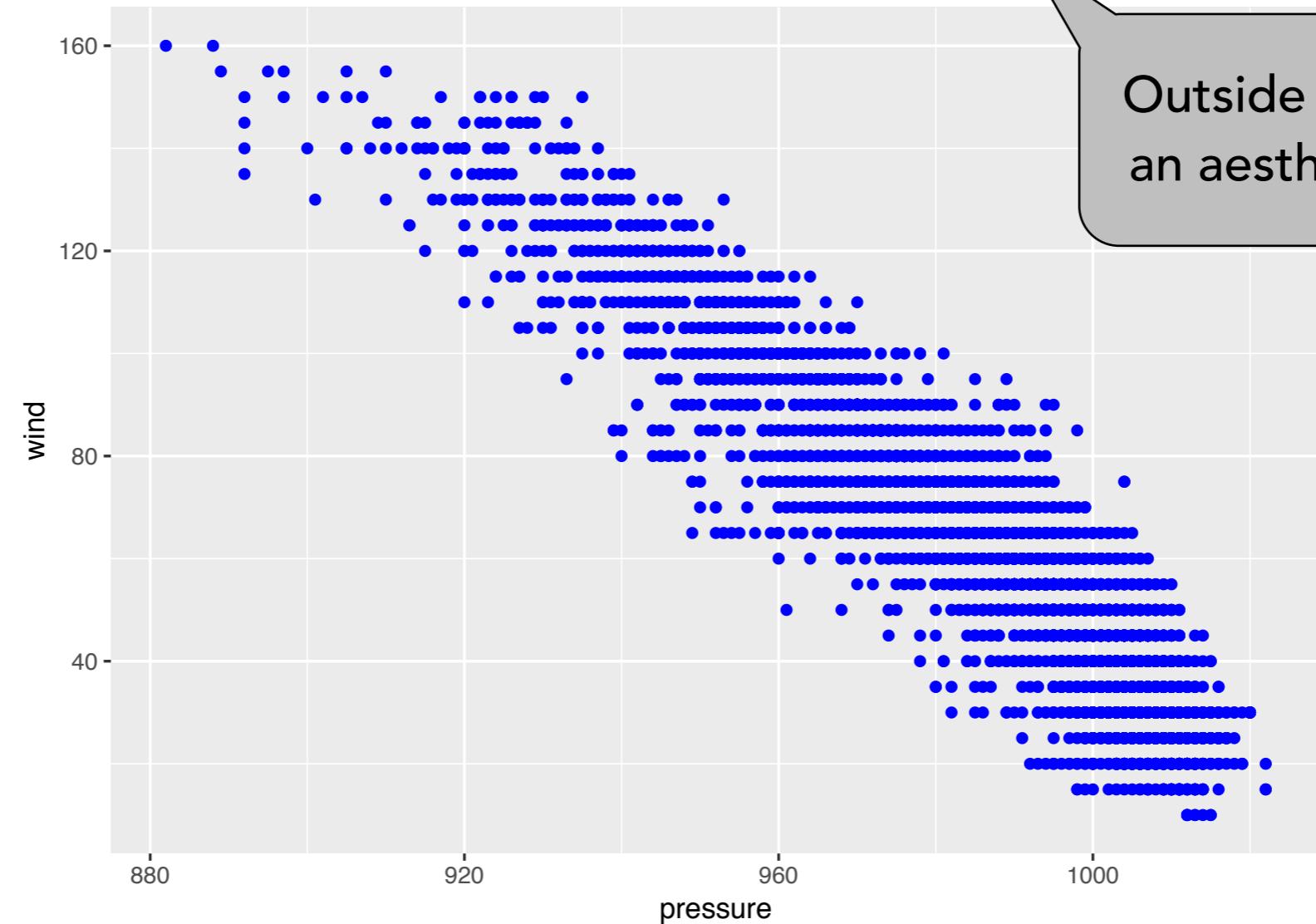
Inside of aes(): maps an aesthetic to a variable

status	
●	hurricane
●	tropical depression
●	tropical storm

AESTHETICS

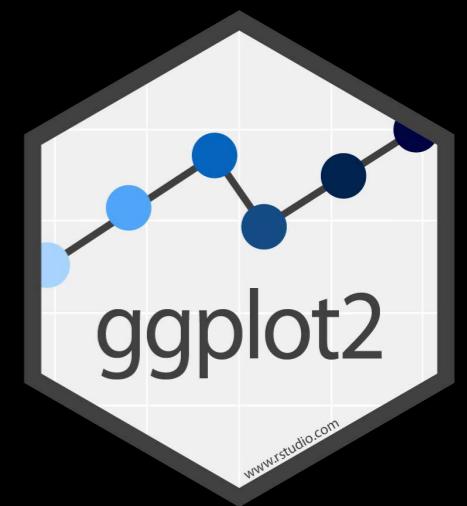


```
ggplot(storms) +  
  geom_point(aes(x = pressure, y = wind), color = "blue")
```

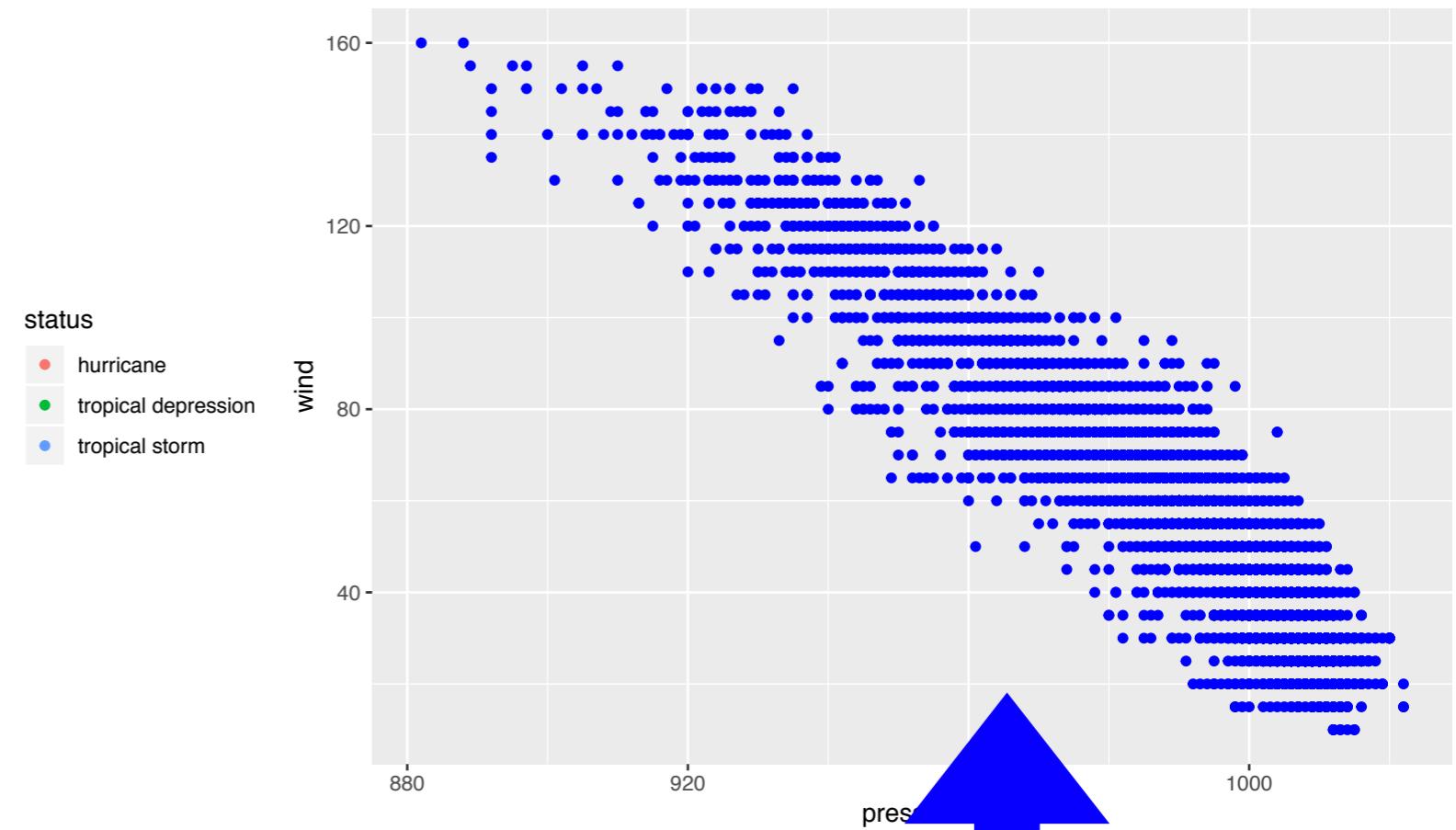
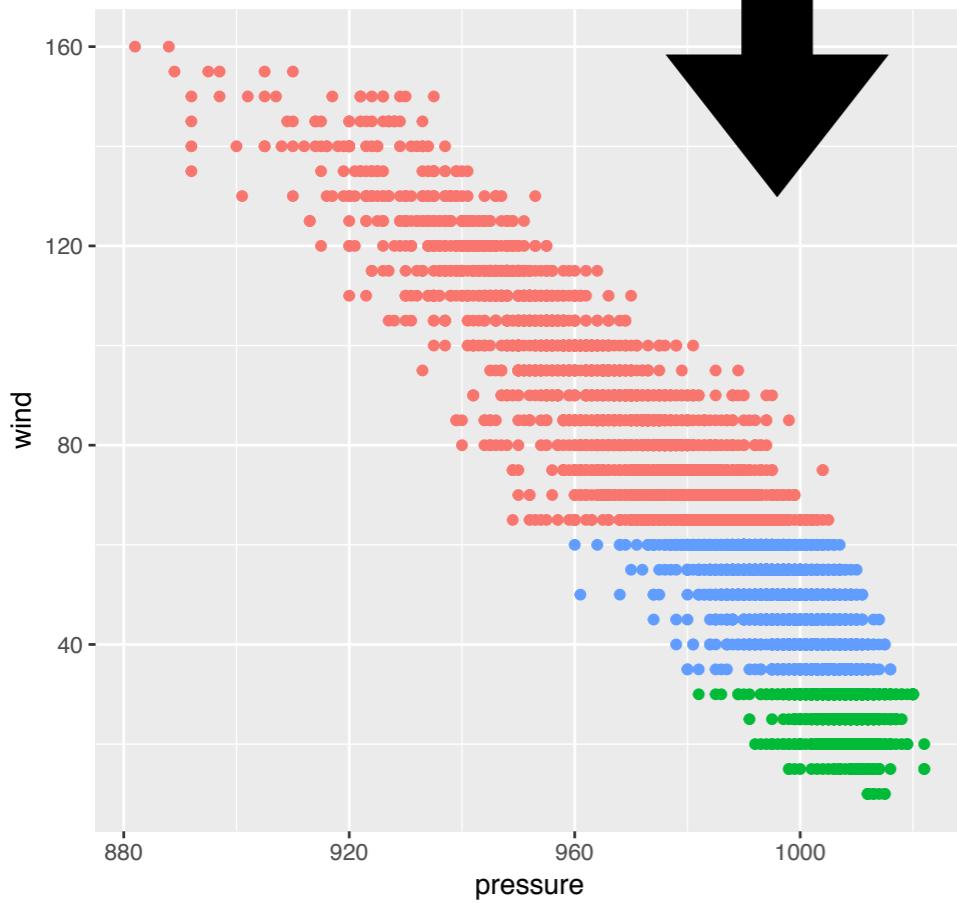


Outside of aes(): sets
an aesthetic to a value

AESTHETICS

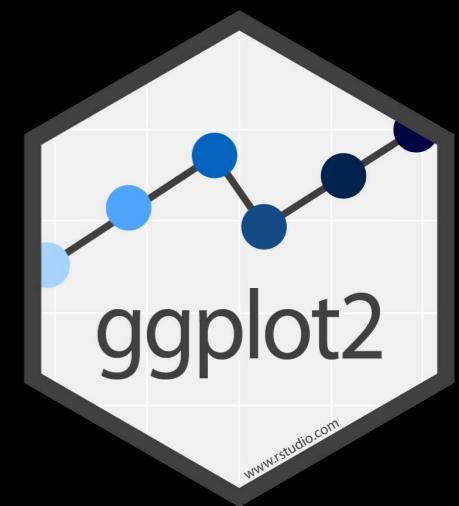


```
ggplot(storms) +  
  geom_point(aes(x = pressure, y = wind, color = status))
```



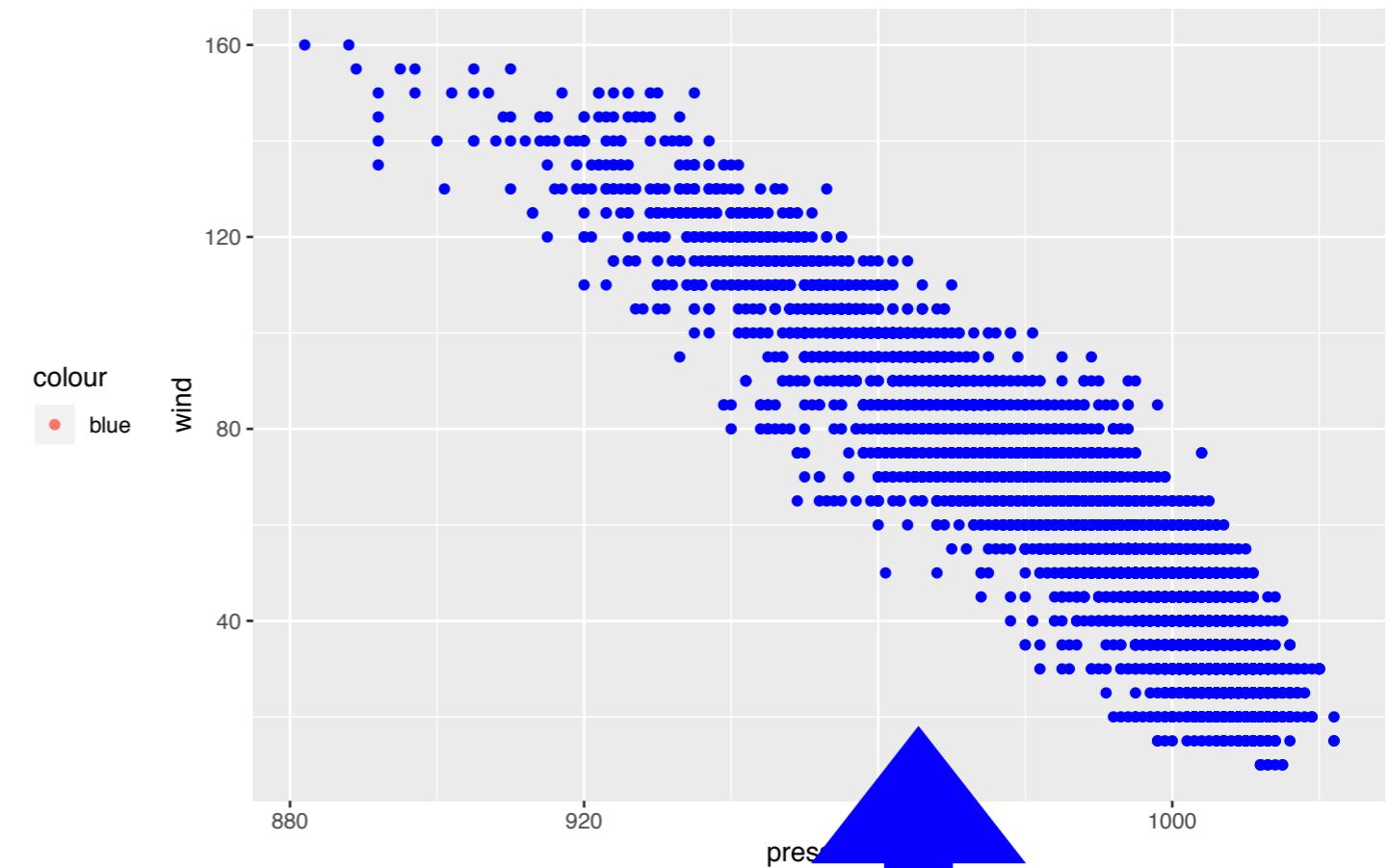
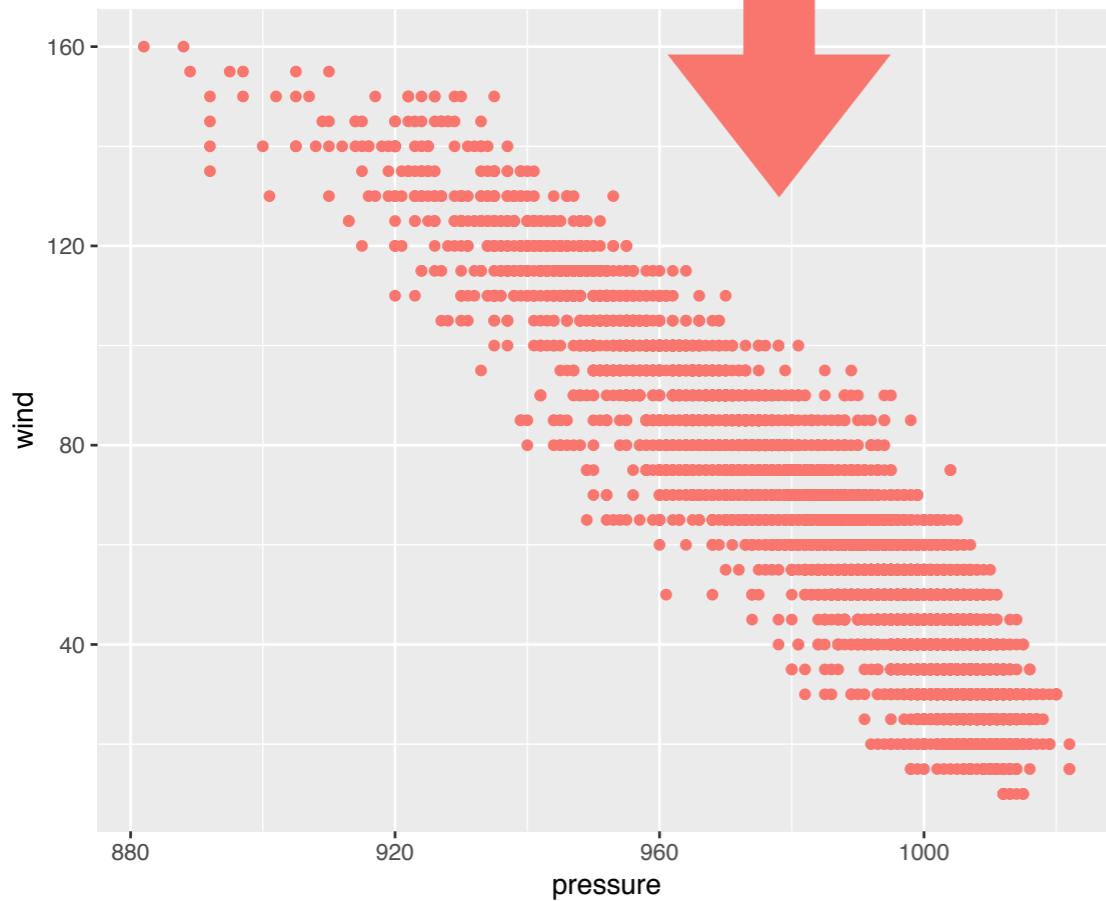
```
ggplot(storms) +  
  geom_point(aes(x = pressure, y = wind), color = "blue")
```

AESTHETICS



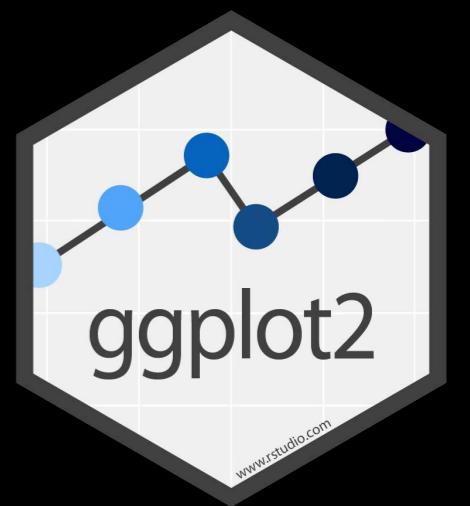
Watch out for this!

```
ggplot(storms) +  
  geom_point(aes(x = pressure, y = wind, color = "blue"))
```

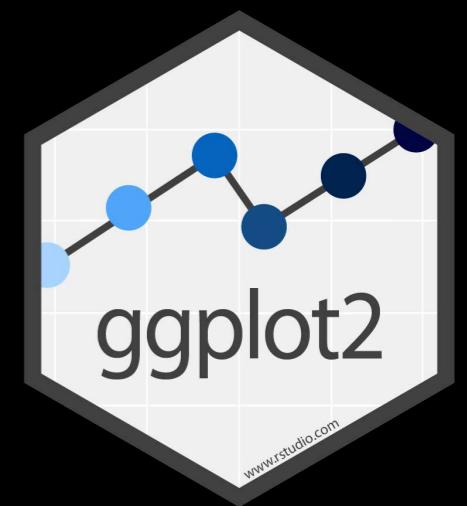


```
ggplot(storms) +  
  geom_point(aes(x = pressure, y = wind), color = "blue")
```

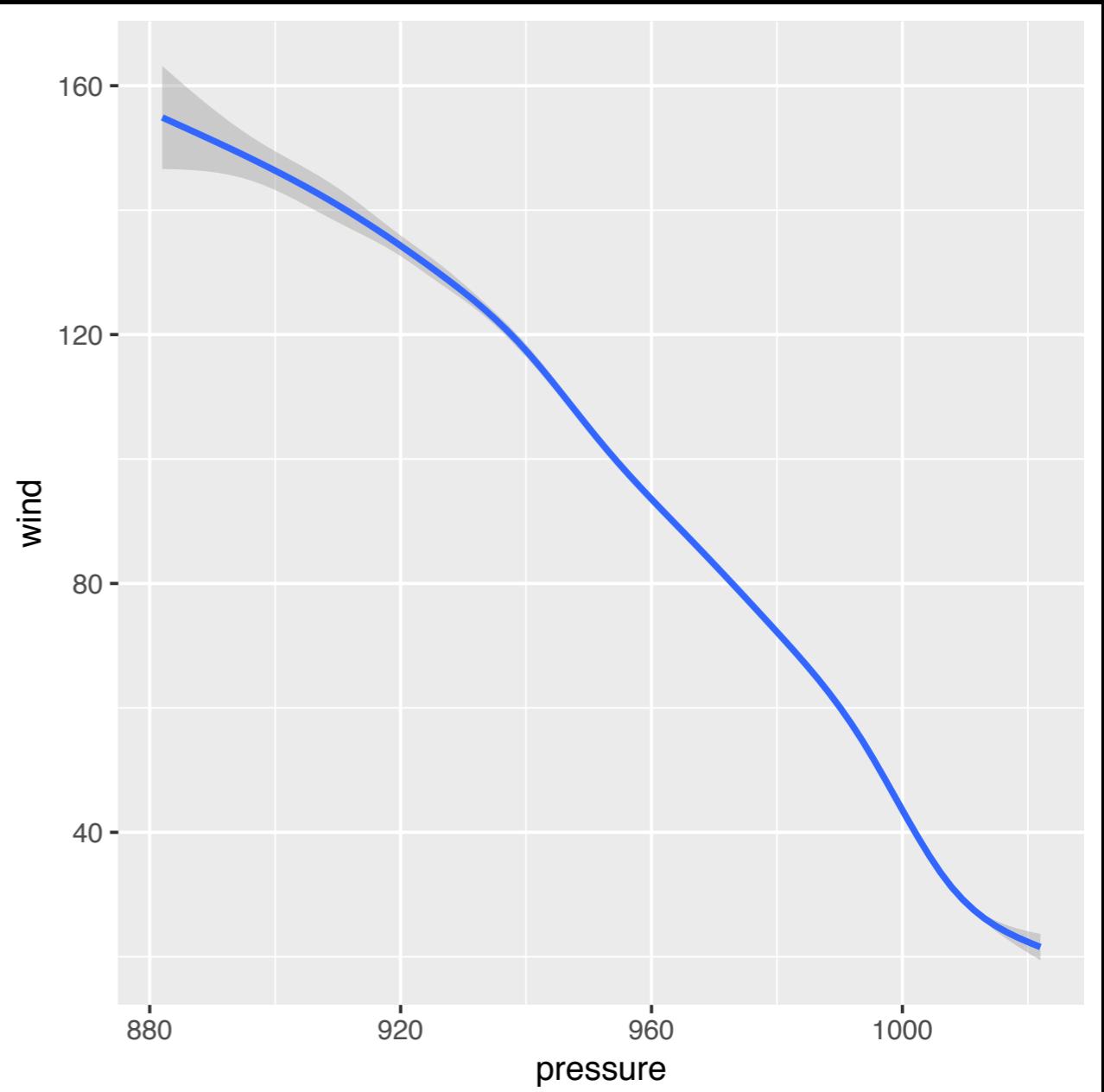
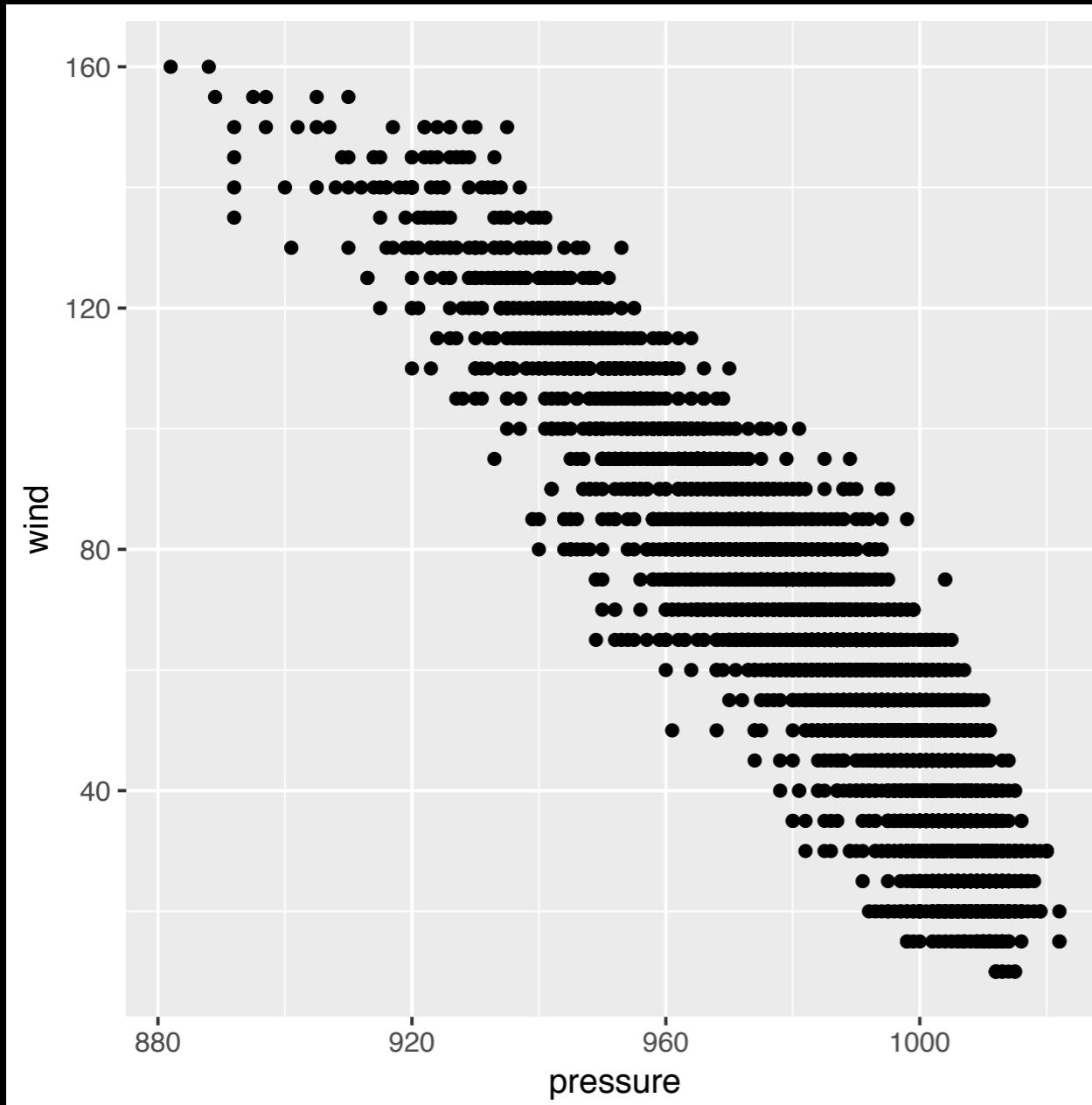
GEOMS



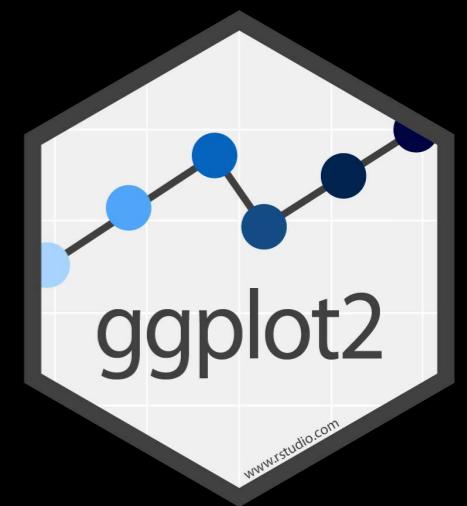
GEOMS



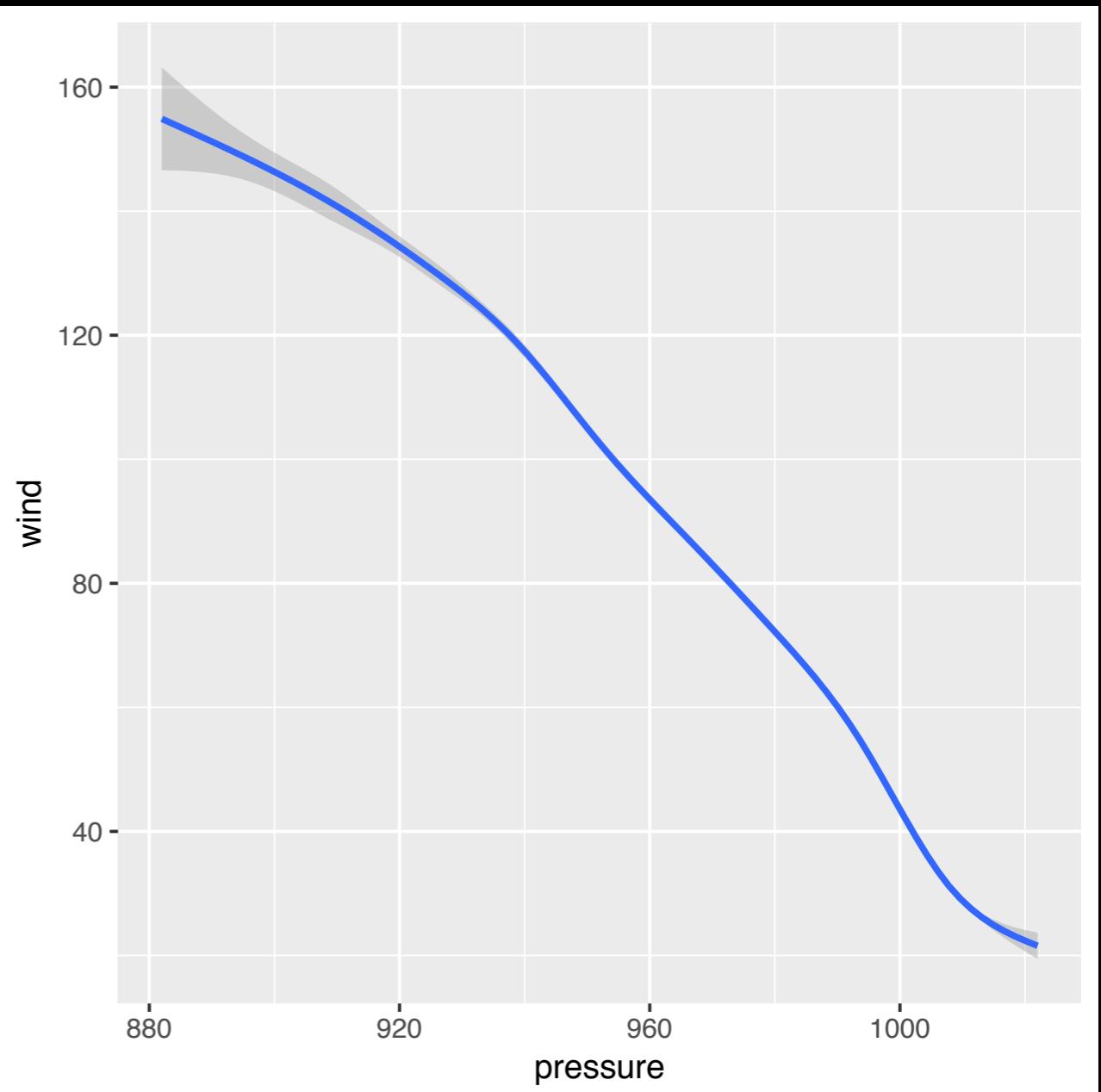
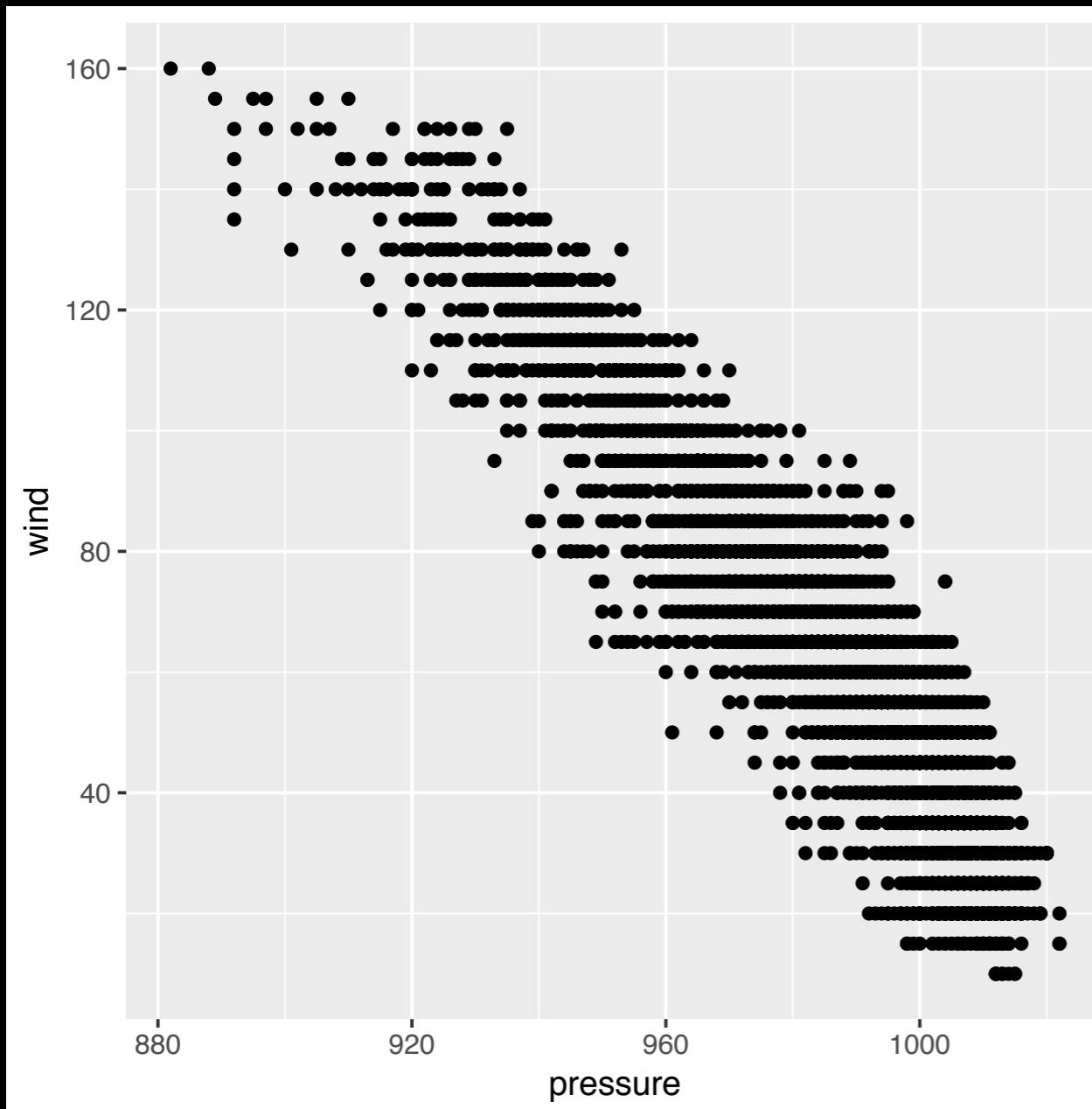
- How are these plots similar?



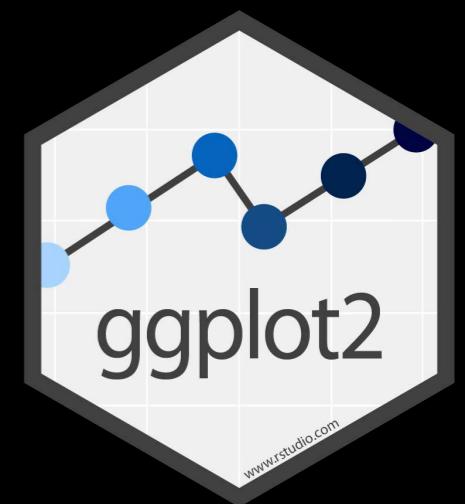
GEOMS



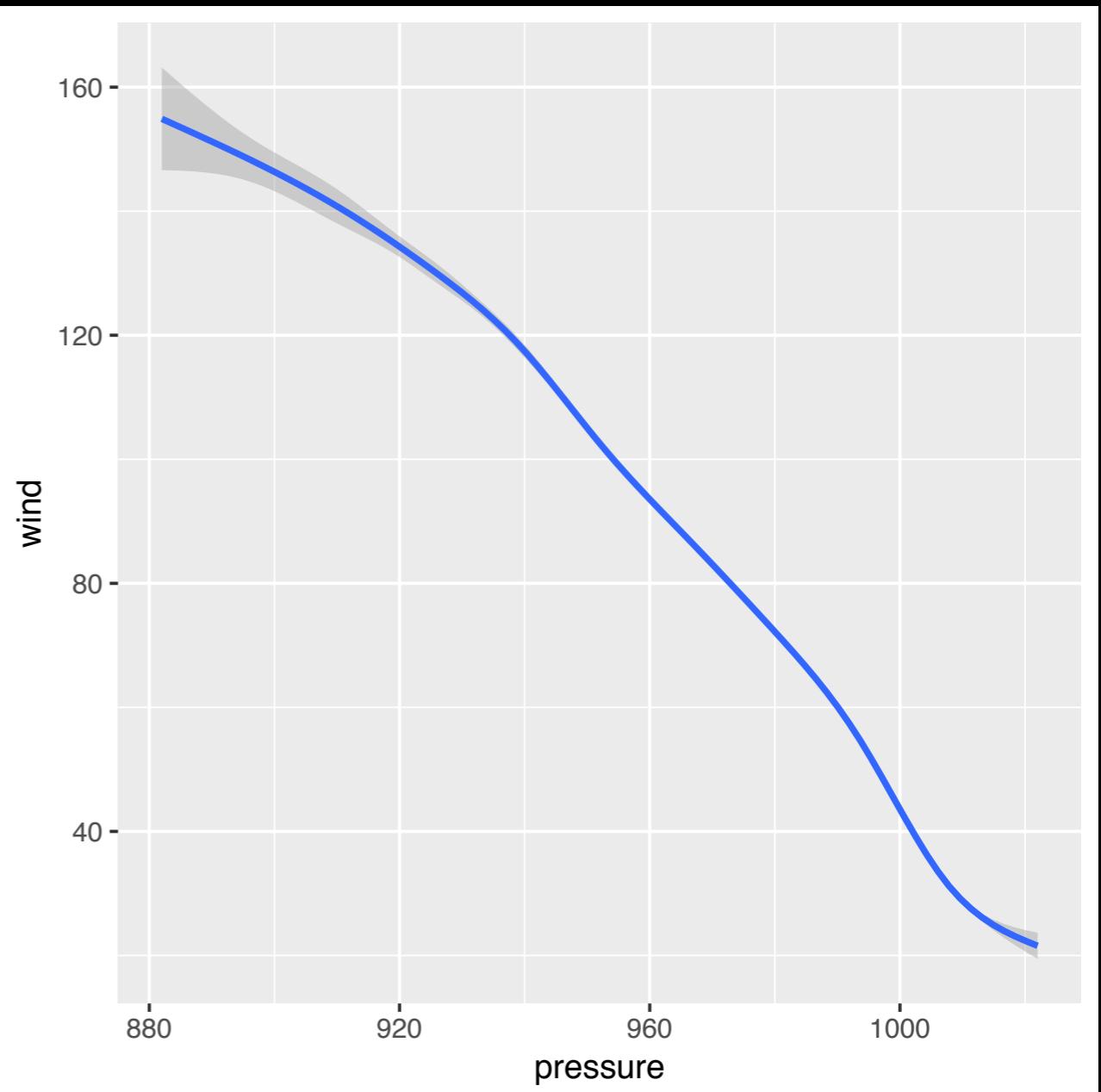
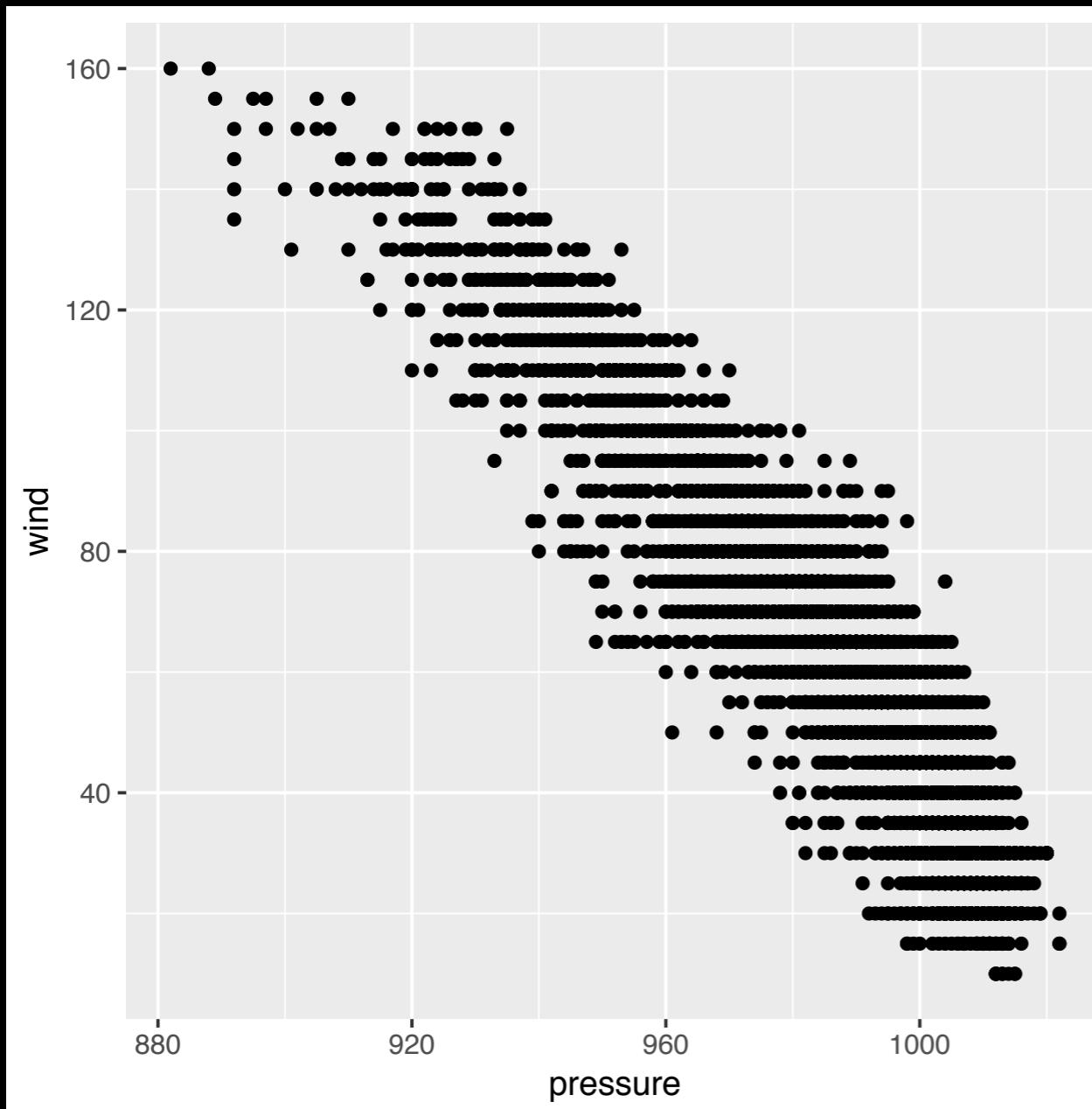
- Same: x variable, y variable → built with same data



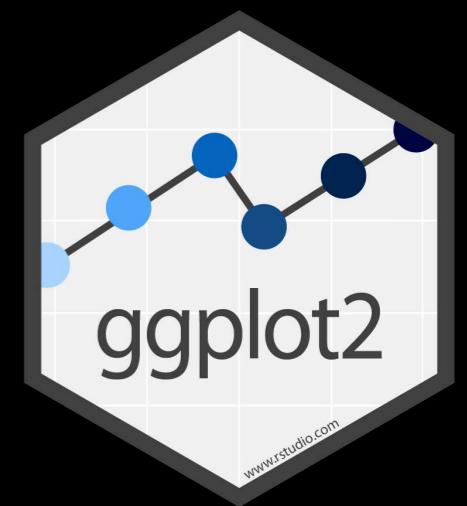
GEOMS



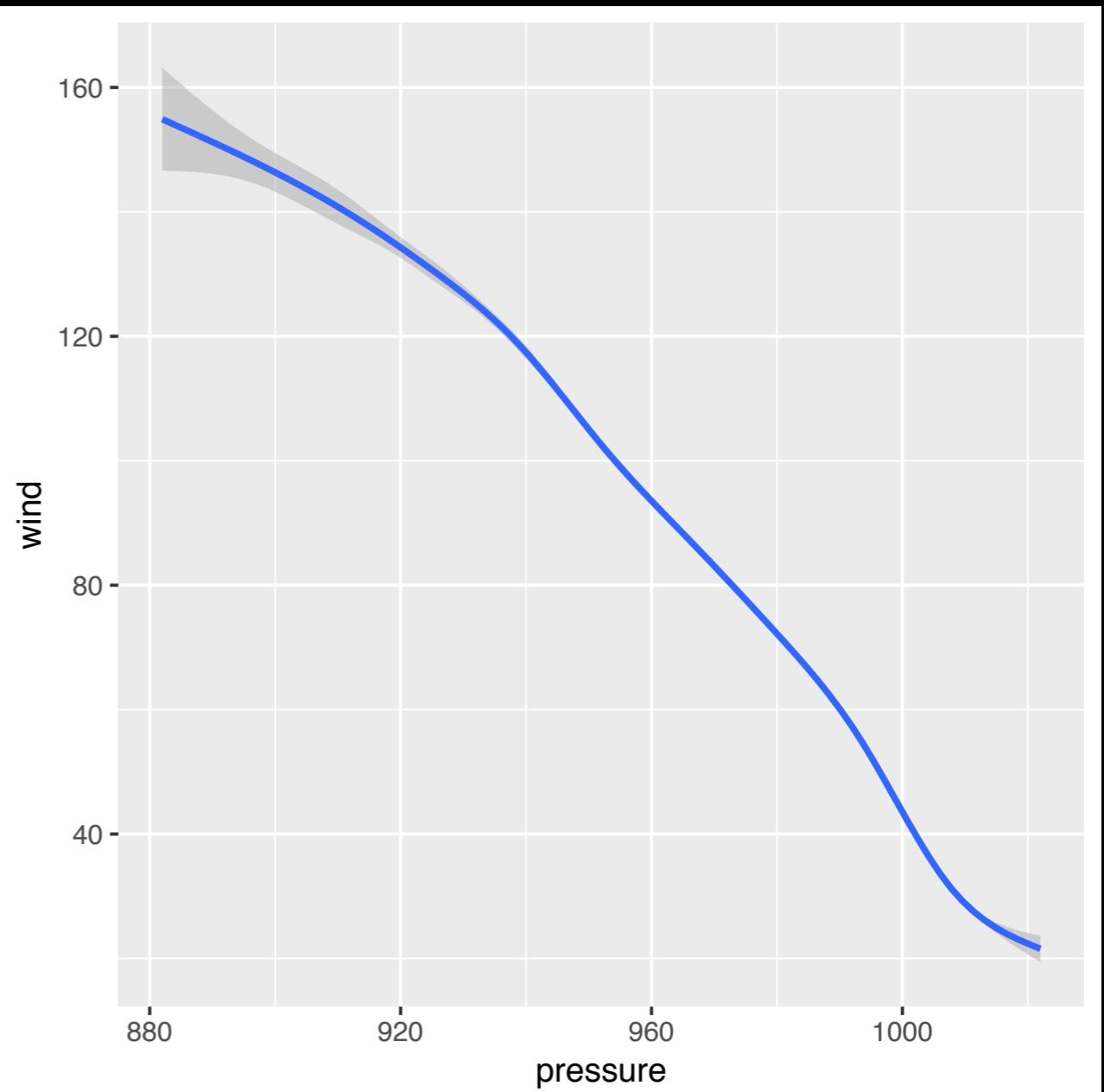
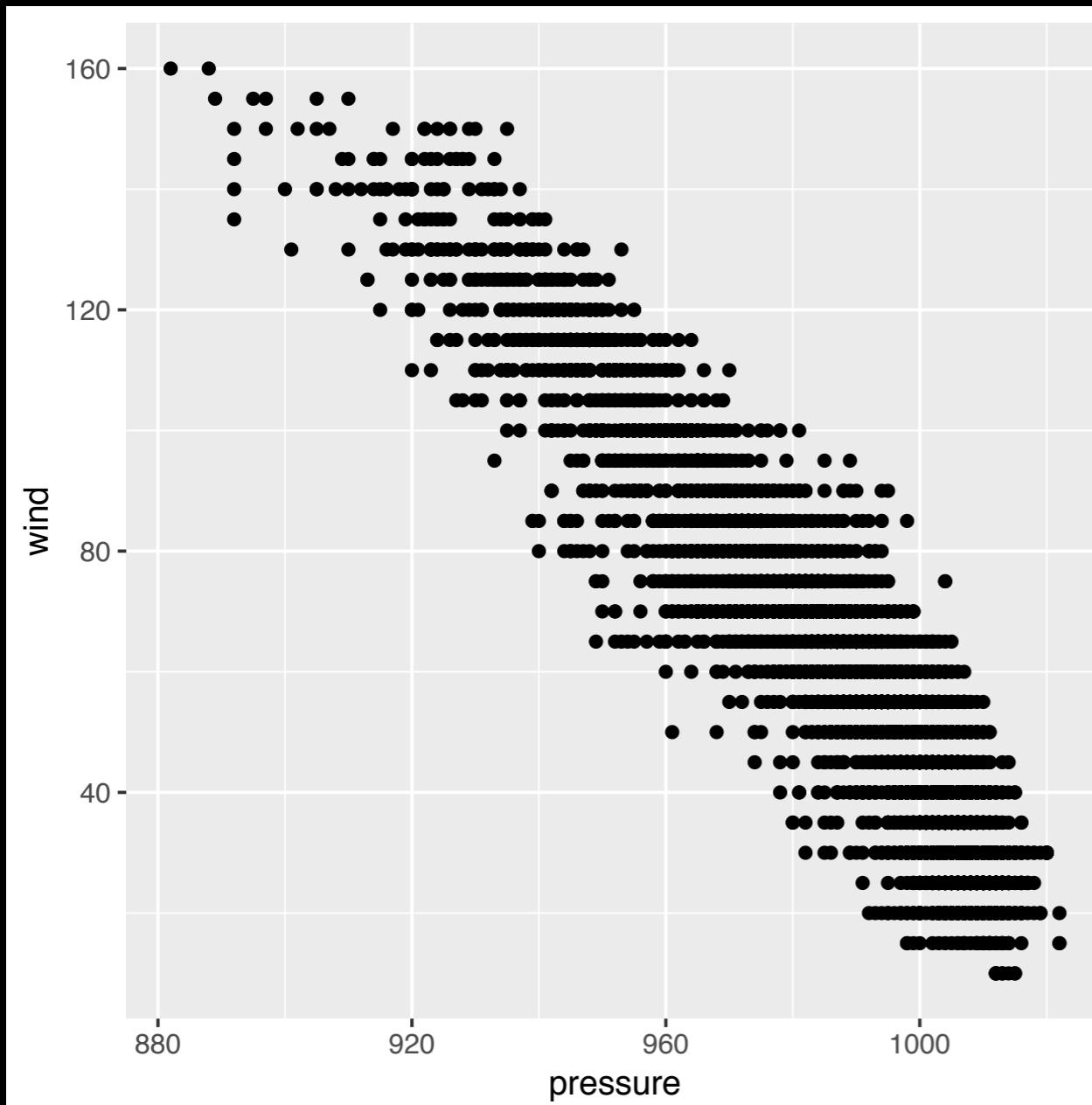
- How are these plots different?



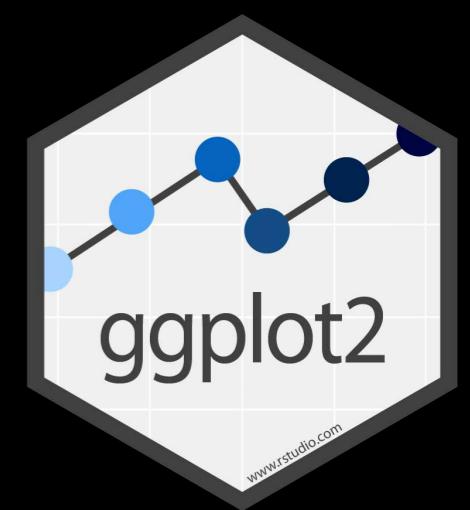
GEOMS



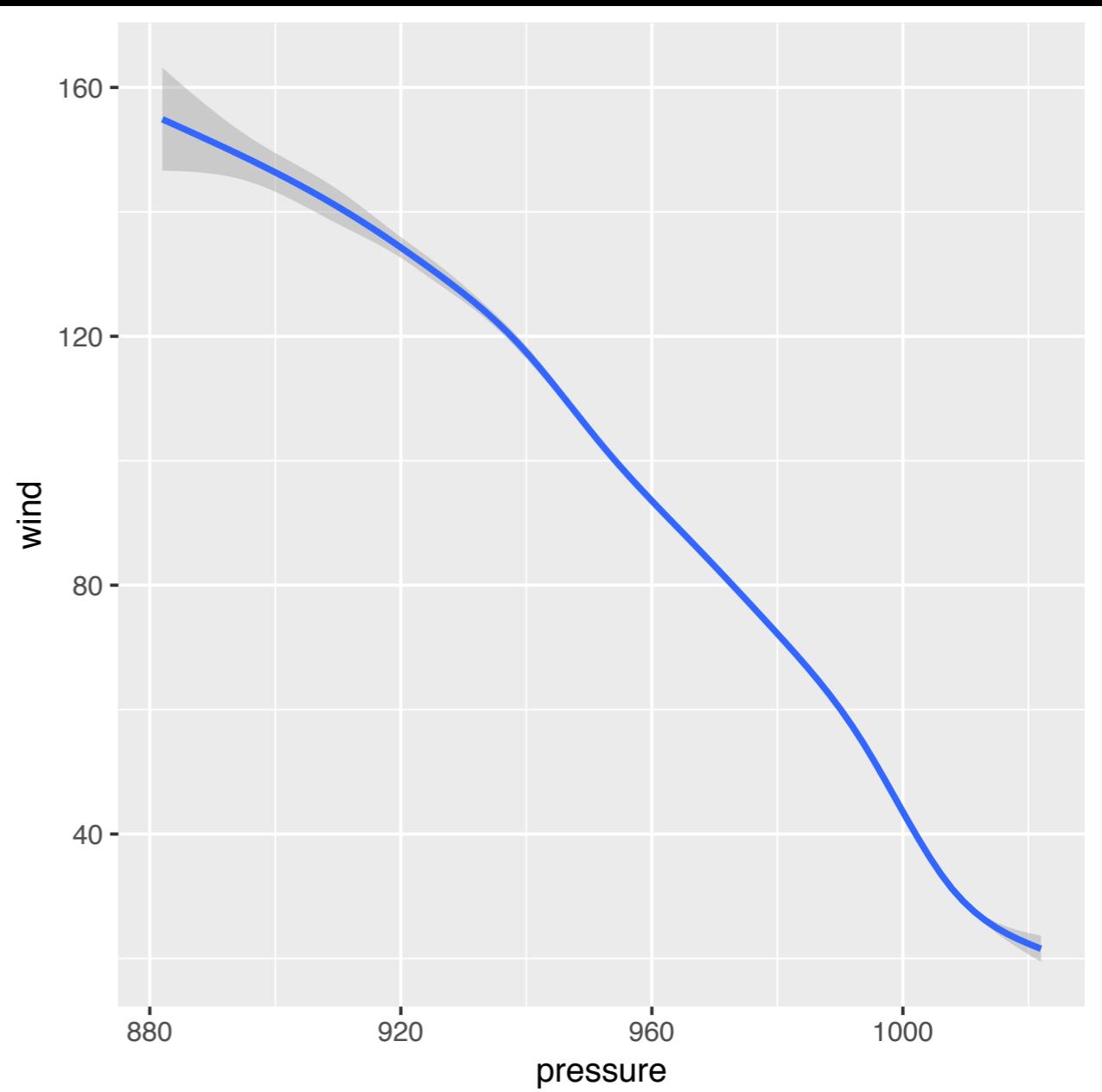
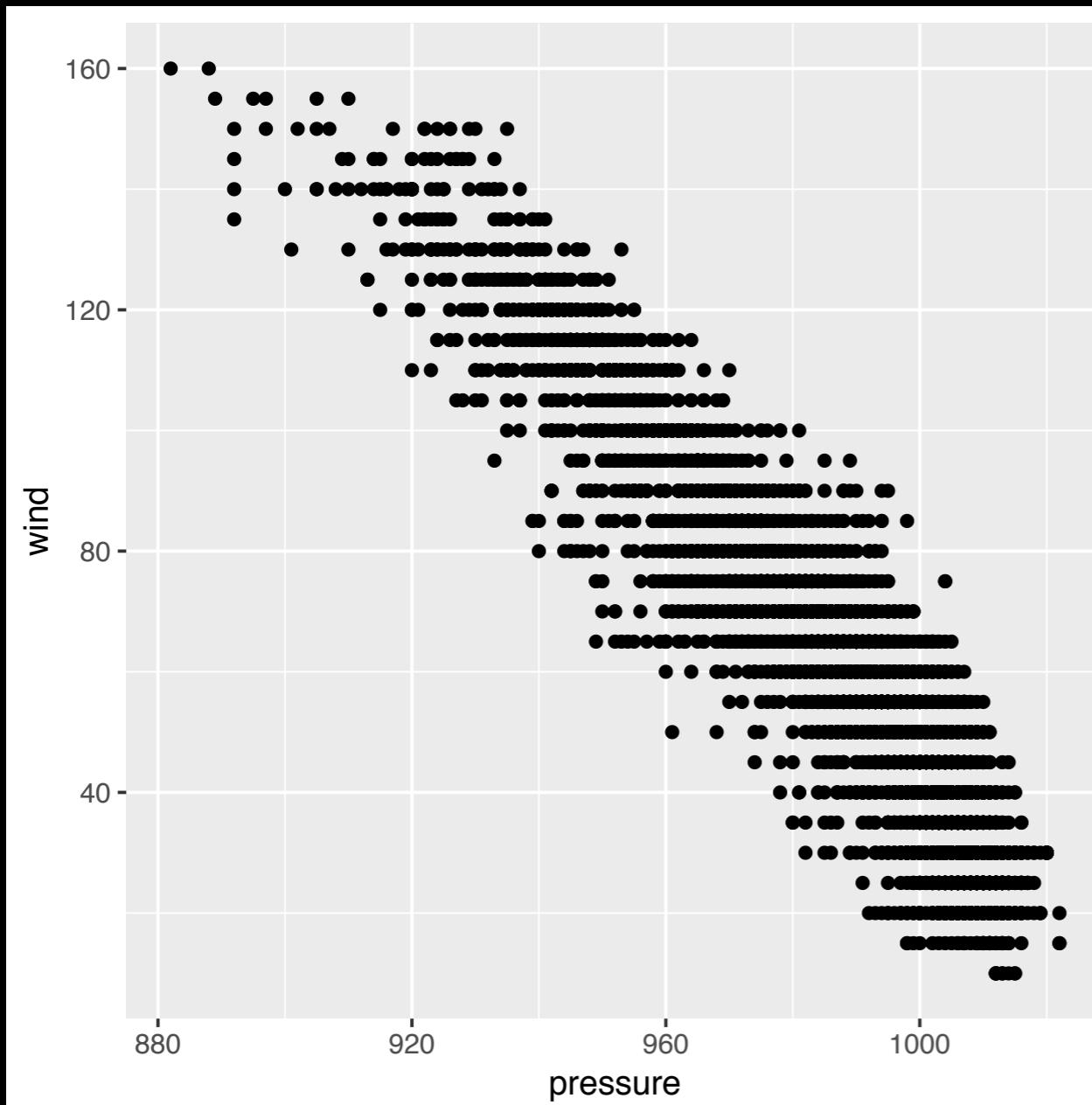
- Different: geometric object (point vs. line & ribbon)



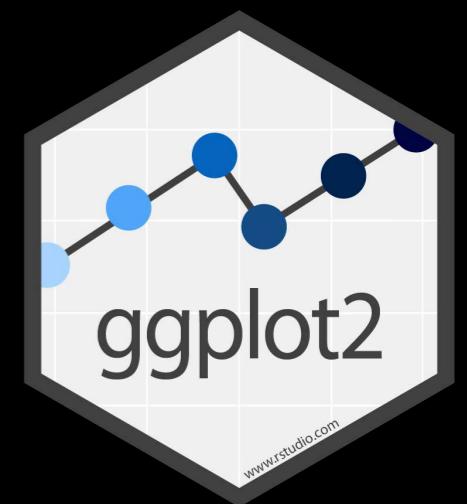
GEOMS



- Different: statistical summarization (identity vs. smooth)



GEOMS



- Remember our template:

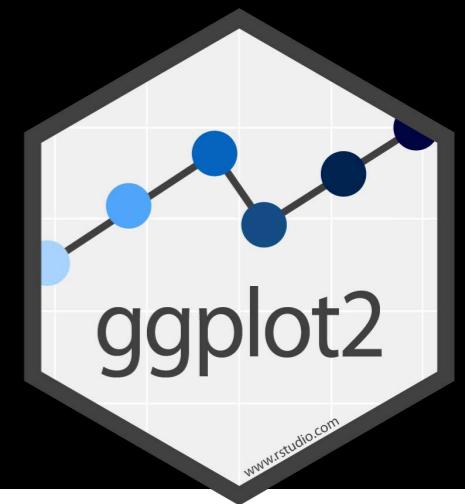
```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>))
```

GEOMS

Each `geom_` function
requires a mapping



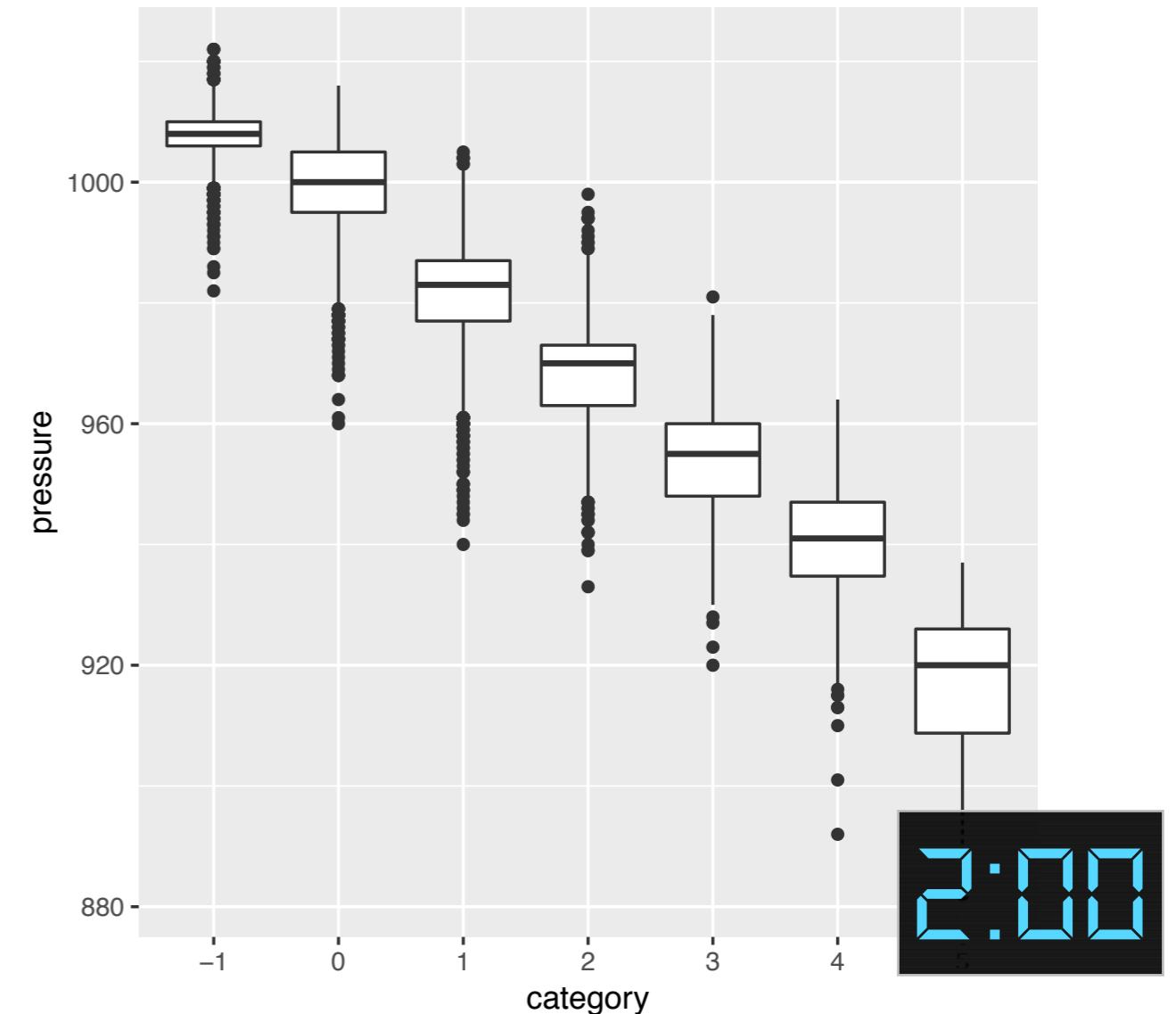
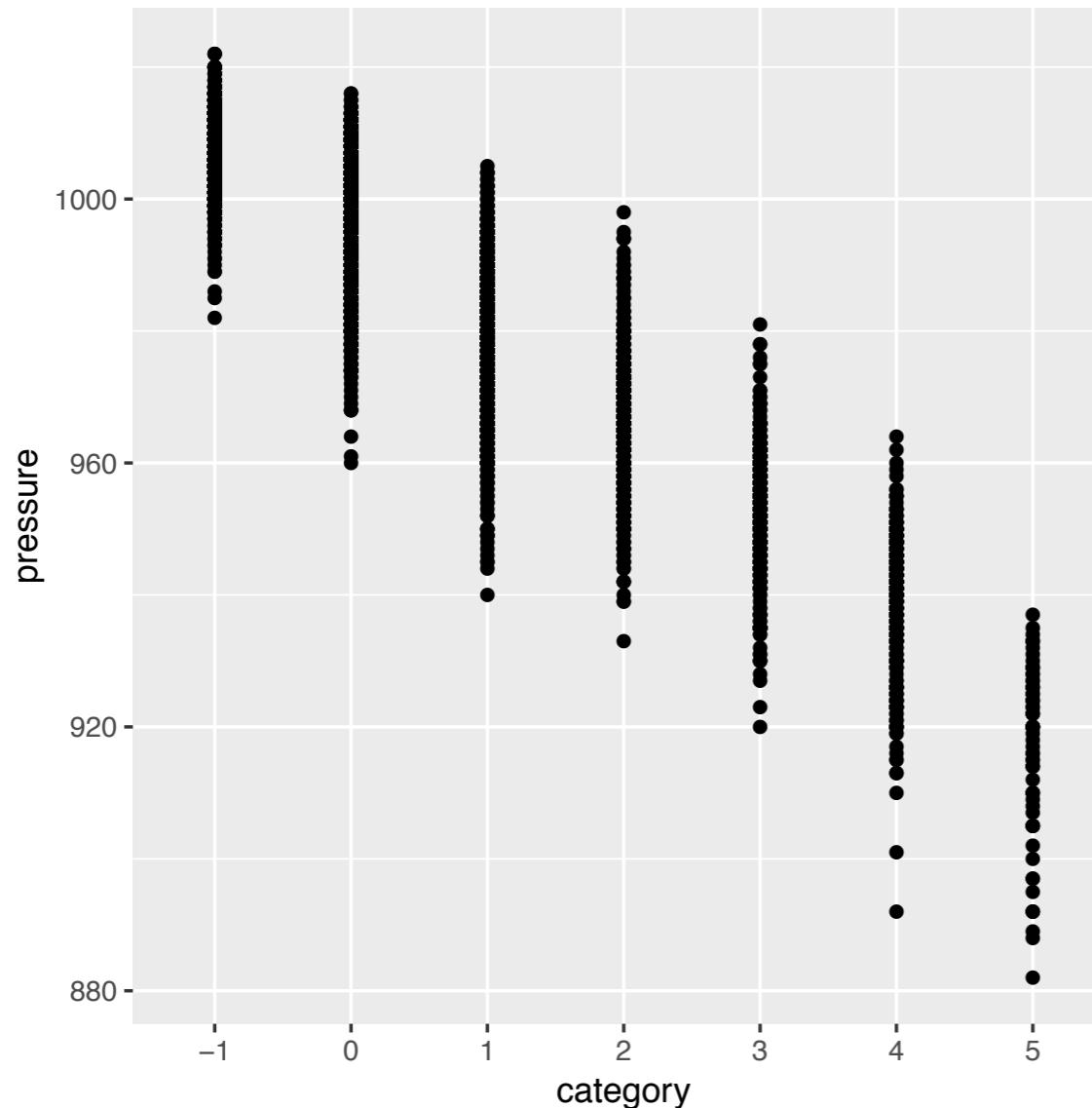
Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.	
One Variable <ul style="list-style-type: none"> Continuous <pre>a <- ggplot(mpg, aes(hwy)) a + geom_area(stat = "bin") x, y, alpha, color, fill, linetype, size b + geom_area(aes(y = ..density..), stat = "bin") a + geom_density(kernel = "gaussian") x, y, alpha, color, fill, linetype, size, weight b + geom_density(aes(y = ..county..)) a + geom_dotplot() x, y, alpha, color, fill a + geom_freqpoly() x, y, alpha, color, linetype, size b + geom_freqpoly(aes(y = ..density..)) a + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight b + geom_histogram(aes(y = ..density..))</pre> Discrete <pre>b <- ggplot(mpg, aes(fl)) b + geom_bar() x, alpha, color, fill, linetype, size, weight</pre> 	Two Variables <ul style="list-style-type: none"> Continuous X, Continuous Y <pre>f <- ggplot(mpg, aes(cty, hwy)) f + geom_blank() (Useful for expanding limits) f + geom_jitter() x, y, alpha, color, fill, shape, size f + geom_point() x, y, alpha, color, fill, shape, size f + geom_quantile() x, y, alpha, color, linetype, size, weight f + geom_rug(sides = "bl") alpha, color, linetype, size f + geom_smooth(method = lm) x, y, alpha, color, fill, linetype, size, weight f + geom_text(aes(label = cty)) AB x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust</pre> Discrete X, Continuous Y <pre>g <- ggplot(mpg, aes(class, hwy)) g + geom_bar(stat = "identity") x, y, alpha, color, fill, linetype, size, weight g + geom_boxplot() lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, shape, size, weight g + geom_dotplot(binaxis = "y", stackdir = "center") x, y, alpha, color, fill g + geom_violin(scale = "area") x, y, alpha, color, fill, linetype, size, weight</pre> Discrete X, Discrete Y <pre>h <- ggplot(diamonds, aes(cut, color)) h + geom_jitter() x, y, alpha, color, fill, shape, size</pre>
Graphical Primitives <ul style="list-style-type: none"> Continuous <pre>map <- map_data("state") c <- ggplot(map, aes(long, lat)) c + geom_polygon(aes(group = group)) x, y, alpha, color, fill, linetype, size</pre> Discrete <pre>d <- ggplot(economics, aes(date, unemploy)) d + geom_path(lineend = "butt", linejoin = "round", linemitre = 1) x, y, alpha, color, linetype, size d + geom_ribbon(aes(ymin = unemploy - 900, ymax = unemploy + 900)) x, ymax, ymin, alpha, color, fill, linetype, size</pre> Discrete <pre>e <- ggplot(seals, aes(x = long, y = lat)) e + geom_segment(aes(xend = long + delta_long, yend = lat + delta_lat)) x, xend, y, yend, alpha, color, linetype, size e + geom_rect(aes(xmin = long, ymin = lat, xmax = long + delta_long, ymax = lat + delta_lat)) xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size</pre> 	Continuous Function <pre>j <- ggplot(economics, aes(date, unemploy)) j + geom_area() x, y, alpha, color, fill, linetype, size j + geom_line() x, y, alpha, color, linetype, size j + geom_step(direction = "hv") x, y, alpha, color, linetype, size</pre>
Visualizing error <pre>df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2) k <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se)) k + geom_crossbar(fatten = 2) x, y, ymax, ymin, alpha, color, fill, linetype, size k + geom_errorbar() x, ymax, ymin, alpha, color, linetype, size, width (also <code>geom_errorbarh()</code>) k + geom_linerange() x, ymin, ymax, alpha, color, linetype, size k + geom_pointrange() x, y, ymin, ymax, alpha, color, fill, linetype, shape, size</pre>	Maps <pre>data <- data.frame(murder = USArrests\$Murder, state = tolower(rownames(USArrests))) map <- map_data("state") l <- ggplot(data, aes(fill = murder)) l + geom_map(aes(map_id = state), map = map) + expand_limits(x = map\$long, y = map\$lat) map_id, alpha, color, fill, linetype, size</pre>
Three Variables <ul style="list-style-type: none"> Continuous <pre>seals\$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)) m <- ggplot(seals, aes(long, lat)) m + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE) x, y, alpha, fill (fast)</pre> Discrete <pre>m + geom_contour(aes(z = z)) x, y, z, alpha, colour, linetype, size, weight</pre> Discrete <pre>m + geom_tile(aes(fill = z)) x, y, alpha, color, fill, linetype, size (slow)</pre> 	



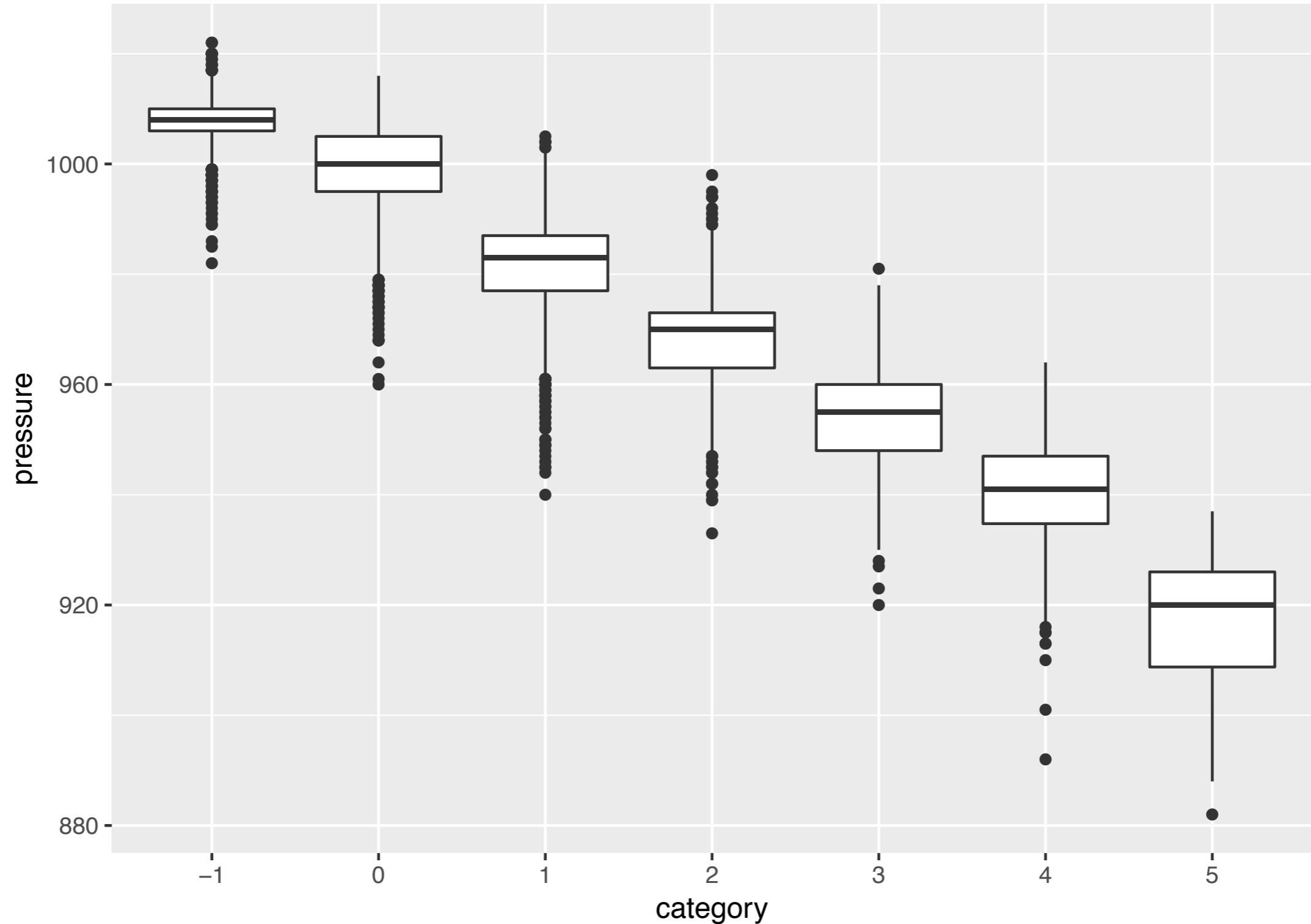
YOUR TURN

- Decide how to replace the scatterplot with boxplots.
Use the cheatsheet and try your best guess.

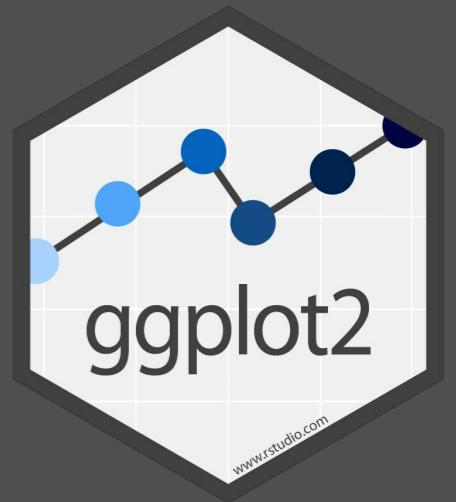
```
ggplot(storms) + geom_point(aes(x = category, y = pressure))
```



```
ggplot(storms) + geom_boxplot(aes(x = category, y = pressure))
```



YOUR TURN

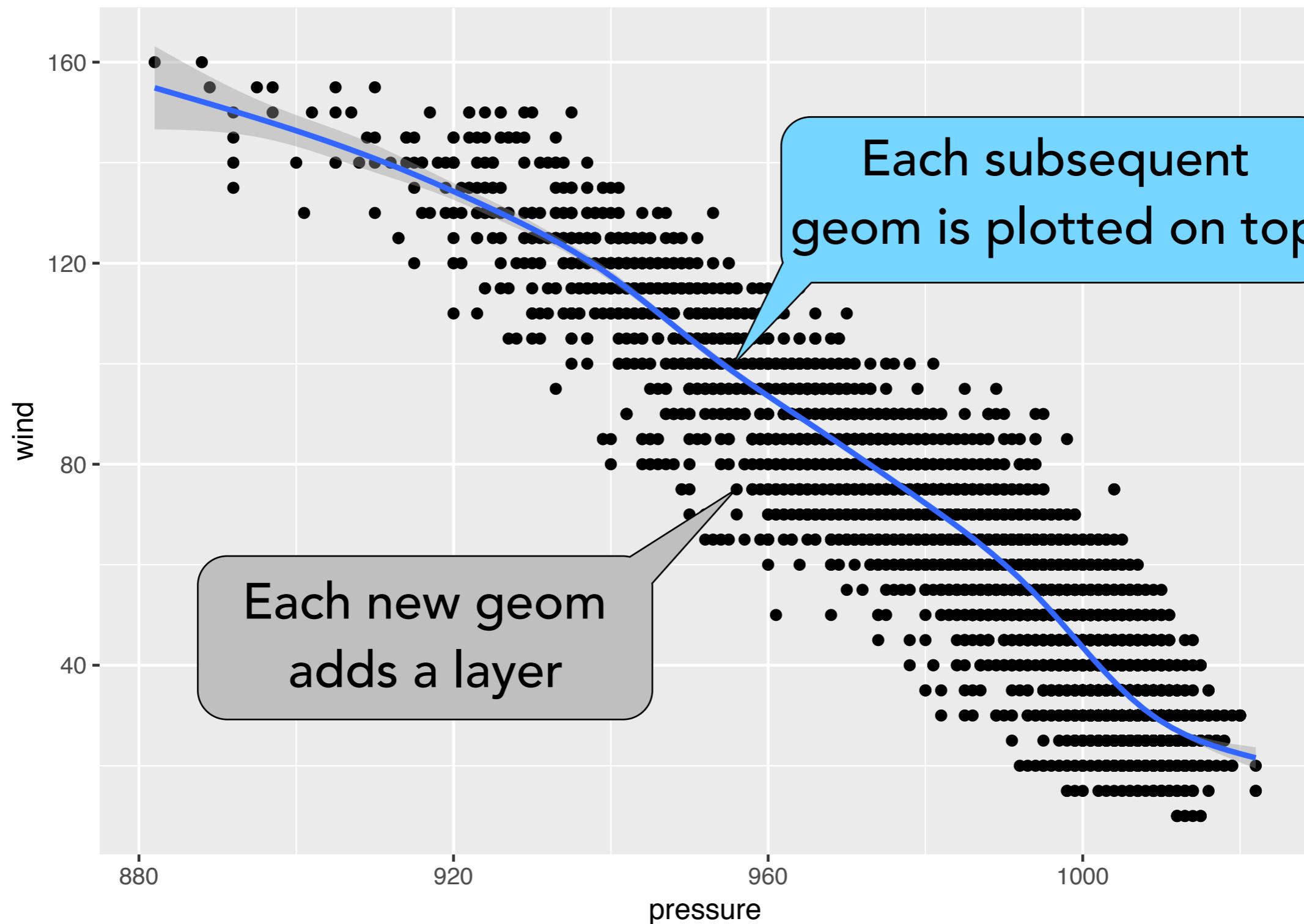


- Predict what this code will do, then run it.

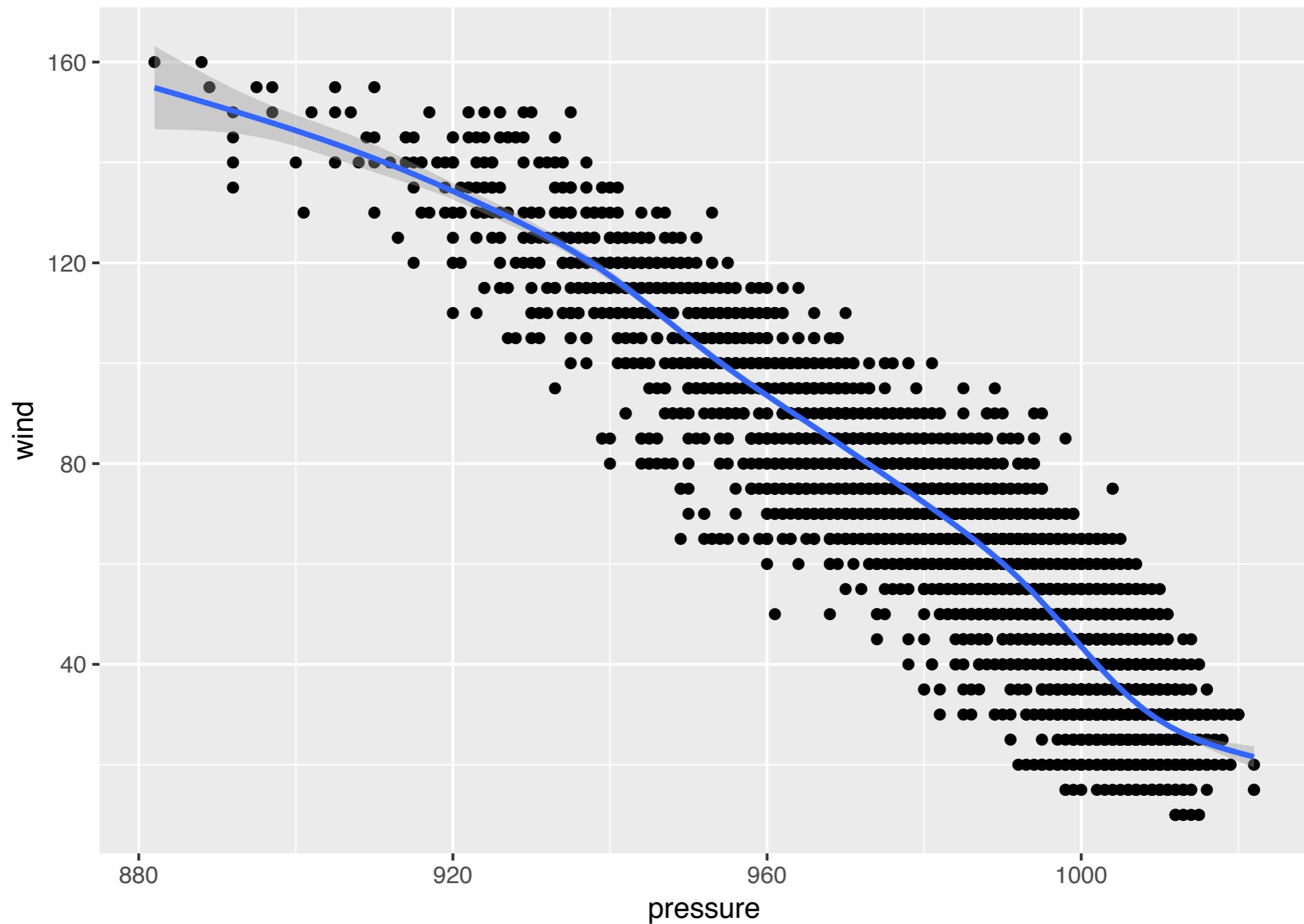
```
ggplot(storms) +  
  geom_point(aes(x = pressure, y = wind)) +  
  geom_smooth(aes(x = pressure, y = wind))
```



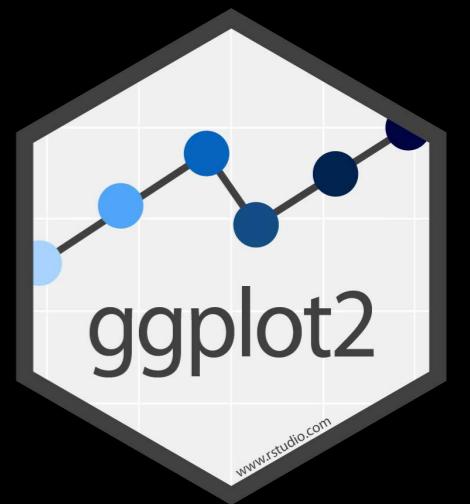
```
ggplot(storms) +  
  geom_point(aes(x = pressure, y = wind)) +  
  geom_smooth(aes(x = pressure, y = wind))
```



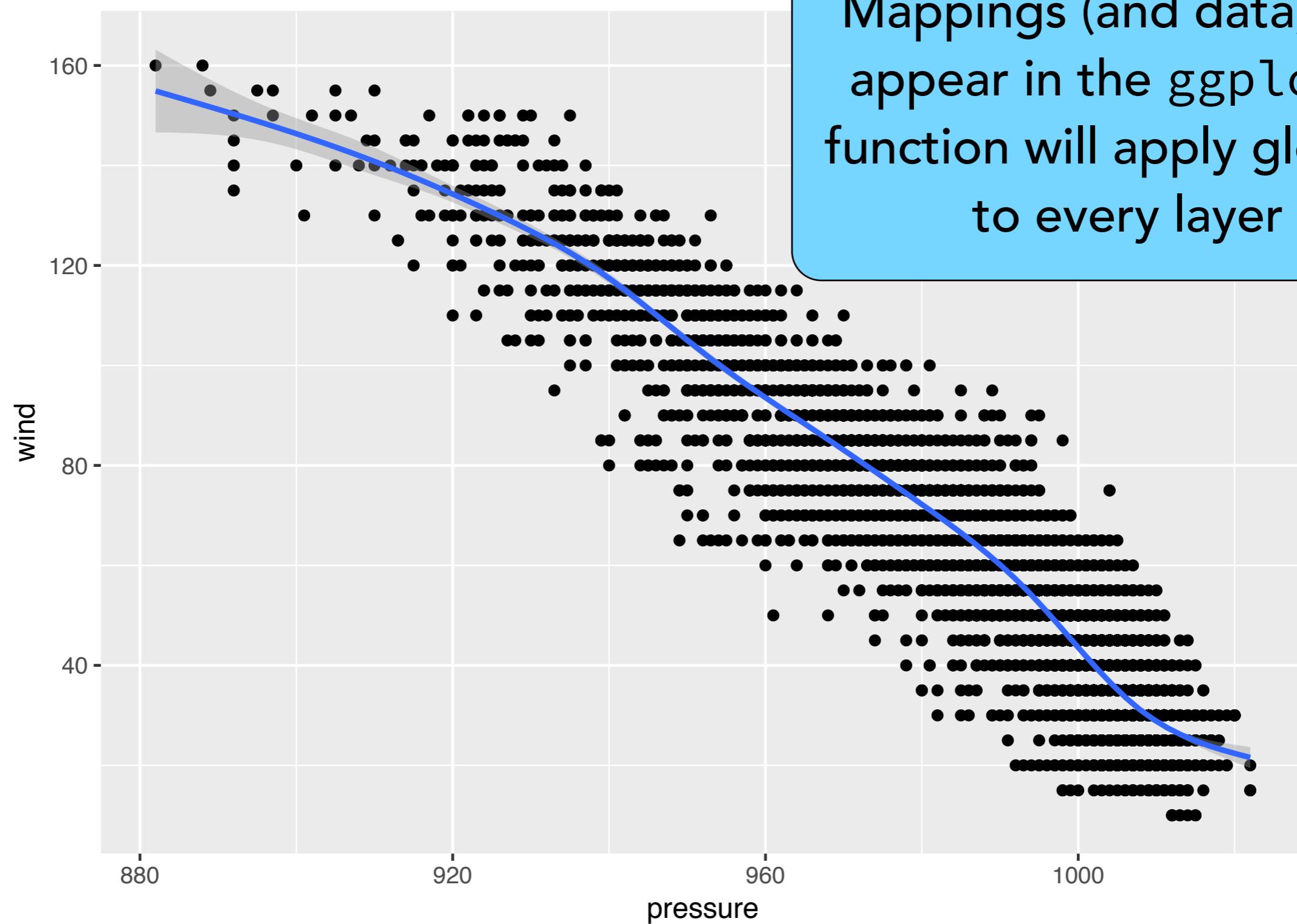
```
ggplot(storms) +  
  geom_point(aes(x = pressure, y = wind)) +  
  geom_smooth(aes(x = pressure, y = wind))
```



GLOBAL VS. LOCAL MAPPING

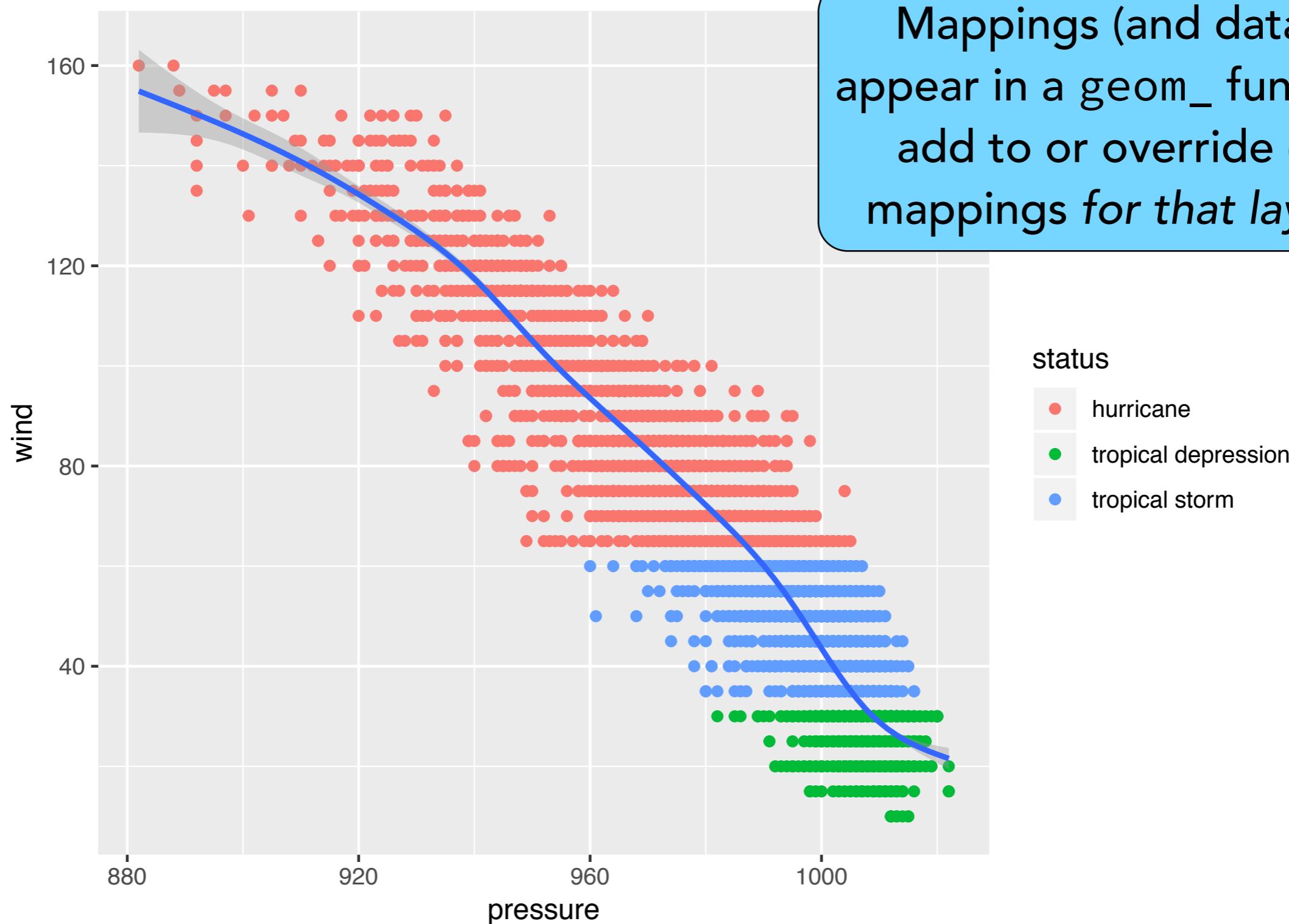


```
ggplot(storms, aes(x = pressure, y = wind)) +  
  geom_point() +  
  geom_smooth()
```



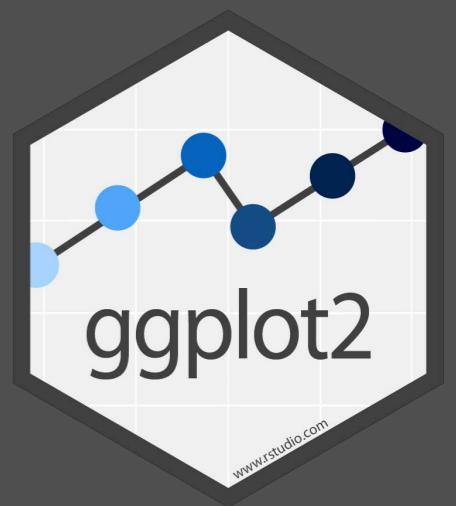
Mappings (and data) that appear in the `ggplot()` function will apply globally to every layer

```
ggplot(storms, aes(x = pressure, y = wind)) +  
  geom_point(aes(color = status)) +  
  geom_smooth()
```



Mappings (and data) that appear in a `geom_` function will add to or override global mappings for that layer only

YOUR TURN

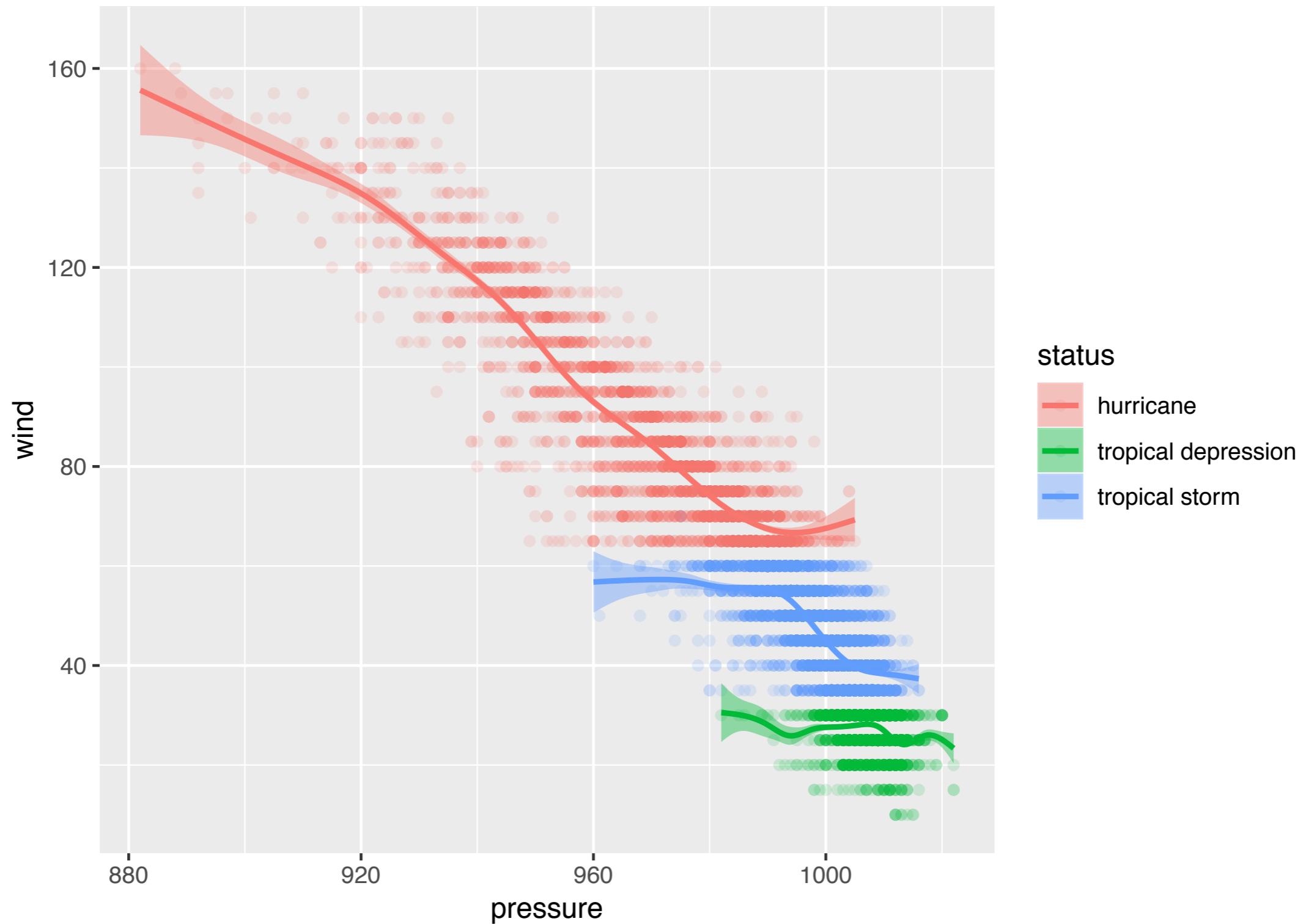


- Predict what this code will do, then run it.

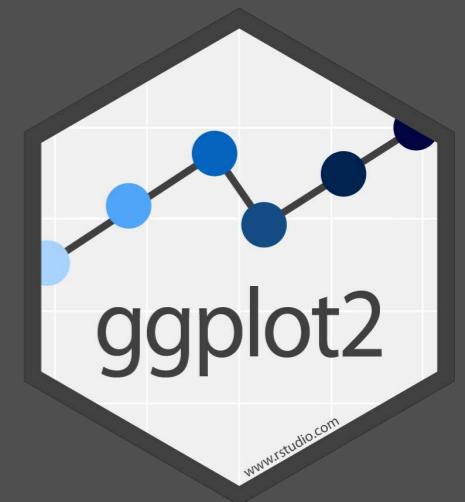
```
ggplot(storms, aes(x = pressure, y = wind,  
                    color = status, fill = status)) +  
  geom_point(alpha = 0.15) +  
  geom_smooth()
```



```
ggplot(storms, aes(x = pressure, y = wind,  
                    color = status, fill = status)) +  
  geom_point(alpha = 0.15) +  
  geom_smooth()
```



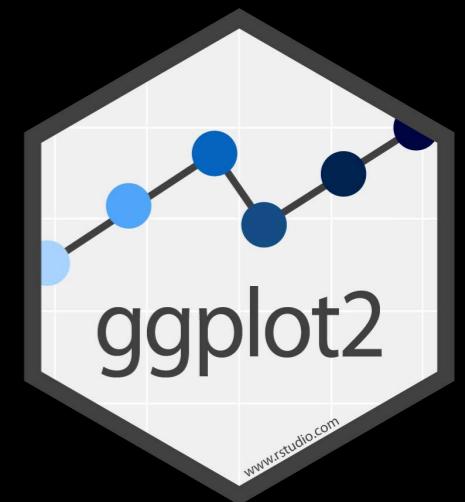
YOUR TURN



- Go to this week's assignments on the course website.
- Download the R Markdown file (save it in this week's folder in your class activities R project).
- Open the R Markdown file in R Studio.
- Follow the instructions to visualize a dataset about mammalian sleep.



WHAT ELSE?



- Stats
- Position adjustments
- Coordinates
- Facets
- Scales
- Themes

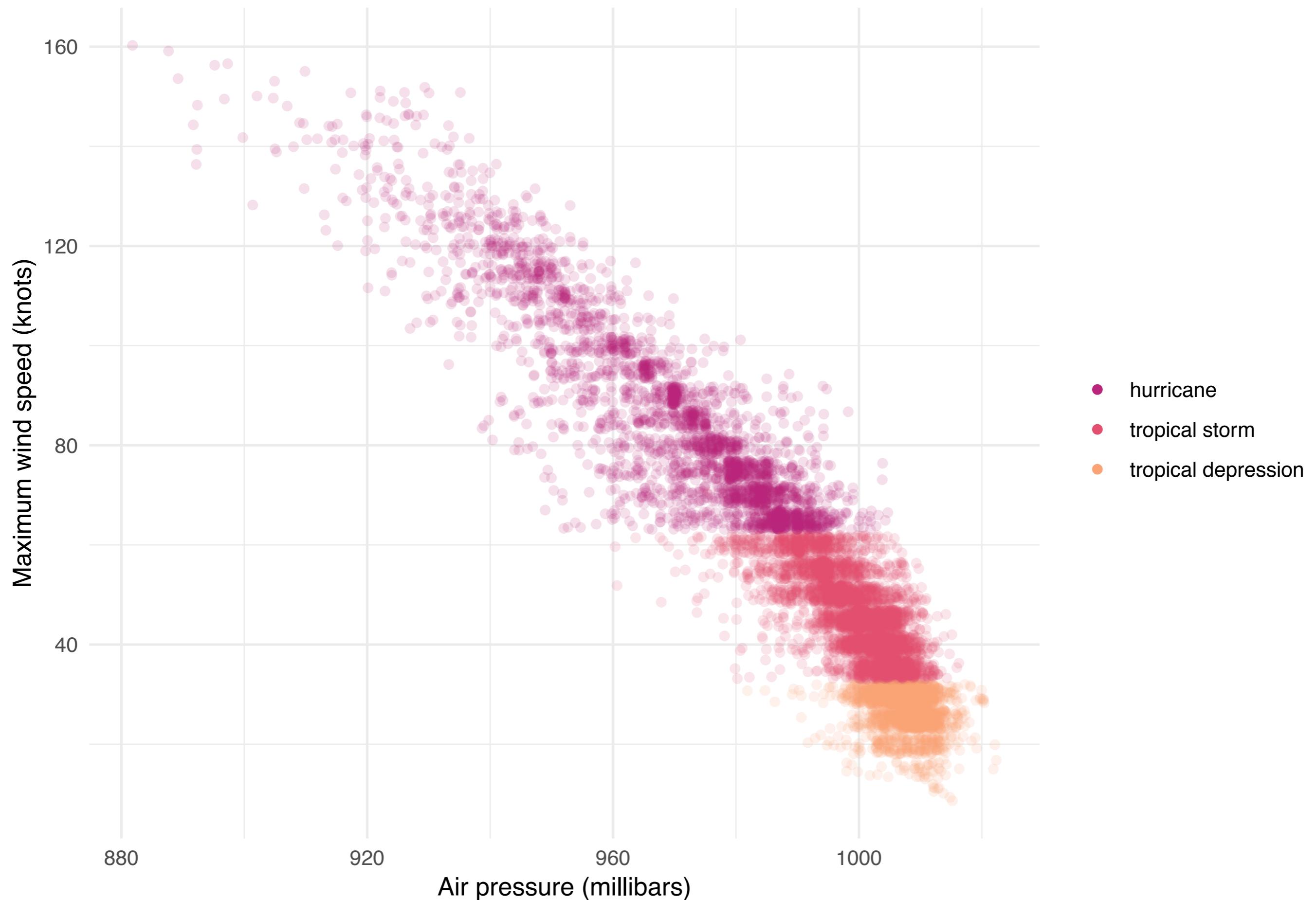
```
ggplot(data = <DATA>) +  
  <GEOM_FUNCTION>(  
    mapping = aes(<MAPPINGS>),  
    stat = <STAT>,  
    position = <POSITION>  
  ) +  
  <COORDINATE_FUNCTION> +  
  <FACET_FUNCTION> +  
  <SCALE_FUNCTION> +  
  <THEME_FUNCTION>
```

Required

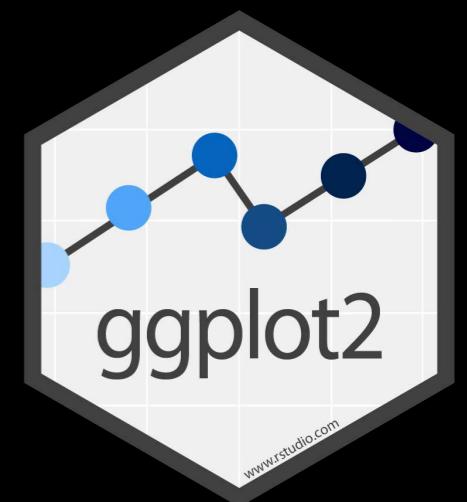
Not required,
sensible
defaults
supplied

Storms data

Status is determined by maximum wind speed



WHERE TO GET HELP



- Cheatsheet for basics
- <https://ggplot2.tidyverse.org/reference/> for very thorough detail, including examples.
- Credit: some ideas, examples, and figures from an [RStudio webinar](#), which is licensed CC by SA.