

R you ready?

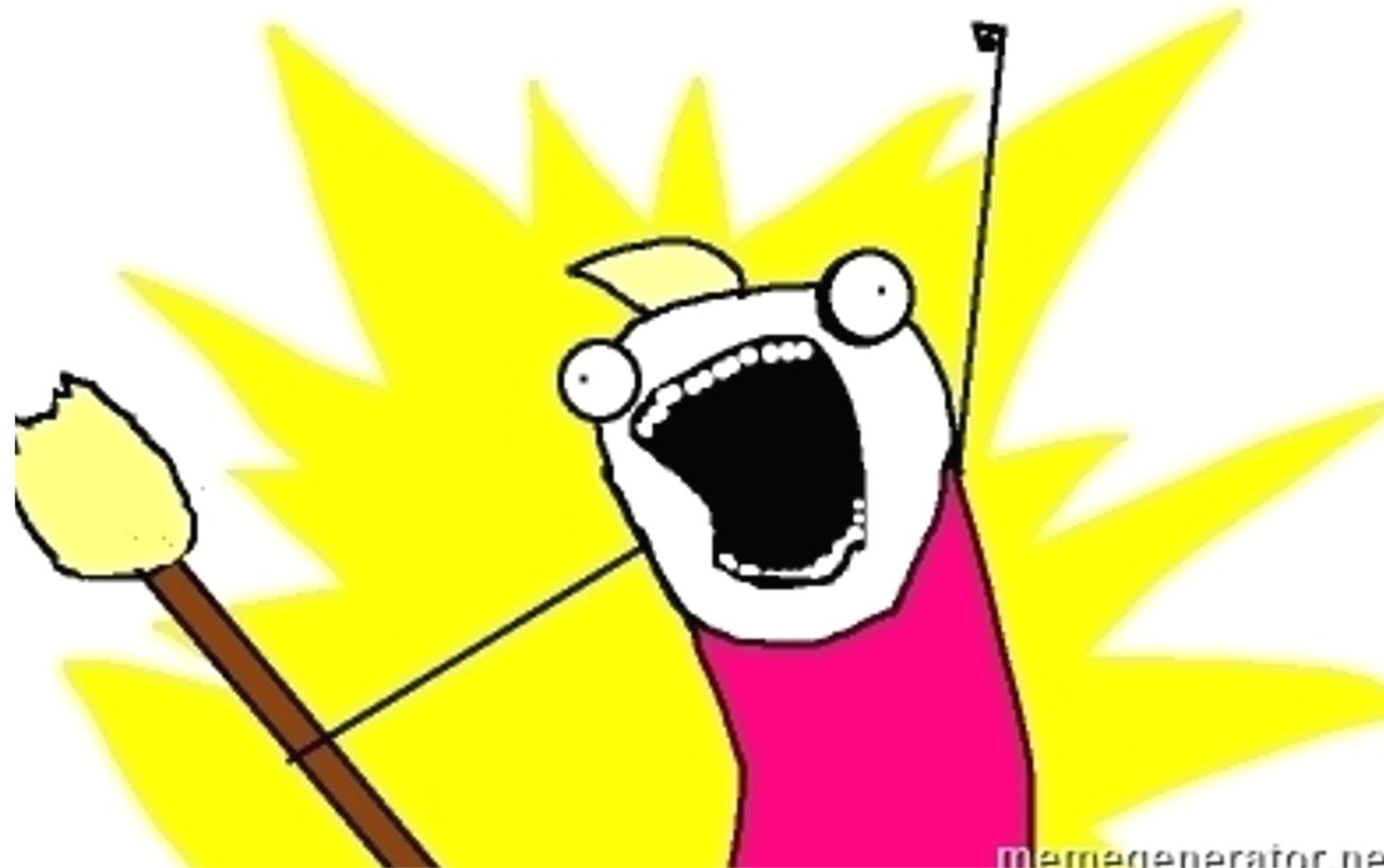
**IntRo to RStudio and R Markdown
for open data and reproducibility**

**Unit 7:
Let's get plotting!**

Mason A. Wirtz



GGPLOT ALL THE THINGS



memegenerator.net

ggplot2: What is it?

ggplot2 is a package from the *tidyverse* and is one of the most-used plotting packages in R

ggplot2 plotting system →

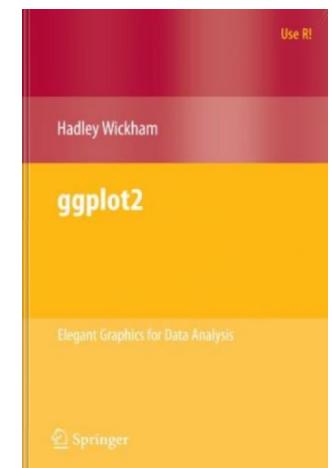
<https://bookdown.org/rdpeng/exdata/the-ggplot2-plotting-system-part-1.html#ggplot2-hello-world>

ggplot2: Elegant Graphics for Data Analysis →

<https://1lib.at/book/704124/daea5e>

ggplot2 cheatsheet →

<https://www.rstudio.com/resources/cheatsheets/>



ggplot2: Why?

- Set of independent components that can be composed in different ways
- Not limited to pre-specified graphics
- Plots can be built iteratively
- i.e. plots are built using a layering principle
- beautiful facetting or multipanel plots
- can easily change or update plots

ggplot2: Terminology

- **Aesthetic mappings (aes())** → describe how variables in the data are mapped to aesthetic attributes
- **Geometric objects (geom)** → represents what we actually see on the plot
- **Statistical transformations (stats)** → summarise data in many useful ways
- **Scales** → map values in the data space to values in an aesthetic space
- **Faceting** → break up the data into subsets

ggplot2: Introduction

The workhorse function:

- `ggplot()`



ONE variable:

`data frame %>%`

`ggplot(aes(x = variable_x_axis))`

Takes the arguments:

- `data = data frame`
- `mapping = aes()`

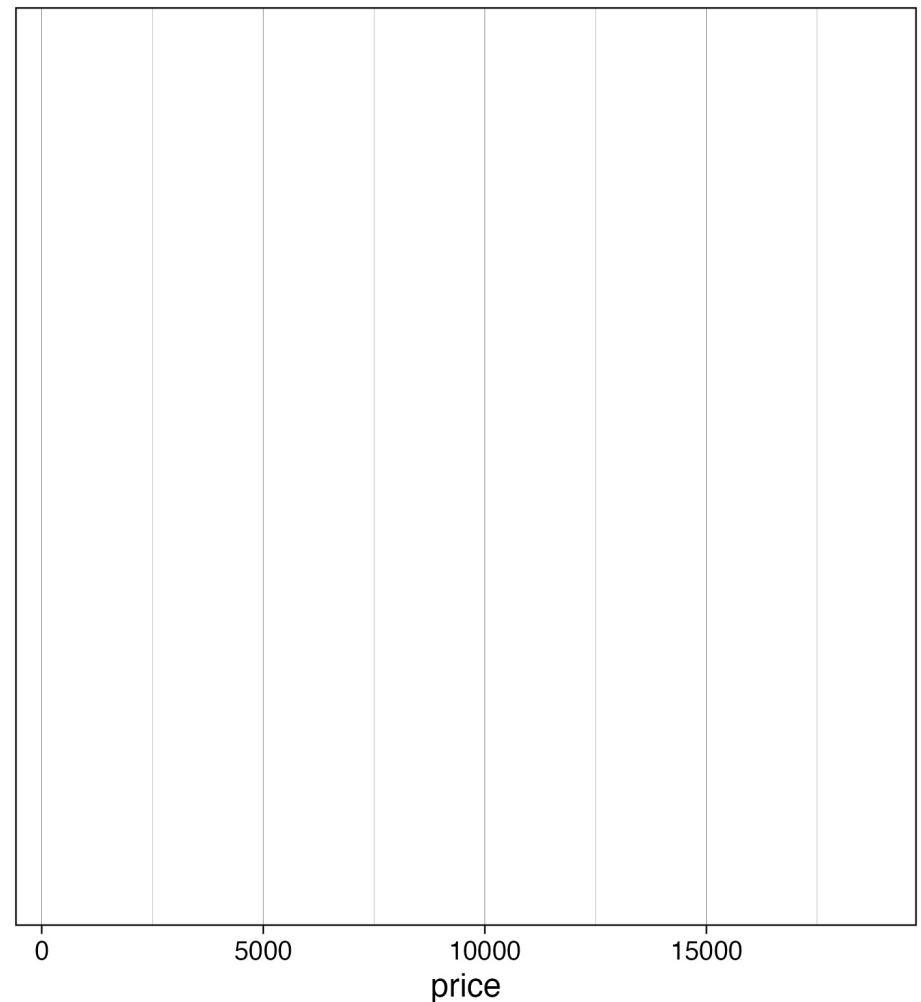
ggplot2: Introduction

Data frame

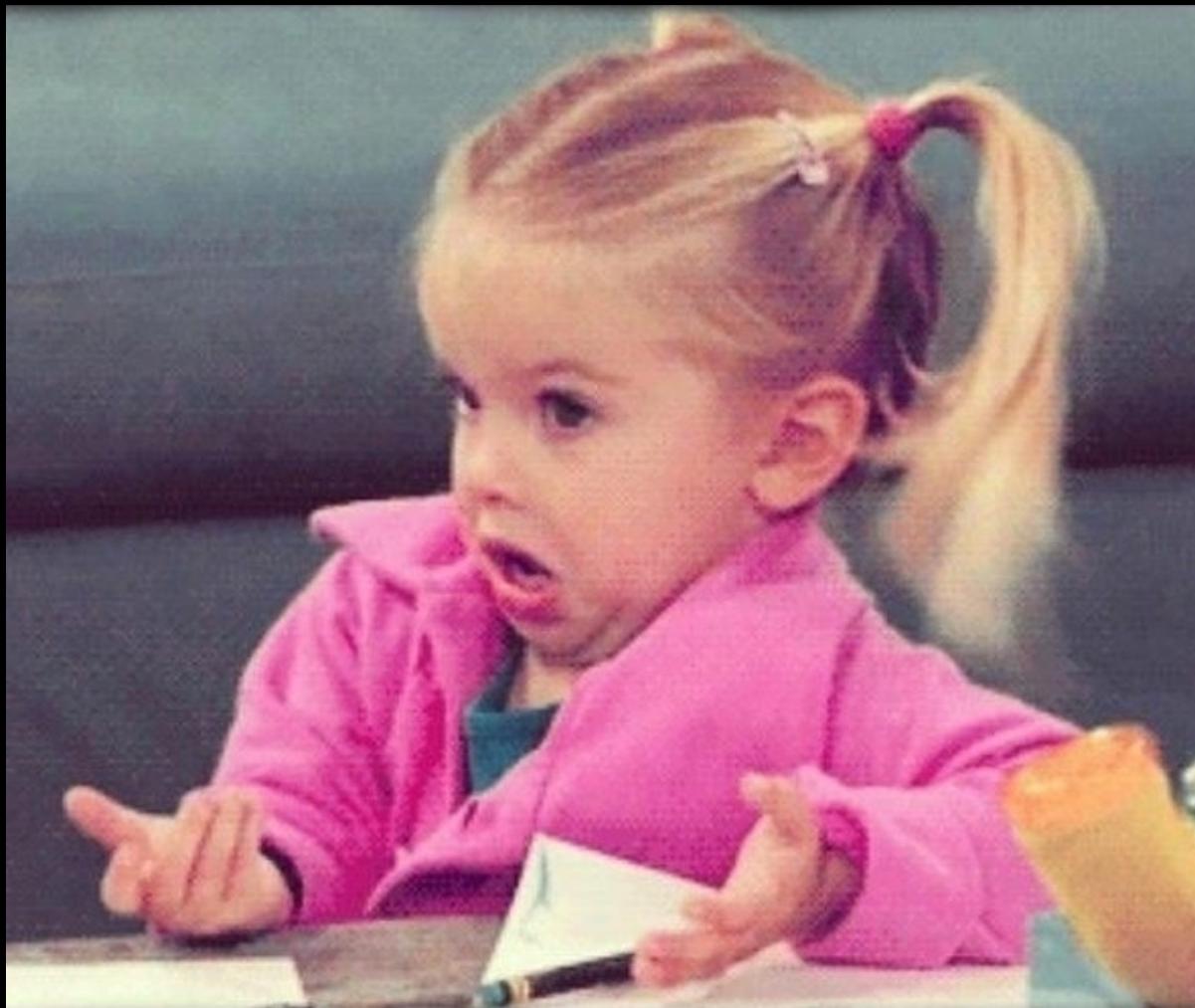
```
diamonds %>%  
  ggplot(aes(x = price))
```

Add aesthetics

We defined our x-axis,
but we have not PLOTTED
anything!



So how do we plot?



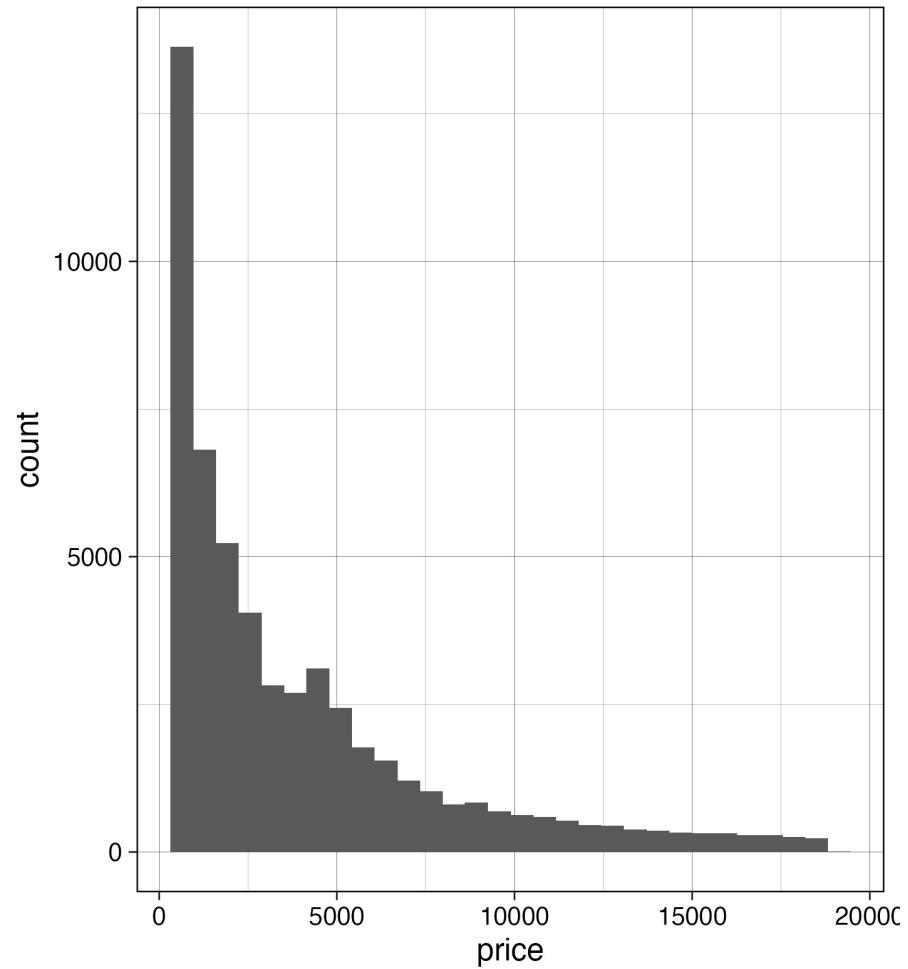
ggplot2: Plots of one variable

One CONTINUOUS variable:

- **Histogram:** *geom_histogram()*
- **Density plot:** *geom_density()*

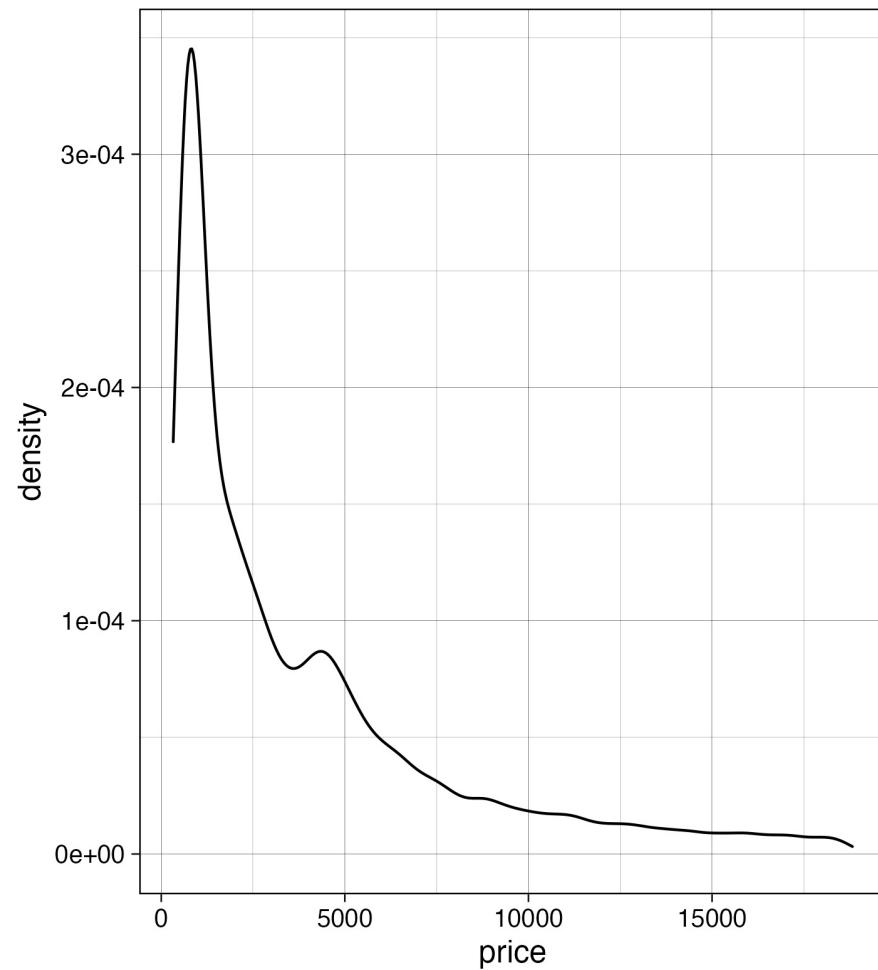
ggplot2: geom_histogram()

```
iamonds %>%  
  ggplot(aes(x = price)) +  
    geom_histogram()
```



ggplot2: geom_density()

```
iamonds %>%  
  ggplot(aes(x = price)) +  
    geom_density()
```



ggplot2: Plots of one variable

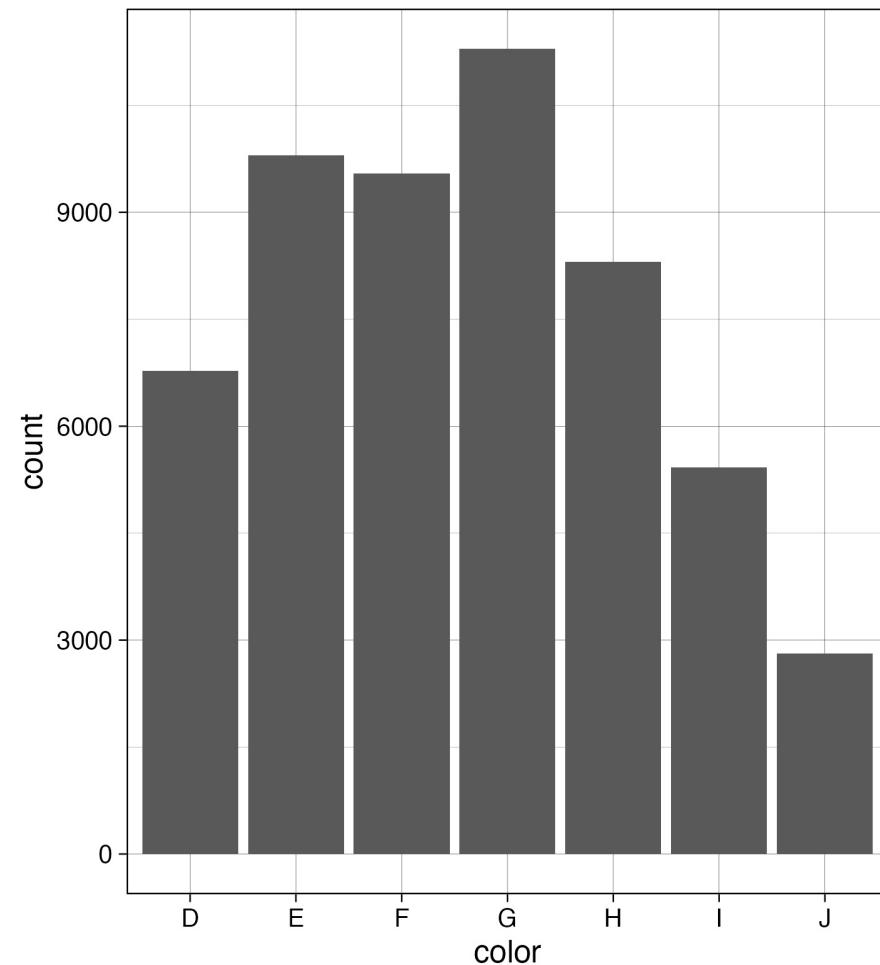
One CATEGORICAL variable:

- **Barplot:** *geom_bar()*

ggplot2: geom_density()

diamonds %>%

```
ggplot(aes(x = color)) +  
  geom_bar()
```



ggplot2: Plots of two variables

Two CONTINUOUS variables:

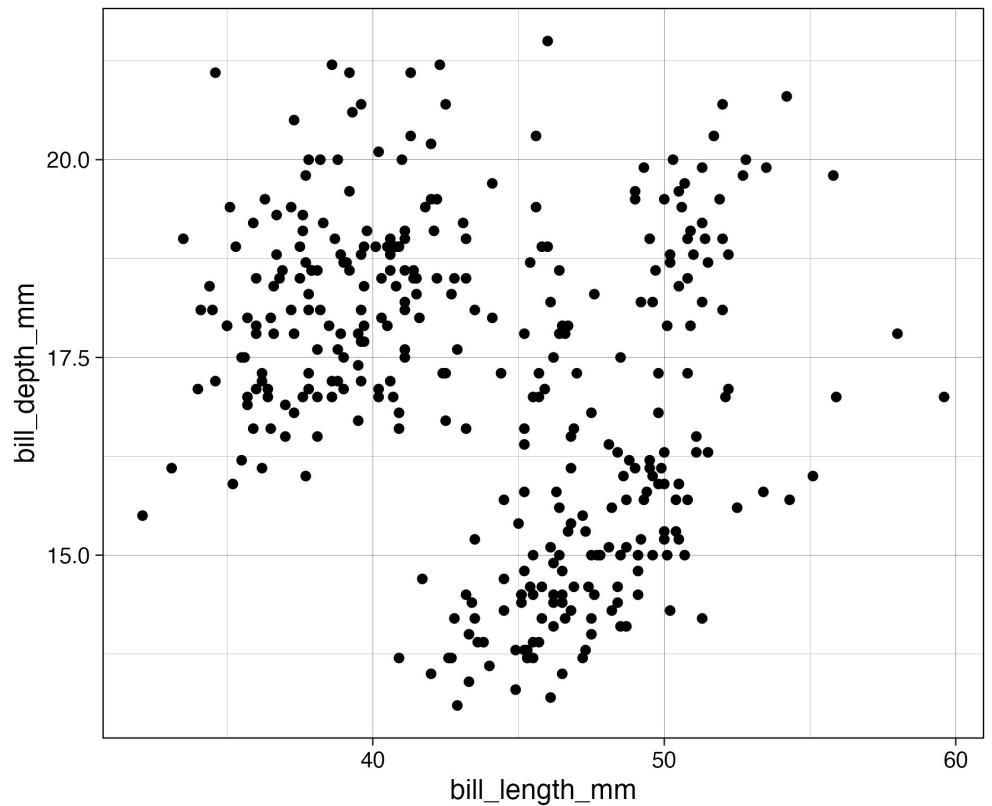
- **Scatter plot:** *geom_point()*
- **Scatter plot:** *geom_jitter()* → jittered (scattered) data points
- **Smoother:** *geom_smooth()* → linear and nonlinear lines

ggplot2: geom_density()

```
penguins %>%
```

```
  ggplot(aes(x = bill_length_mm,  
             y = bill_depth_mm)) +
```

```
    geom_point()
```



ggplot2: geom_smooth()

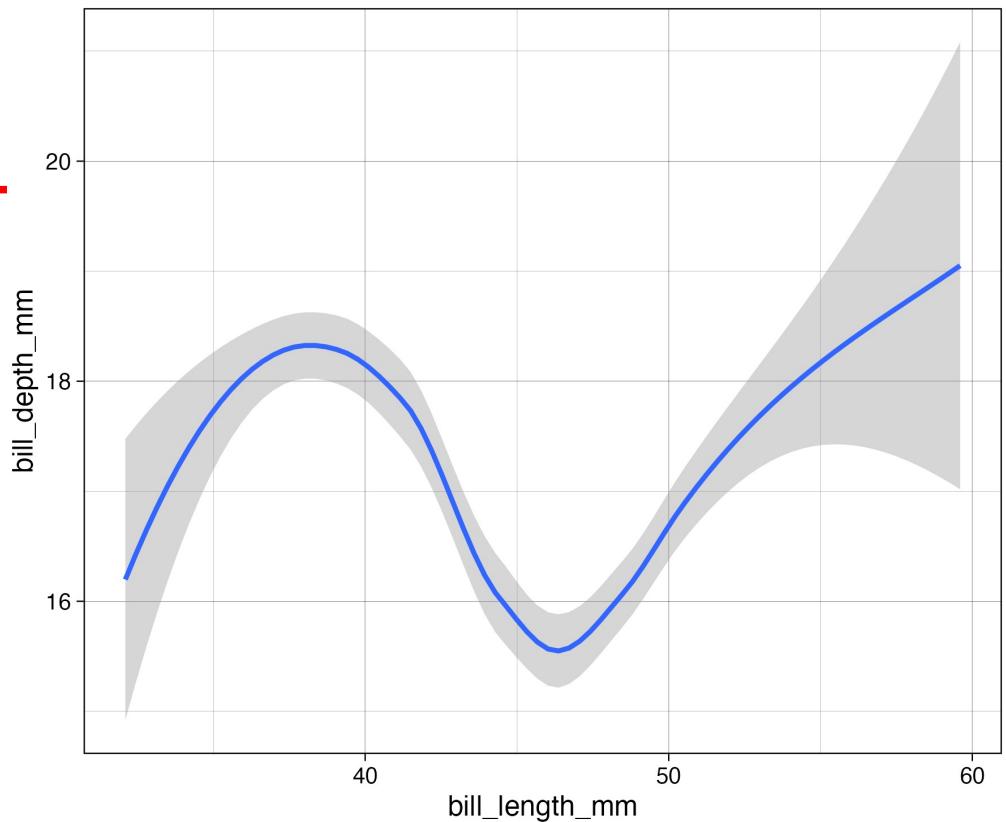
```
penguins %>%
```

```
  ggplot(aes(x = bill_length_mm,  
             y = bill_depth_mm)) +
```

```
    geom_smooth()
```

Default method:
Generalized additive model
(GAM) with cubic spline

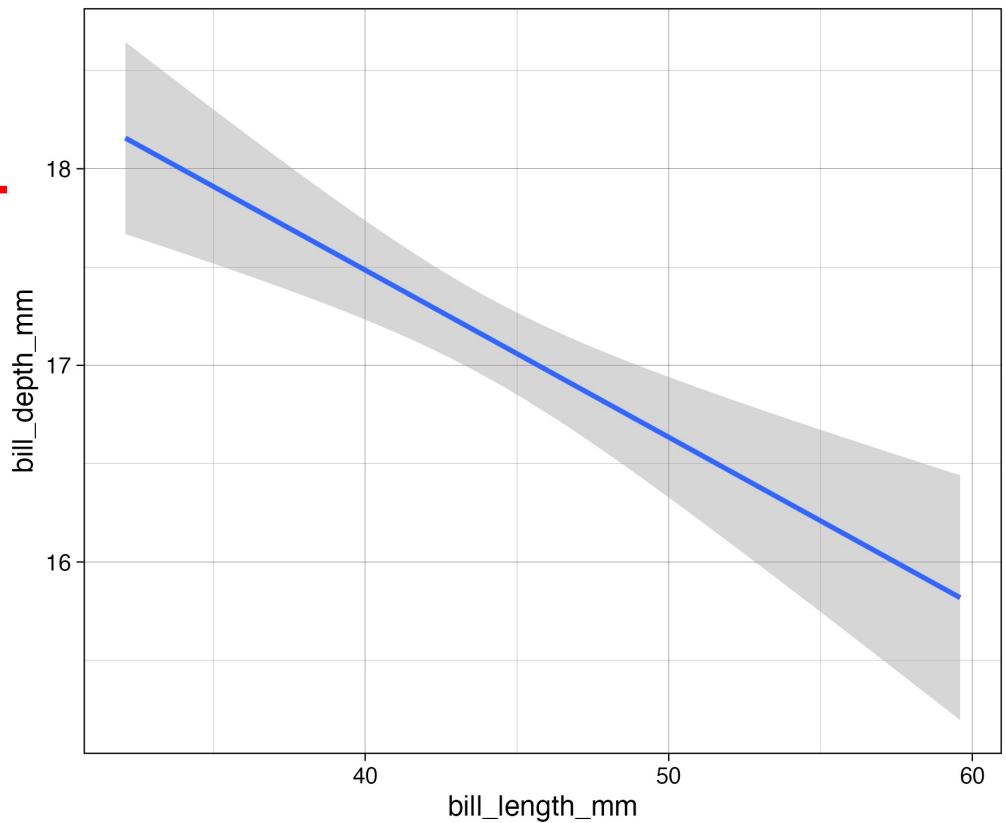
$y \sim s(x, bs = "cs")$



ggplot2: geom_smooth()

```
penguins %>%  
  ggplot(aes(x = bill_length_mm,  
             y = bill_depth_mm)) +  
  geom_smooth(method = "lm")
```

Change method:
method = “lm”
→ Linear model



ggplot2: Plots of two variables

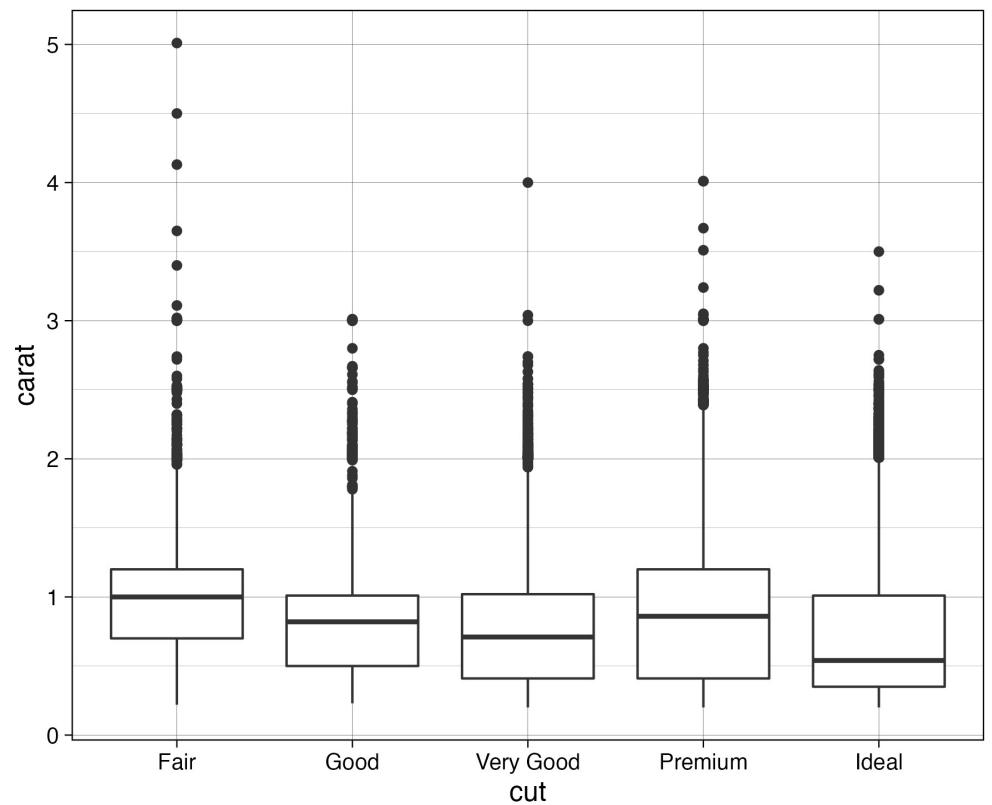
CATEGORICAL x and CONTINUOUS y:

- **Boxplot:** *geom_boxplot()*
- **Barplot:** *geom_bar(stat = "identity")*
- **Violin plot:** *geom_violin()*

ggplot2: geom_boxplot()

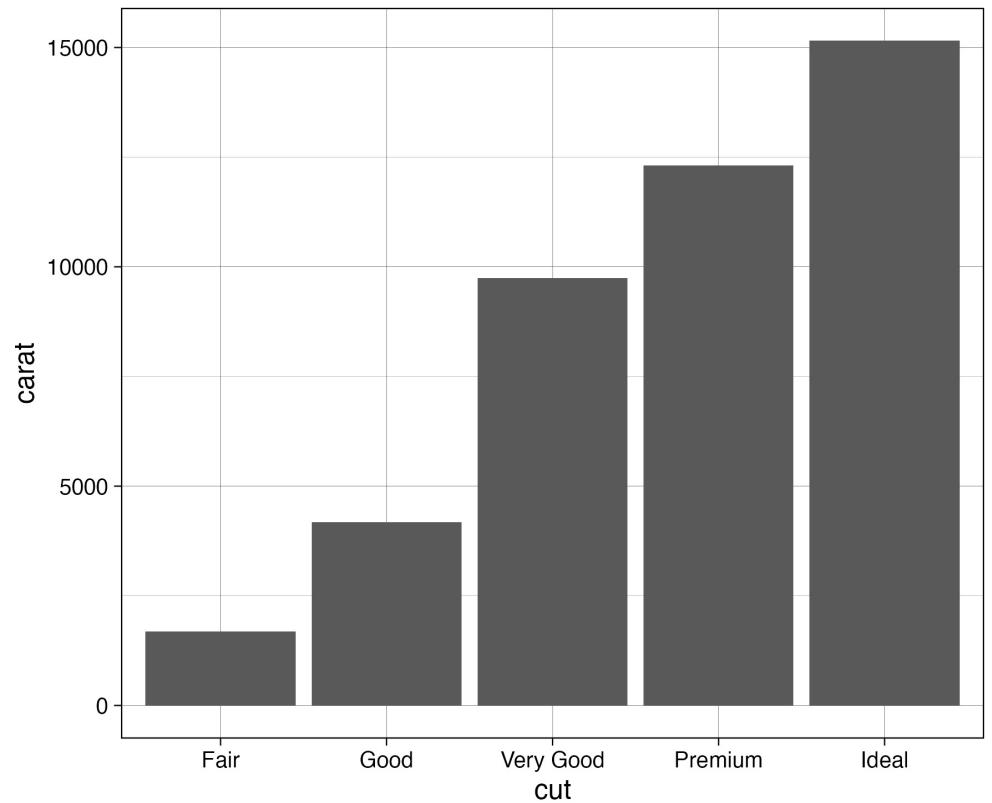
diamonds %>%

```
ggplot(aes(x = cut,  
           y = carat)) +  
  geom_boxplot()
```



ggplot2: geom_bar()

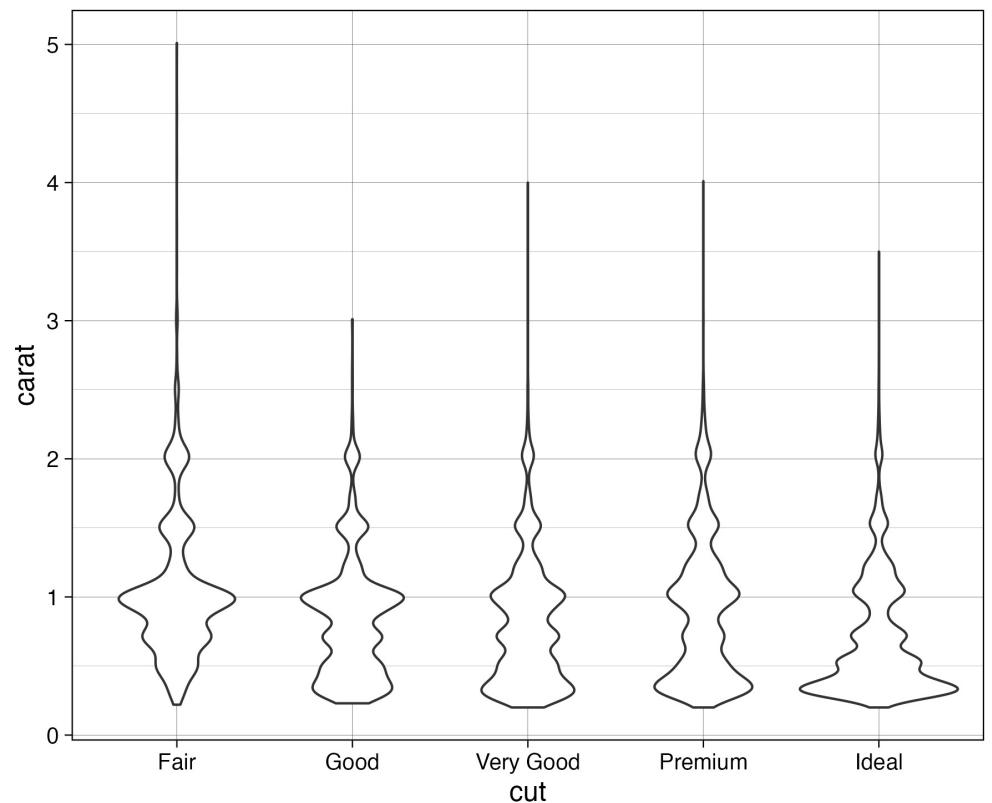
```
iamonds %>%  
  ggplot(aes(x = cut,  
             y = carat)) +  
  geom_bar(stat = "identity")
```



ggplot2: geom_bar()

```
diamonds %>%
```

```
  ggplot(aes(x = cut,  
             y = carat)) +  
    geom_violin()
```



AESTHETICS



ggplot2: Aesthetics

Aesthetics:

- **Size**

Change the size of e.g. data points

- **Shape**

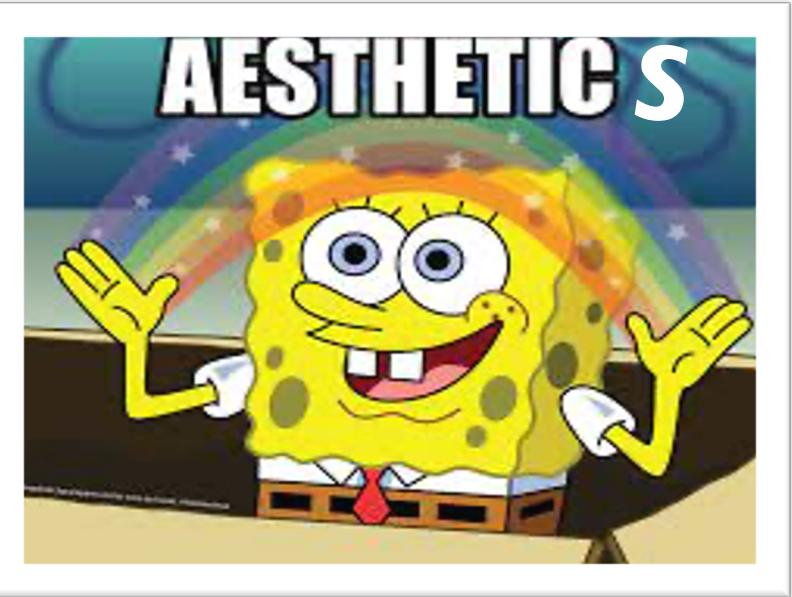
Change the shape of e.g. data points

- **Color**

Change the color of e.g. data points

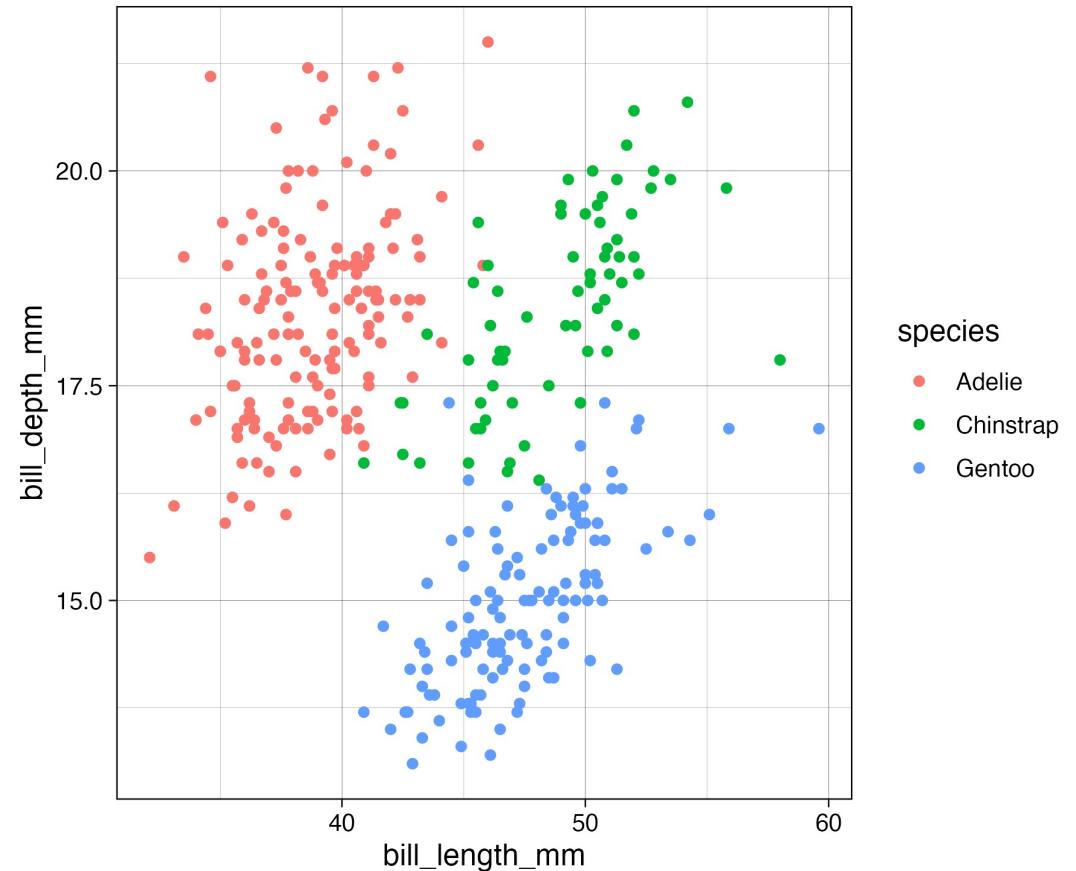
- **Fill**

Change the fill of geom, e.g. a boxplot



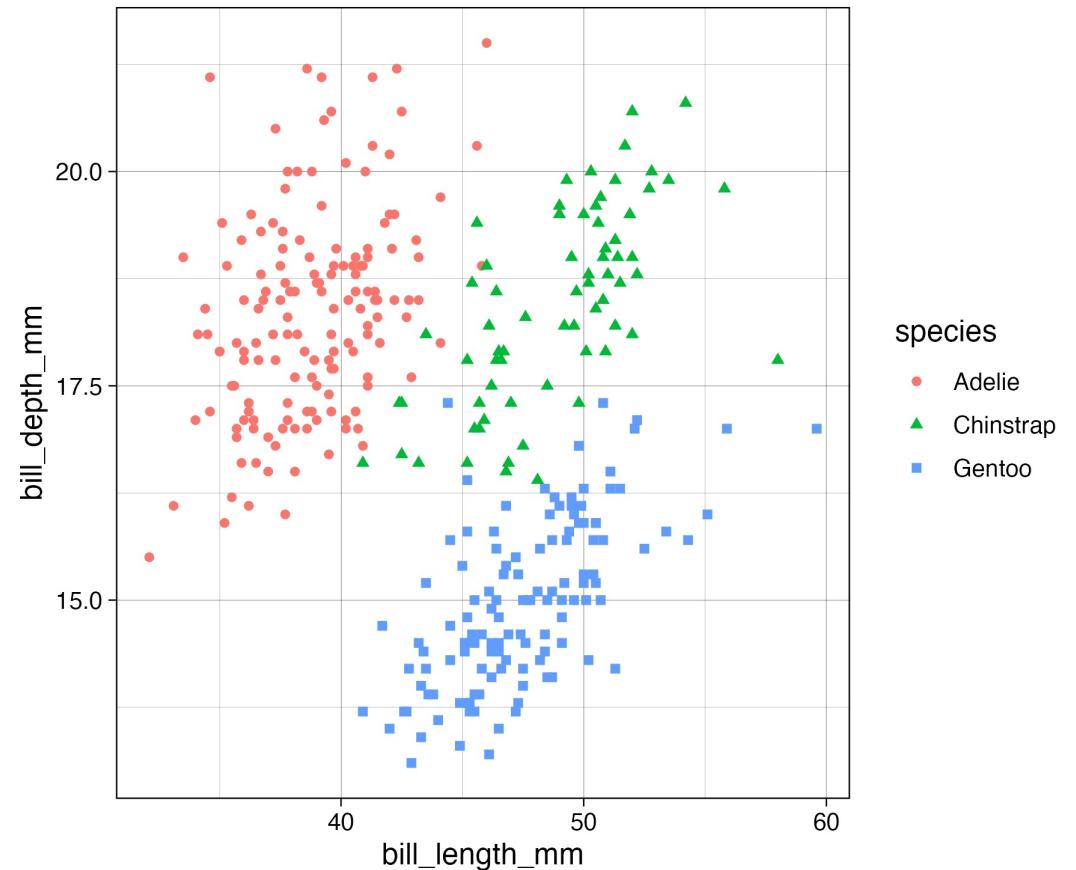
ggplot2: geom_point() + color

```
penguins %>%  
  ggplot(aes(x = bill_length_mm,  
             y = bill_depth_mm,  
             color = species)) +  
  geom_point()
```



ggplot2: geom_point() + color + shape

```
penguins %>%  
  ggplot(aes(x = bill_length_mm,  
             y = bill_depth_mm,  
             color = species,  
             shape = species)) +  
  geom_point()
```

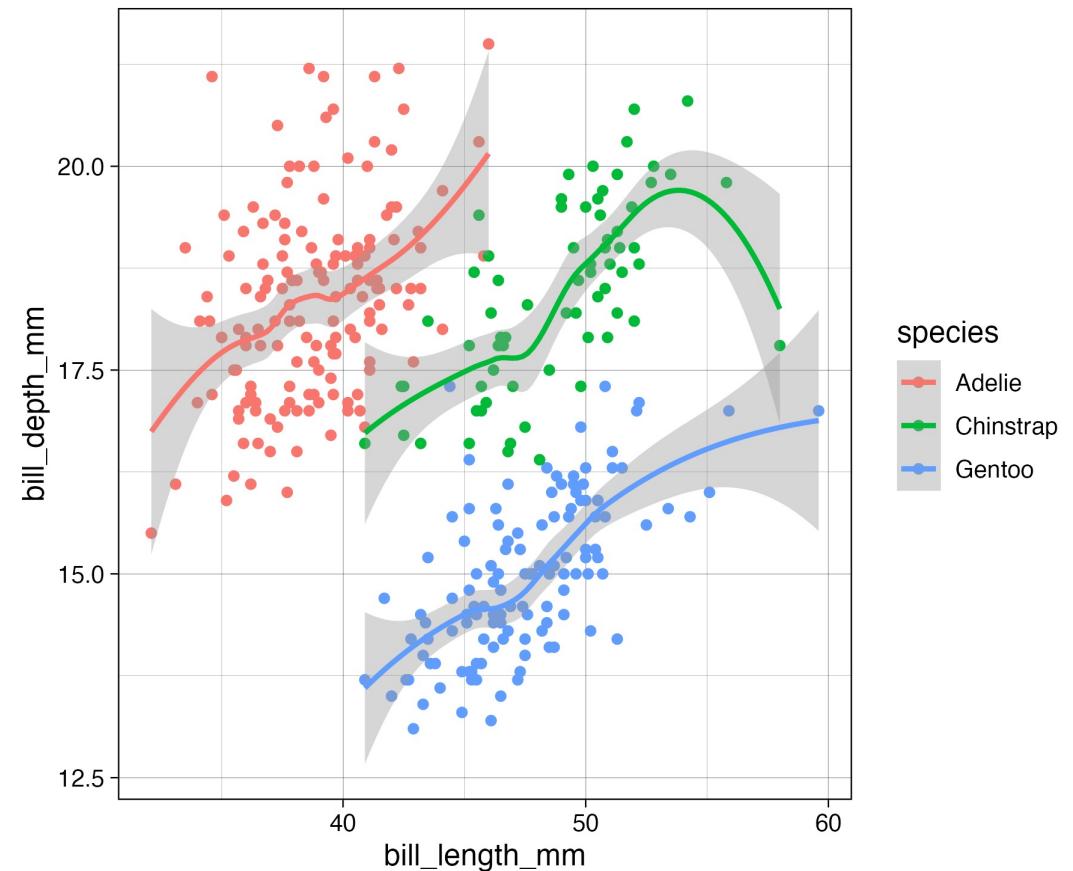


ggplot2: geom_point() + color + geom_smooth

penguins %>%

```
ggplot(aes(x = bill_length_mm,  
           y = bill_depth_mm,  
           color = species)) +
```

```
geom_point() +  
geom_smooth()
```



ggplot2: geom_point() + color + geom_smooth() + geom_smooth()

penguins %>%

```
ggplot(aes(x = bill_length_mm,  
          y = bill_depth_mm,  
          color = species)) +
```

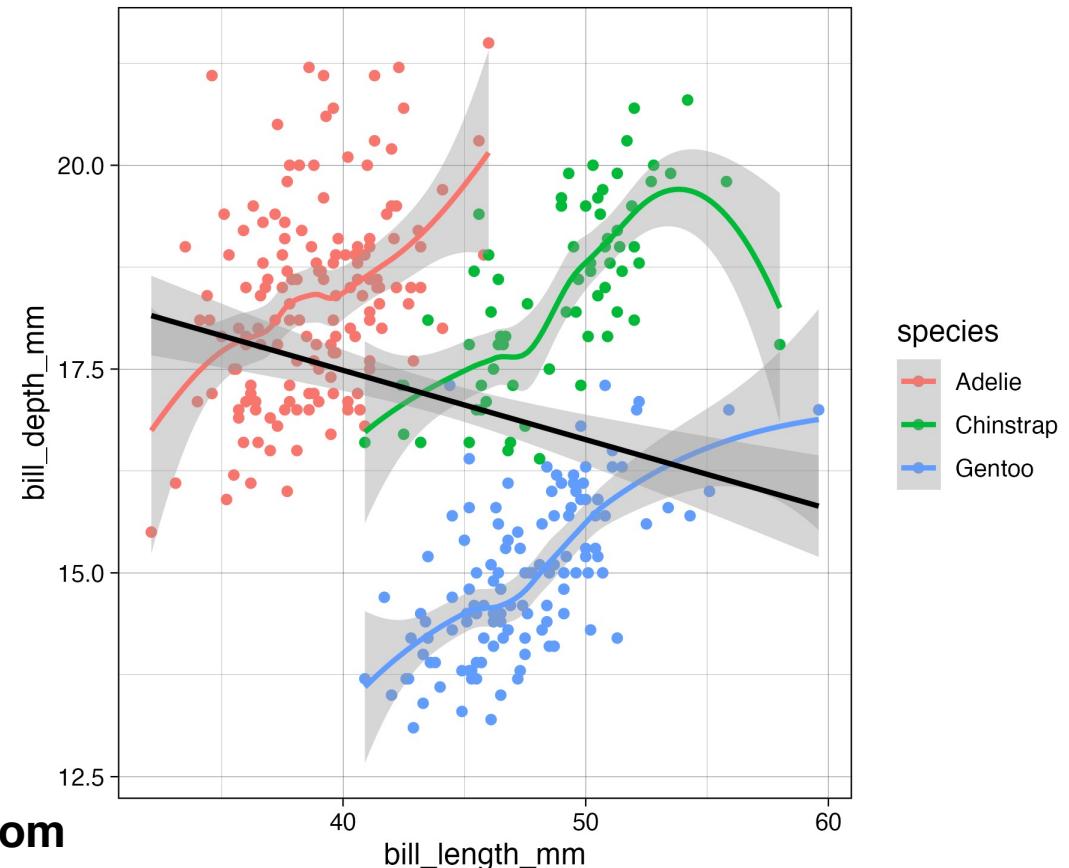
```
geom_point() +
```

```
geom_smooth() +
```

```
geom_smooth(method = "lm",  
            color = "black")
```



Overall between-species trend different from
individual species group trends;
i.e. *Simpson's paradox*



ggplot2: Where to put the arguments?

OVERALL aesthetics:

- Place in the `ggplot(aes(...))`

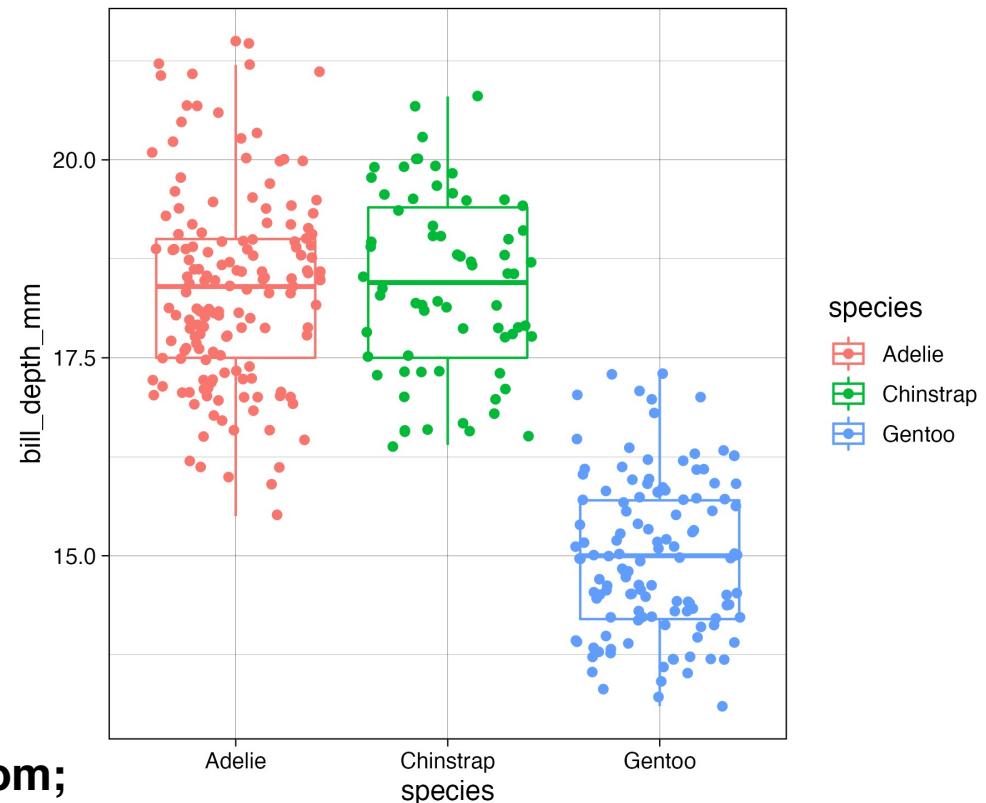
SPECIFIC aesthetics (i.e. only apply to ONE geom)

- Place in respective geom

ggplot2: geom_boxplot()

```
penguins %>%
```

```
  ggplot(aes(x = species,  
             y = bill_depth_mm,  
             color = species)) +  
  geom_boxplot() +  
  geom_jitter()
```

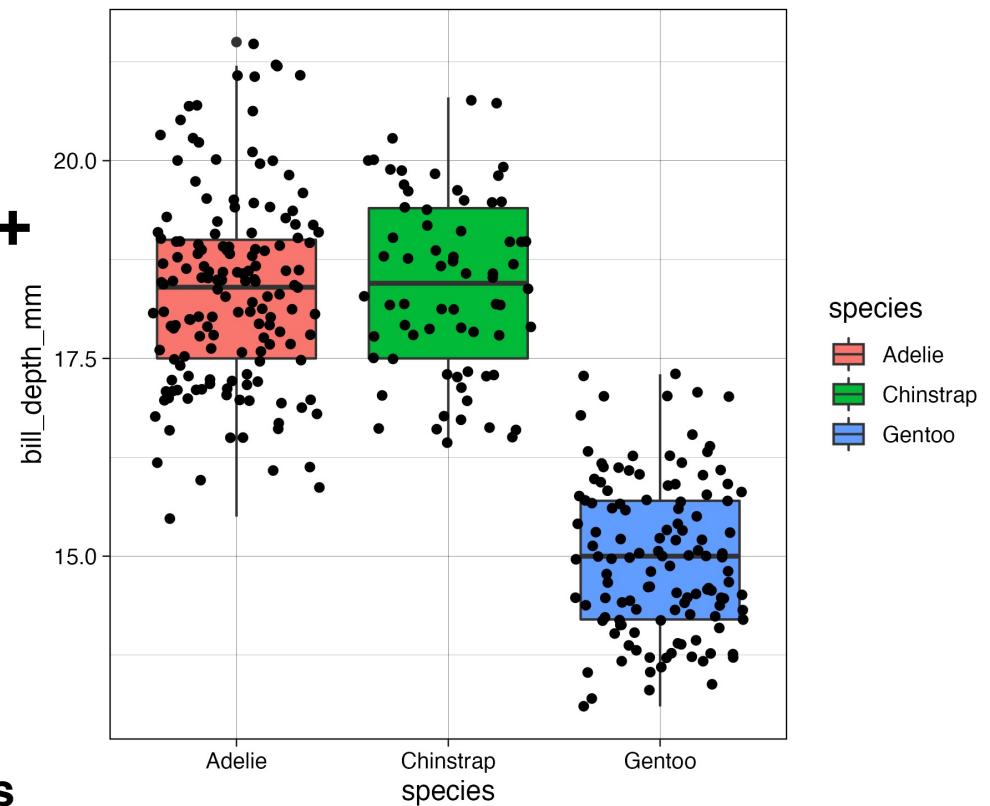


COLOR changes the OUTER color of a geom;
FILL changes how e.g. boxplot and violinplots
are filled

ggplot2: geom_boxplot()

```
penguins %>%
```

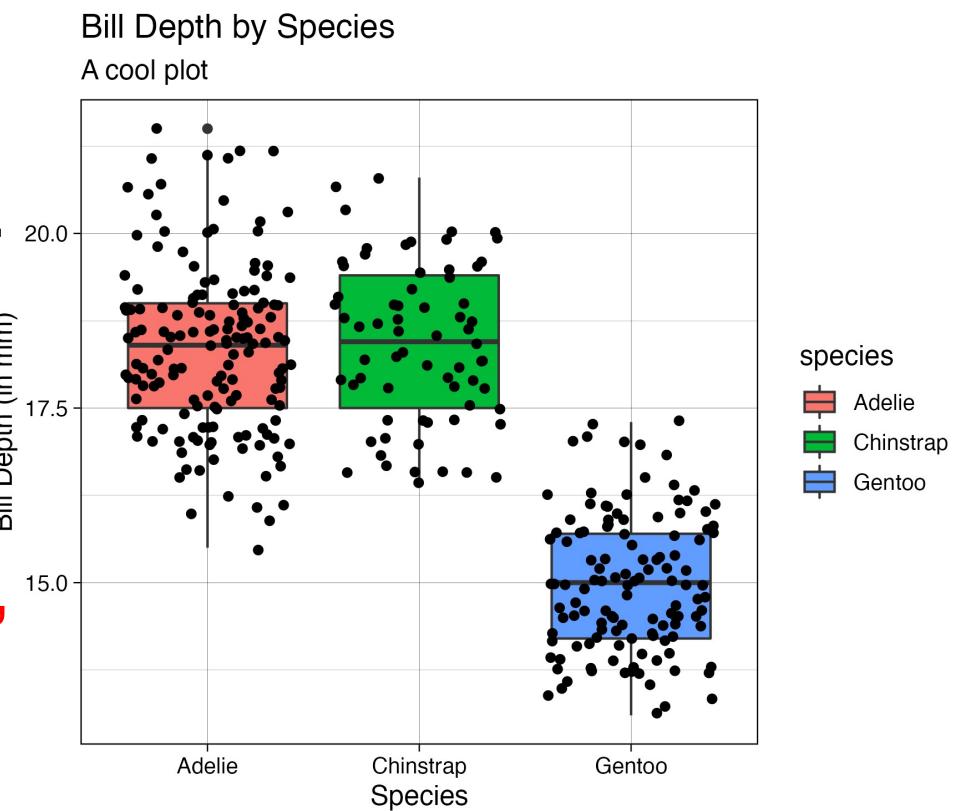
```
  ggplot(aes(x = species,  
             y = bill_depth_mm)) +  
    geom_boxplot(aes(fill = species)) +  
    geom_jitter()
```



COLOR changes the OUTER color of a geom;
FILL changes how e.g. boxplot and violinplots
are filled

ggplot2: Add plot labels via labs()

```
penguins %>%  
  ggplot(aes(x = species,  
             y = bill_depth_mm)) +  
  geom_boxplot(aes(fill = species)) +  
  geom_jitter() +  
  labs(x = "Species",  
       y = "Bill Depth (in mm)",  
       title = "Bill Depth by Species",  
       subtitle = "A cool plot")
```



WITH GREAT POWER COMES

A LOT OF CUSTOMIZATION

ggplot2: Saving ggplots

SAVE your plot as an object (just like usual, using the = or -> operator)

e.g.

```
billDepth = penguins %>%  
  ggplot(aes(x = species,  
             y = bill_depth_mm)) +  
  geom_boxplot(aes(fill = species)) +  
  geom_jitter() +  
  labs(x = "Species",  
       y = "Bill Depth (in mm)",  
       title = "Bill Depth by Species",  
       subtitle = "A cool plot")
```

```
ggsave(plot = ....,  
       filename = "RELATIVE PATH")
```

e.g.

```
ggsave(plot = billDepth,  
       filename = "./Figures/billDepth.png")
```



**DON'T FORGET TO SPECIFY
WHAT TYPE OF FILE YOU ARE
SAVING THE PLOT AS (.png, .pdf,
.eps, etc.)**

LET'S GET OUR HANDS DIRTY



makeameme.org