W3: Data Visualization

https://bit.ly/ir-w3

Announcements

- Data Snacks
  - Tips and Tricks with R and Python
  - Interactive Tutorials
- Commnunity Session time: 12-1:30 PM Wednesday
  - Compile a list of topics you want to see for Community Session
  - Invite Others (all are welcome): https://www.addevent.com/event/Dl23075333

Slido Link:
https://bit.ly/ir-w3

Last Week Today

(Sorry, John Oliver)

Clear Points

> implicit subsetting

Glad it was clear!

> today was pretty clear, i need to figure out the syntax for diving into specific parts of data frames but I think I need to just play with this.

Keep at it!

Clear Points

> How to make a vector, how to select values out from the vector, how to select values from a data frame, … lots of things!

> how to use vectors, what they are used for, and how to start thinking about manipulating data frames

Yay!

Muddy Points

> implicit subsetting

We will revisit this when we get to data wrangling in a couple weeks. Let it sit and we'll revisit it.

Muddy Points

> Can vectors be thought of as data columns or are they sometimes rows or neither?

In R, the columns of a `data.frame` are vectors. These three things are equivalent:

- Column
- Variable
- Vector

In general, rows are not vectors in a `data.frame`. This is because you can have mixed data types across a row.

Muddy Points

> When to use $

We use `$` to refer to a column within a `data.frame`. We mostly do it when we need to implicitly subset based on a criteria of that column. One of the joys of the `tidyverse` is that you don't have to use it.

Muddy Points

> Nothing muddy but i think since vectors are such a crucial concept to learn - live coding with a simpler data set would be easier. Sometimes the medical data can go over my head. […]

> Some of the in class exercises were difficult to follow because we could not see the initial vector definition before each operation was applied to it. This made it hard for me to understand […]

Thanks for the feedback. There is a preview of the data we're using today in a slide.

I have used toy datasets before.

Muddy Points

> Not necessarily unclear, but it would be helpful to do a quick rundown of any tips to remember the ordering for how to recall information and also the ordering of arguments for other functions.

For the most part, we've been using the order to distinguish our arguments from each other. You can mix the order up by using the argument names. These are all equivalent:

```r
seq(1,10,2)
seq(from=1, to=10, by=2)
seq(by=2, from=1, to=10)
```

When working with functions, auto-completion is your friend. Try using the key when you start working with a function.

Data Visualization

Penguins Dataset

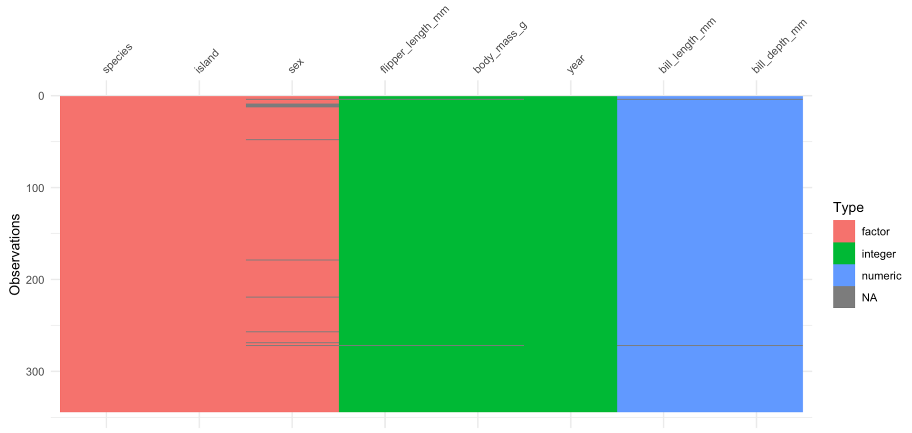
```r
gt::gt(head(penguins))
```

species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
Adelie	Torgersen	39.1	18.7	181	3750	male	2007
Adelie	Torgersen	39.5	17.4	186	3800	female	2007
Adelie	Torgersen	40.3	18.0	195	3250	female	2007
Adelie	Torgersen	NA	NA	NA	NA	NA	2007
Adelie	Torgersen	36.7	19.3	193	3450	female	2007
Adelie	Torgersen	39.3	20.6	190	3650	male	2007

- Note that our dataset has column names
- In `ggplot2`, we don't need to use the `$` operator: `penguins$species`
- We use the bare column name to refer to it: `species`
  - `bill_depth_mm`:`numeric`
  - `bill_length_mm`:`numeric`
  - `species`:`character`

{visdat} for Exploratory Data Analysis

```r
library(visdat)
vis_dat(penguins)
```



https://bit.ly/ir-w3

Common Plots

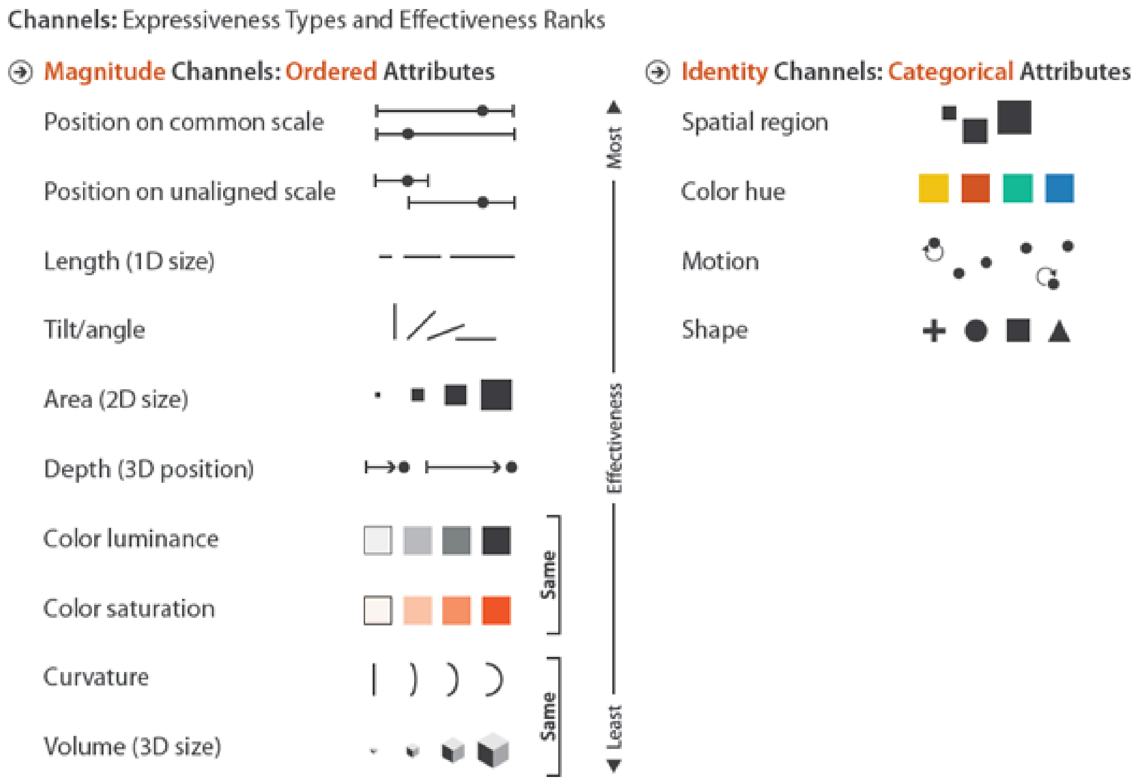
One Variable

- Numeric: histogram
- Character: bar plots

Two Variables

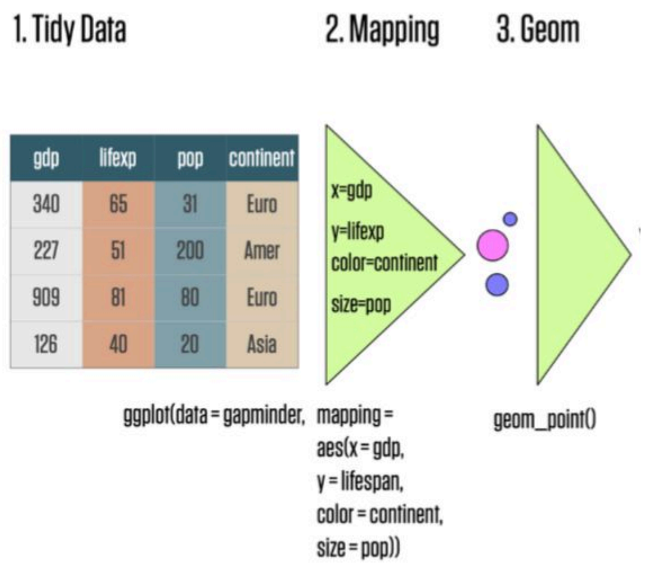
- Numeric vs. Numeric: Scatterplot, line plot
- Numeric vs. Character: Box plot

Why focus on these plots?



https://bit.ly/ir-w3

We build a plot one part at a time



gdp	lifexp	pop	continent
340	65	31	Euro
227	51	200	Amer
909	81	80	Euro
126	40	20	Asia

Data +

Mapping to data +

Geometry

Think about making plots like using recipes from a cookbook: https://r-graphics.org/

One variable plots

Building a Histogram

ggplot(penguins) +	Data +
aes(x = bill_length_mm) +	Mapping to data +
geom_histogram()	Geometry

`ggplot(penguins)`

ggplot(penguins) +

- We always start with `ggplot()`
- The first argument to `ggplot()` is the data
- We add details to the plot with the `+` (plus sign)

`aes()`:

aes(x = bill_length_mm) +

- We map data in with the `aes()` (aesthetic) function
- `x` is an *aesthetic* - it maps data to a visual property
- In the `aes()` function, we use bare column names: `bill_depth_mm`
- If you want to know what aesthetics to map, look at the geom documentation:
  - `?geom_histogram()`

Some aesthetic properties

Aesthetic	Description
`x`	x-coordinate on graph
`y`	y-coordinate on graph
`color`	color of point or line
`alpha`	transparency
`size`	size of point or thickness of line
`group`	group that data belongs to

Not all `geom_`s support every aesthetic properties

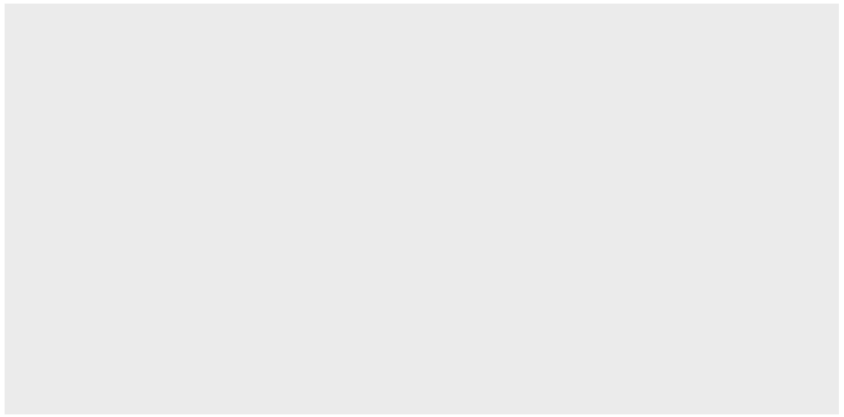
`geom_histogram()`

geom_histogram()

- All geometries begin with `geom_`
- `geom_`s require specific aesthetics
- Tells ggplot2 how to arrange data on page
- When in doubt, look at the documentation:
  - `?geom_histogram`

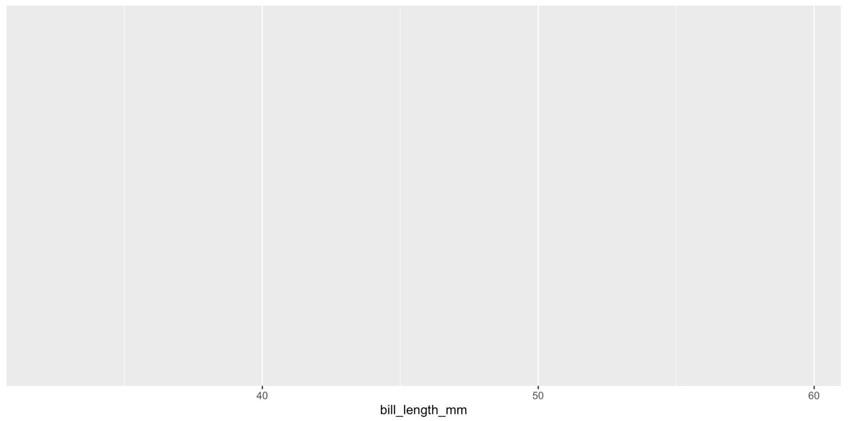
Taking it one part at a time

```r
ggplot(penguins)
```



Taking it one part at a time

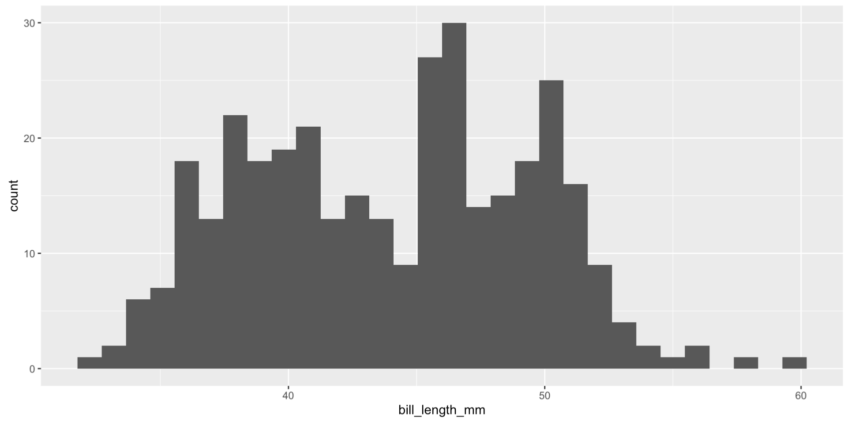
```r
ggplot(penguins) +
  aes(x = bill_length_mm)
```



https://bit.ly/ir-w3

Taking it one part at a time

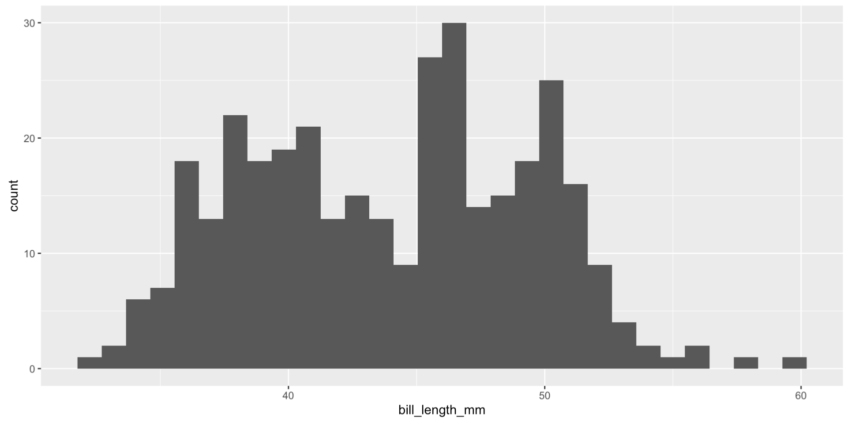
```
ggplot(penguins) +
  aes(x = bill_length_mm) +
  geom_histogram()
```



https://bit.ly/ir-w3

Histogram recap

```
ggplot(penguins) +
aes(x = bill_length_mm) +
geom_histogram()
```

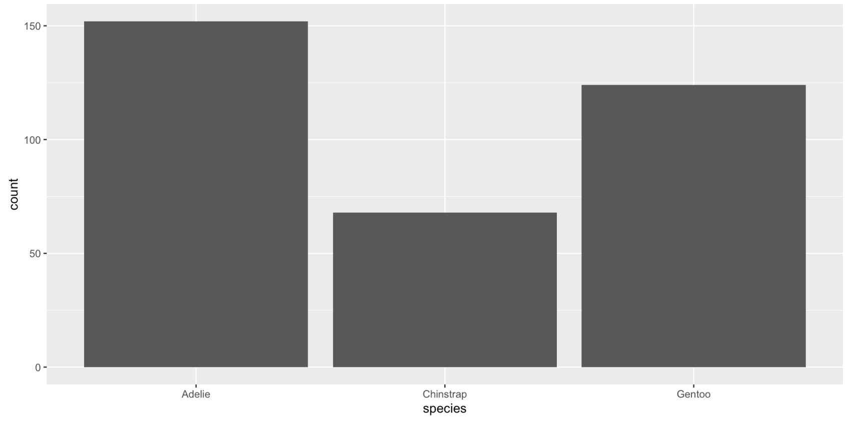


https://bit.ly/ir-w3

Bar plots

Made for categorical data. Bar plots automatically count each group for you, so you only need to provide one variable (axis).

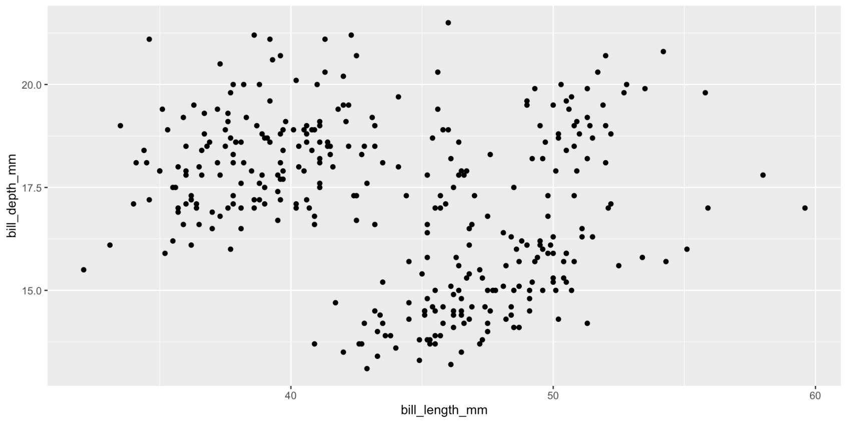
```
ggplot(penguins) +
aes(x = species) +
geom_bar()
```



Two Variable Plots

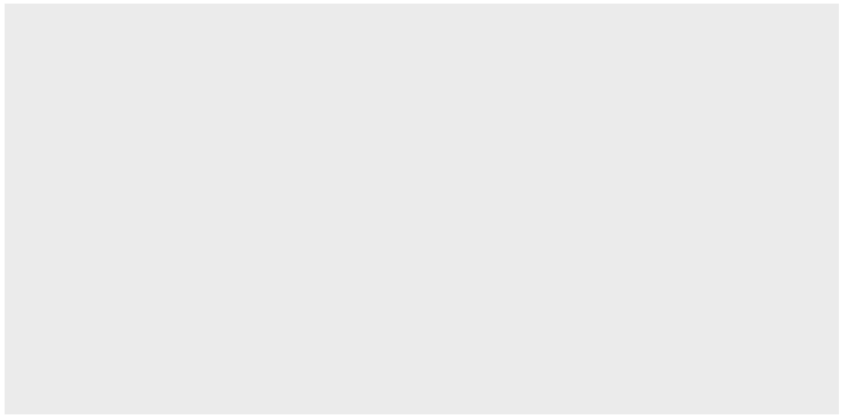
Scatterplot

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



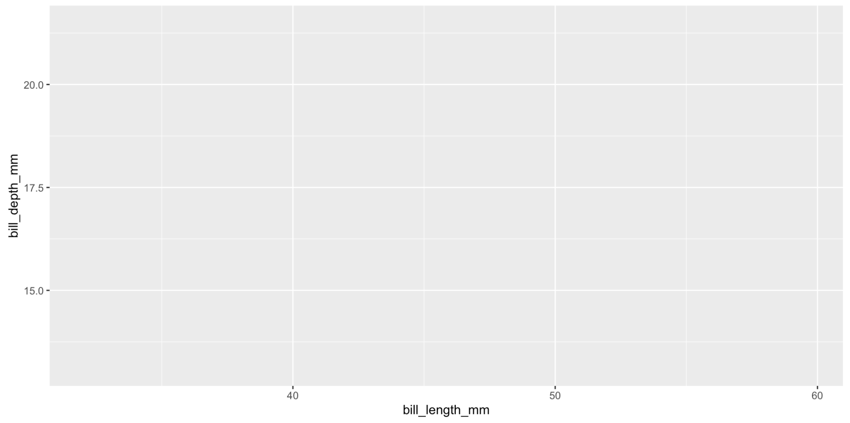
Scatterplot (data)

```r
ggplot(penguins)
```



Scatterplot (aesthetics)

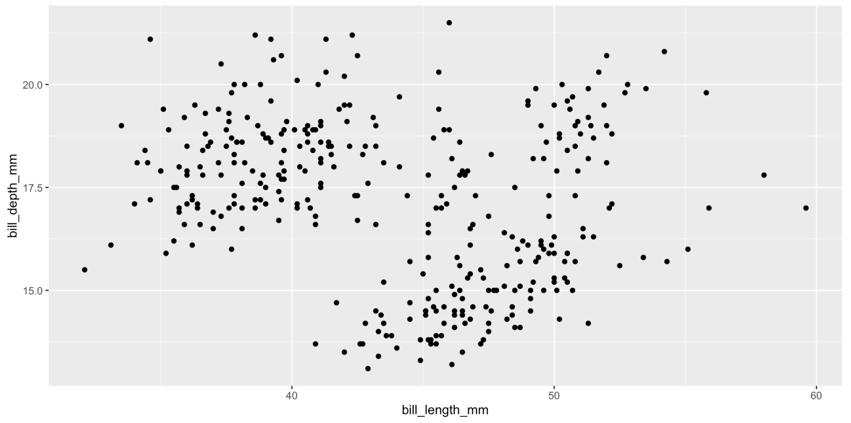
```r
ggplot(penguins) +
  aes(x = bill_length_mm,
      y=bill_depth_mm)
```



https://bit.ly/ir-w3

Scatterplot (geometry)

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



Note: Where to put aes()

Our code looks like this:

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

Most ggplot code looks like this:

```r
ggplot(penguins, mapping = aes(x = bill_length_mm, y=bill_depth_mm)) +
  geom_point()
```

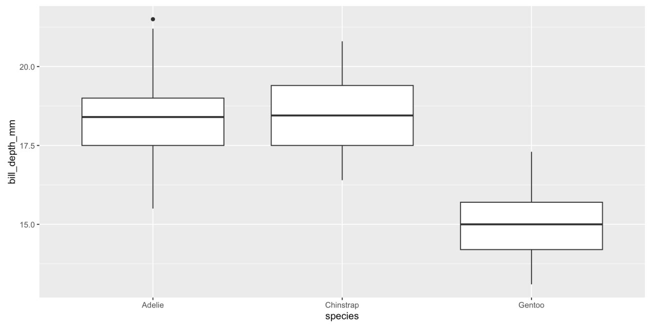
Either is acceptable!

Boxplot

ggplot(penguins) +

aes(x = species, y =
bill_depth_mm) +

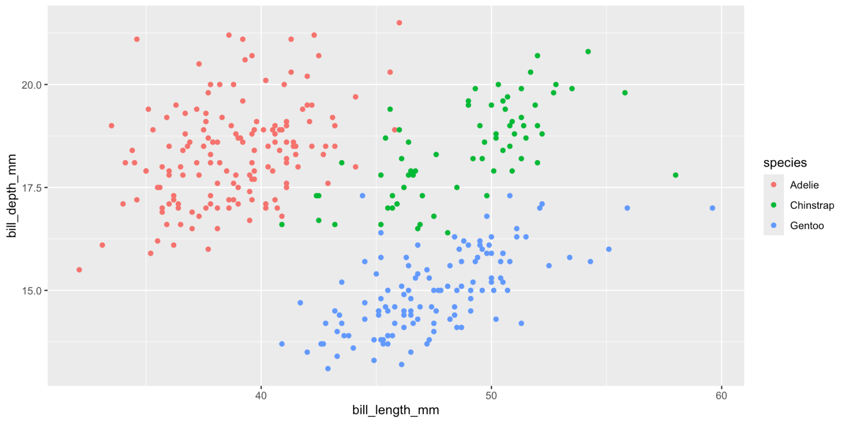
geom_boxplot()



What about more than two variables?

Three Variables

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



https://bit.ly/ir-w3

Additions to Basic Plots

And the Rest

Data +

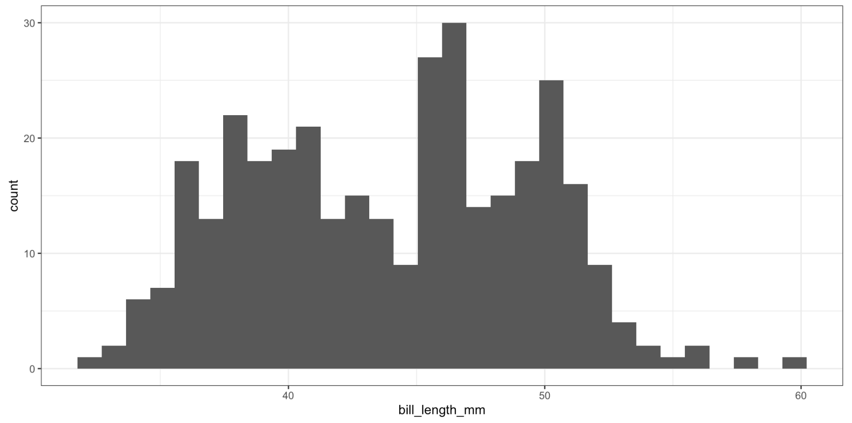
Mapping to data +

Geometry +

Layout Options

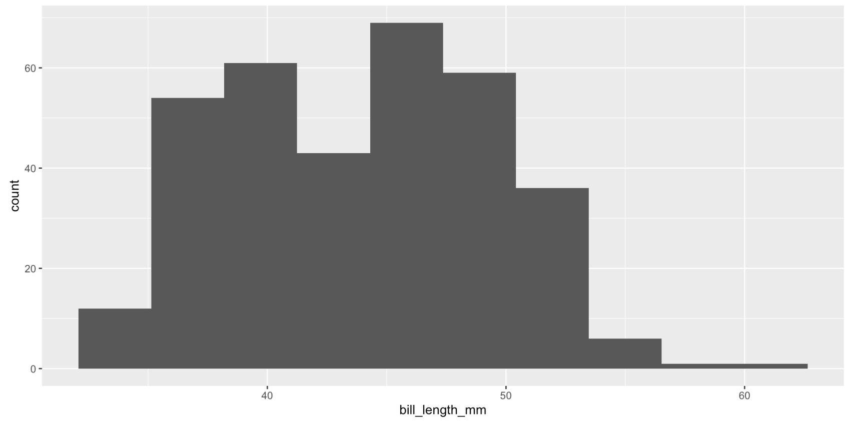
Histogram with a plot theme

```
ggplot(penguins) +
aes(x = bill_length_mm) +
geom_histogram() +
theme_bw()
```



Histogram with options

```
ggplot(penguins) +
aes(x = bill_length_mm) +
geom_histogram(bins = 10)
```

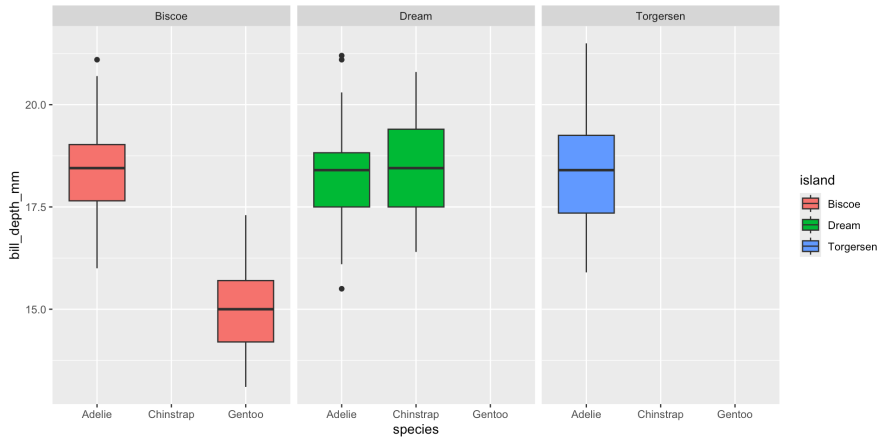


https://bit.ly/ir-w3

Faceting

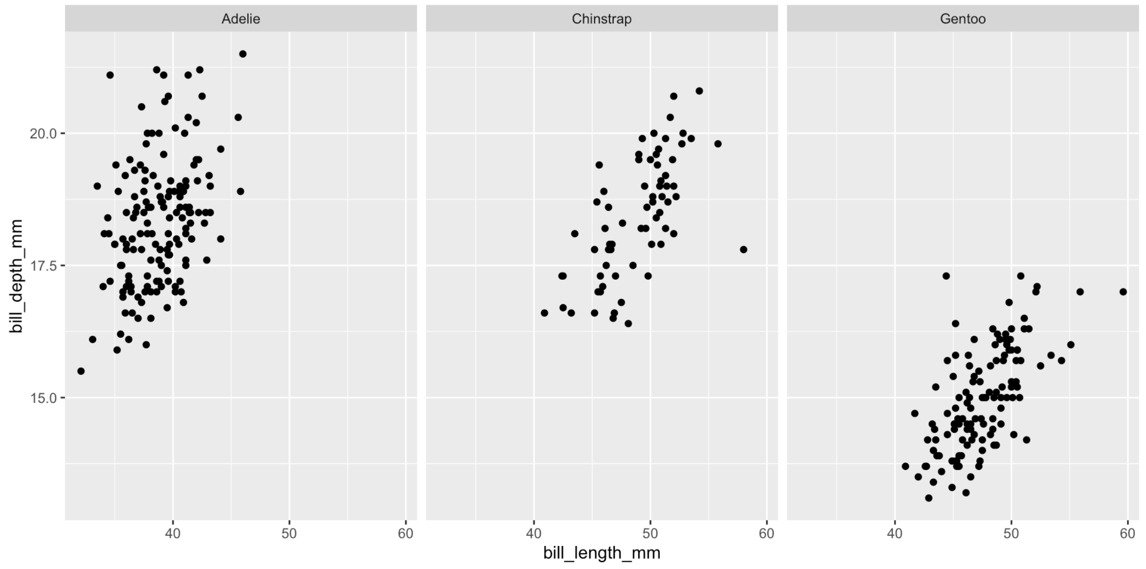
Stratify our plot based on another categorical variable

```
ggplot(penguins) +
aes(x = species, y = bill_depth_mm, color = species) +
geom_boxplot() +
facet_wrap(~island)
```



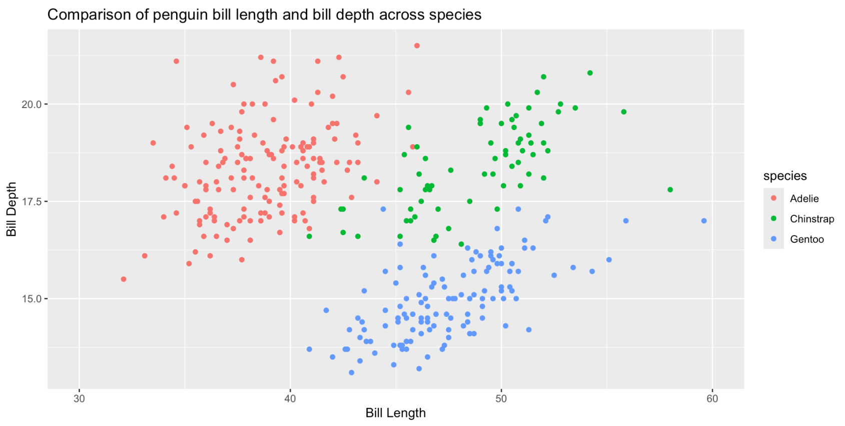
Multivariate Scatterplot by facet

```
ggplot(penguins) +
aes(x = bill_length_mm, y = bill_depth_mm) +
geom_point() + facet_wrap(~species)
```



Some additional options

```
ggplot(data = penguins) +
aes(x = bill_length_mm, y = bill_depth_mm, color = species) +
geom_point() +
labs(x = “Bill Length”, y = “Bill Depth”, title = “Comparison of penguin bill length and bill depth across species”)
```

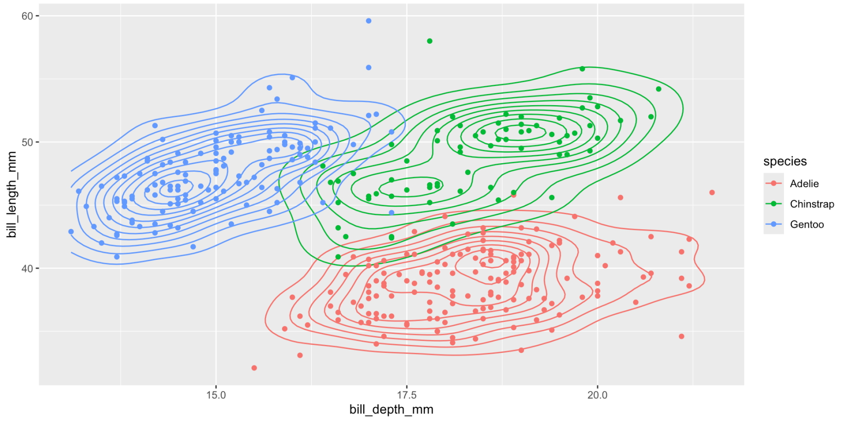


Layering Geometries

Adding on to a plot

- You can layer multiple compatible geometries
  - Must share aesthetics
  - Added one layer at a time

```
ggplot(penguins) +
  aes(x=bill_depth_mm,
      y=bill_length_mm,
      color=species) +
  geom_density_2d() +              # 2d density geom
  geom_point()
```



https://bit.ly/ir-w3

geom_tile() + geom_text() = heatmap

Why is this heatmap missing boxes? Hint: look at penguin counts.

Look at the `count()` function and see if there's an argument we can set to fill in the missing boxes.

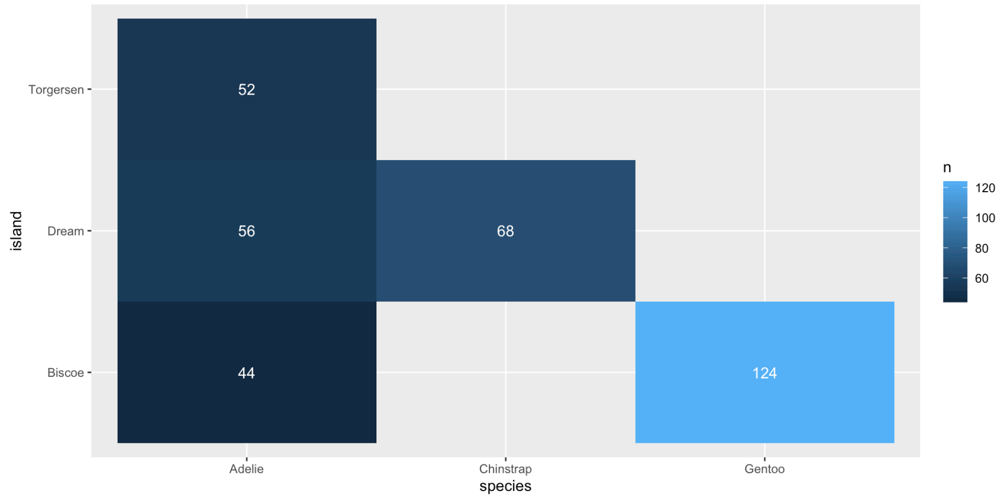
```r
penguin_counts <- count(x=penguins, species, island)
penguin_counts
```

```
# A tibble: 5 × 3
  species   island        n
  <fct>     <fct>     <int>
1 Adelie    Biscoe       44
2 Adelie    Dream        56
3 Adelie    Torgersen    52
4 Chinstrap Dream        68
5 Gentoo    Biscoe      124
```

Missing Values - How to Fix?

A task in your exercise for the week!

```r
ggplot(penguin_counts) +
  aes(x=species,
      y=island,
      fill=n) +
  geom_tile() +
  geom_text(aes(label=n),
            color="white")
```



https://bit.ly/ir-w3

`esquisse` as a helper

Consider the `esquisse` package to help generate your ggplot code via drag and drop.

```r
library(esquisse)
esquisser(penguins)
```

For More Practice:

- R-Bootcamp Chapter 1
- R-Bootcamp Chapter 2
- Better Plots

Community Session Next Week

- Optional
- Suggest a Topic and Vote in Google Doc