Customizing scales `ggplot2`

Data Science for Biologists, Fall 2020

Stephanie J. Spielman

> WORD TO THE WISE: You will often be tempted to google how to use `ggplot2`. This is absolutely the right thing to do, but be careful!! Not all internet posts are created equally, and *many webpages or blogs may use DIFFERENT VERSIONS of* `ggplot2`. When asking the internet for help, always make sure you can locally (on your computer/in RStudio Cloud) run and understand little code tidbits you find. Remember: just because a page looks helpful doesn't mean it is.

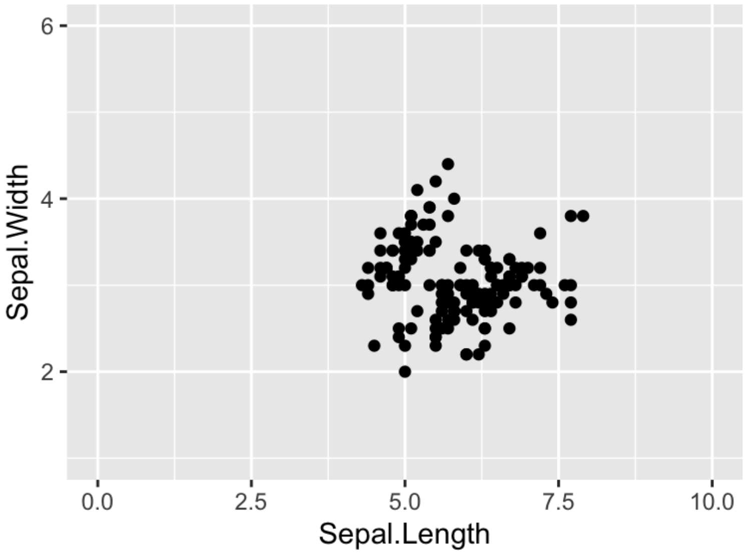
> This tutorial uses some theme customization in some of the plots. You may want to read this theme tutorial (http://htmlpreview.github.io/?https://github.com/sjspielman/datascience_for_biologists/blob/master/fall2020/tutorials/themes.html) first.

The specific ranges and positioning of axes are a type of *scale* in `ggplot2`. This tutorial demonstrates a minimal, but important, subset of ways you can modify the axis scales. All examples below will modify a version of one of these two plots:

Simply changing the ranges (limits)

`ggplot2` will automatically decide the axis limits for a given plot you make, but you can easily customize this with the quick functions `xlim(<low>, <high>)` and `ylim(<low>, <high>)`. For example:

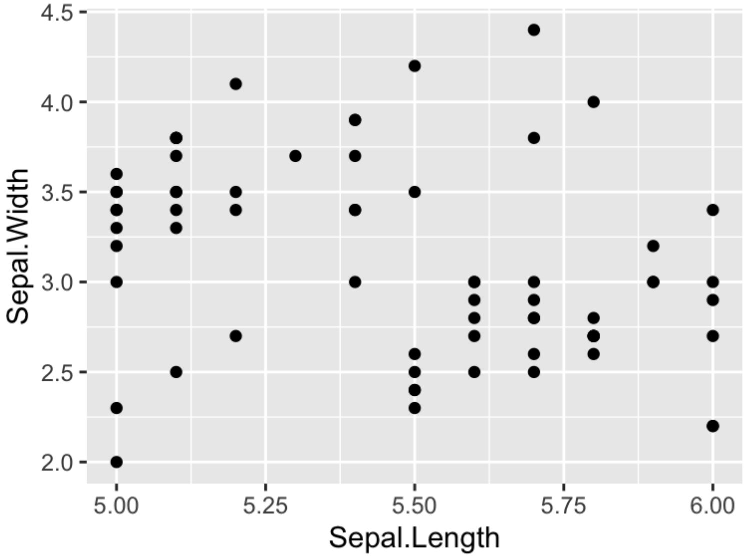
```
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  xlim(0, 10) + # Set x-axis to range from 0-10
  ylim(1, 6) # Set y-axis to range from 1-6
```



Be careful to pay attention to *warning messages* if you change the axis limits, as you may accidentally remove points from the plot! **Unless you have a specific reason for doing this, don't ever intentionally exclude points.** For example, this is usually a bad idea:

```
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  xlim(5, 6) # We know that points range from 4-8! We have therefore removed a b
unch of points from the final plot
```

```
## Warning: Removed 83 rows containing missing values (geom_point).
```



Changing the tick locations and labels

`ggplot2` will also decide where the tickmarks ("breaks") and associated axis text ("labels") appear automatically, but of course, anything can be customized! When changing the breaks and labels, you first need to know *what kind of axis you are dealing with*: Does it use *continuous values* (the full number line) or does it use *discrete values*?

There are different functions to use depending on the type of axis you have: Axes showing *continuous values* use functions `scale_x_continuous()` and `scale_y_continuous()`, and axes showing *discrete values* use functions `scale_x_discrete()` and `scale_y_discrete()` for customizing X and Y axes, respectively.

The key arguments you can use include:

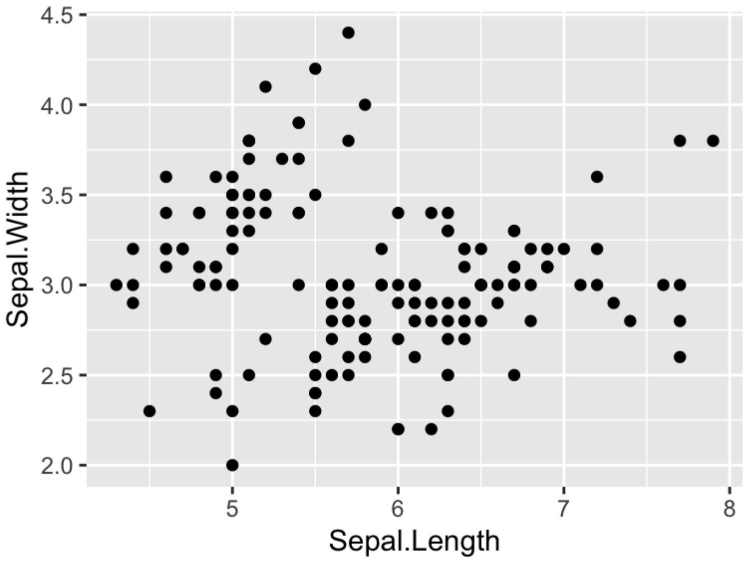
- `breaks`, an array (remember `c()` ??) of where tick marks should be placed along the given axes. The values in `breaks` must be compatible with values on your axis.
- `n.breaks` can instead be used to give the actual number of *major breaks* (ticks) you'd like to see.
- `labels`, an array (remember `c()` ??) of labels for associated ticks, *if you would like to CHANGE them further.* The length of this array must match the length of the `breaks` (or the default number of breaks if you aren't specifying breaks).
- `limits`: an array of `c(low, high)` for the axis limits. This has the same functionality as functions `xlim()` and/or `ylim()`, but can also be specified here if you want.
- `name`: You can also rename the axis (same as using `labs()`) with the argument `name` if you want.

Examples: Continuous axes

> Remember the documentation (https://ggplot2.tidyverse.org/reference/scale_continuous.html)!!

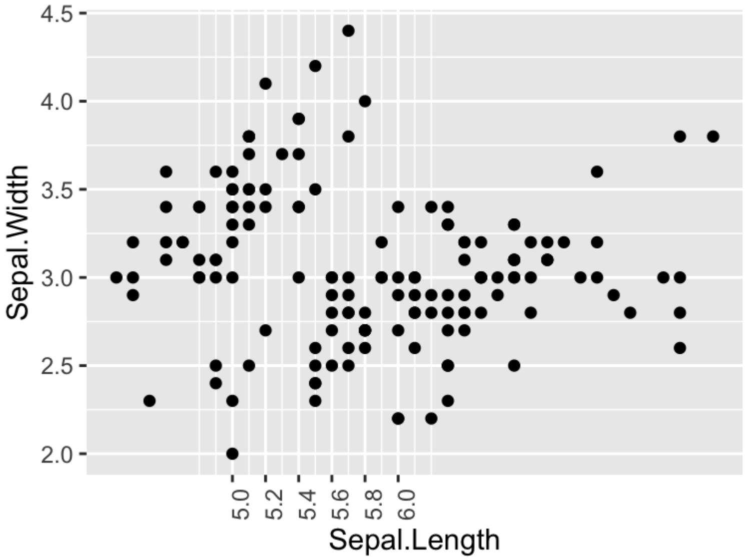
All examples below modify this plot:

```
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point()
```



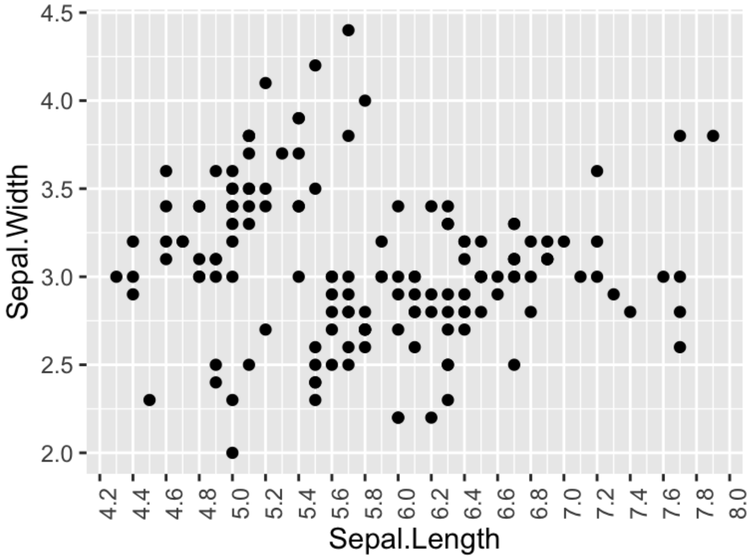
Change the location of tickmarks:

```
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  # Specify placement for ticks (terrible plot, good demo!)
  scale_x_continuous(breaks = c(5, 5.2, 5.4, 5.6, 5.8, 6.0)) +
  theme(axis.text.x = element_text(angle = 90)) # so you can see the tick labels clearly
```



Change the number of breaks to 20, which `ggplot2` will automatically place for you:

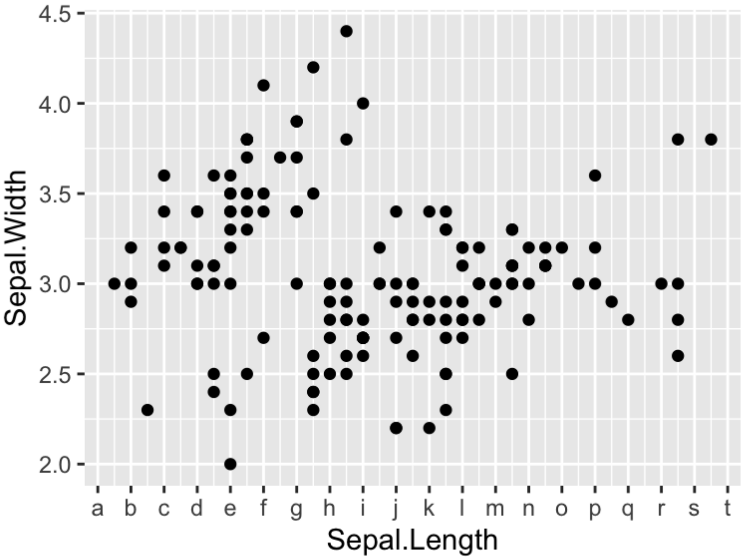
```
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  scale_x_continuous(n.breaks = 20) +
  theme(axis.text.x = element_text(angle = 90)) # so you can see the tick labels clearly
```



Change the number of breaks to 20, but add your own labels!!

```
new_labels <- letters[1:20] # R has built-in variables letters (a-z) and LETTERS
(A-Z). This line says, "take the 1st-20th value in letters and save to a variabl
e called `new_labels`".
# See the variable we will use for labels. Using a variable means our code is mo
re reproducible!
new_labels
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r"
 "s"
## [20] "t"

# And the plot:
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  scale_x_continuous(n.breaks = 20, labels = new_labels)
```

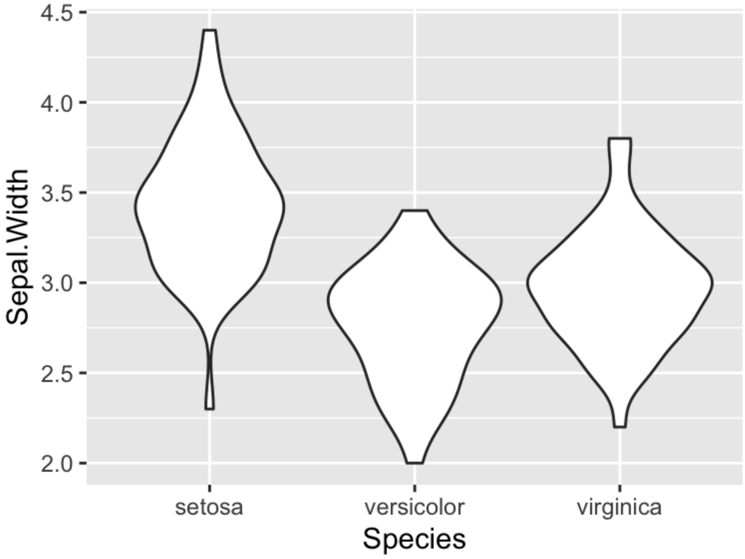


Example: Discrete axes

> Remember the documentation (https://ggplot2.tidyverse.org/reference/scale_discrete.html)!!

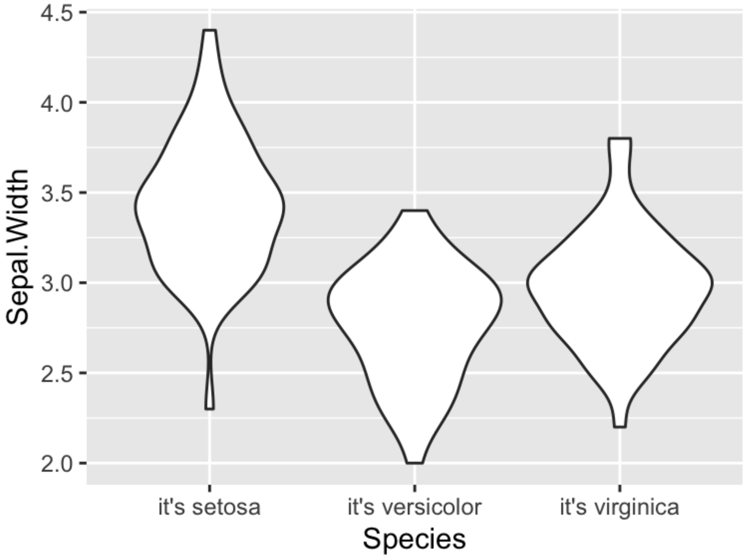
The subsequent example below modifies this plot:

```
ggplot(iris, aes(x = Species, y = Sepal.Width)) +
  geom_violin()
```



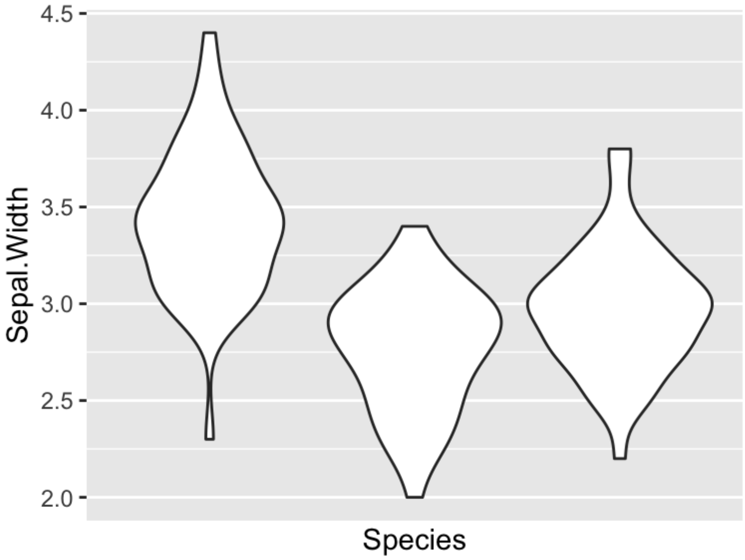
Change the labels of tickmarks:

```
ggplot(iris, aes(x = Species, y = Sepal.Width)) +
  geom_violin() +
  scale_x_discrete(labels = c("it's setosa", "it's versicolor", "it's virginica"
))
```



For discrete data like `Species`, you will not have much luck changing the number of breaks. This is because the axis is discrete, and the breaks are what they are:

```
ggplot(iris, aes(x = Species, y = Sepal.Width)) +
  geom_violin() +
  # there aren't 10 values along the X-axis; only 3!! So, ggplot2 bails.
  scale_x_discrete(breaks = 10)
```


4.5
4.0
3.5
3.0
2.5
2.0
Sepal.Width
Species