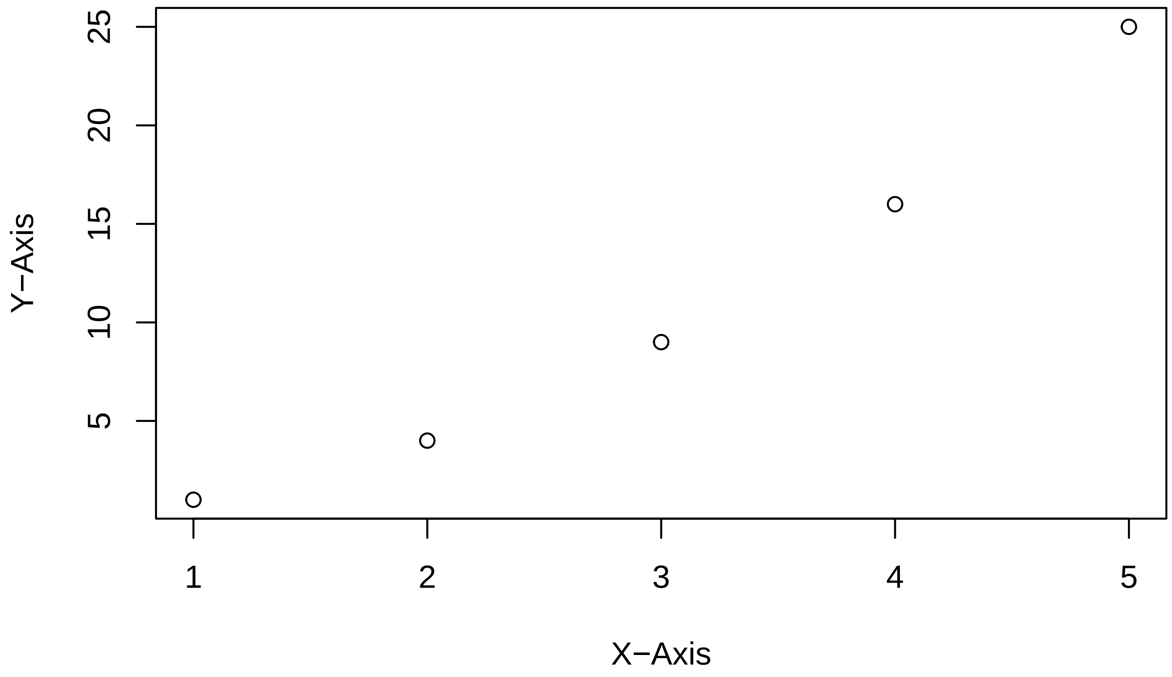
CM515 Day 1: Plotting with ggplot()

1. Basic Plot Functions

R has a built-in basic plot function. However it is pretty limited in scope.

```
####
#01#
####
plot(c(1,2,3,4,5),c(1,4,9,16,25), xlab="X-Axis", ylab="Y-Axis")
```



2. ggplot()

Before I welcome you to the future with ggplot, we need some data.

A ton of datasets can be accessed with data().

```
####
#02#
####
data()
```

2b. ggplot()

We will use the iris data set primarily.

It is worth getting familiar with the data.

```
####
#02b#
####

?iris

head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##       Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

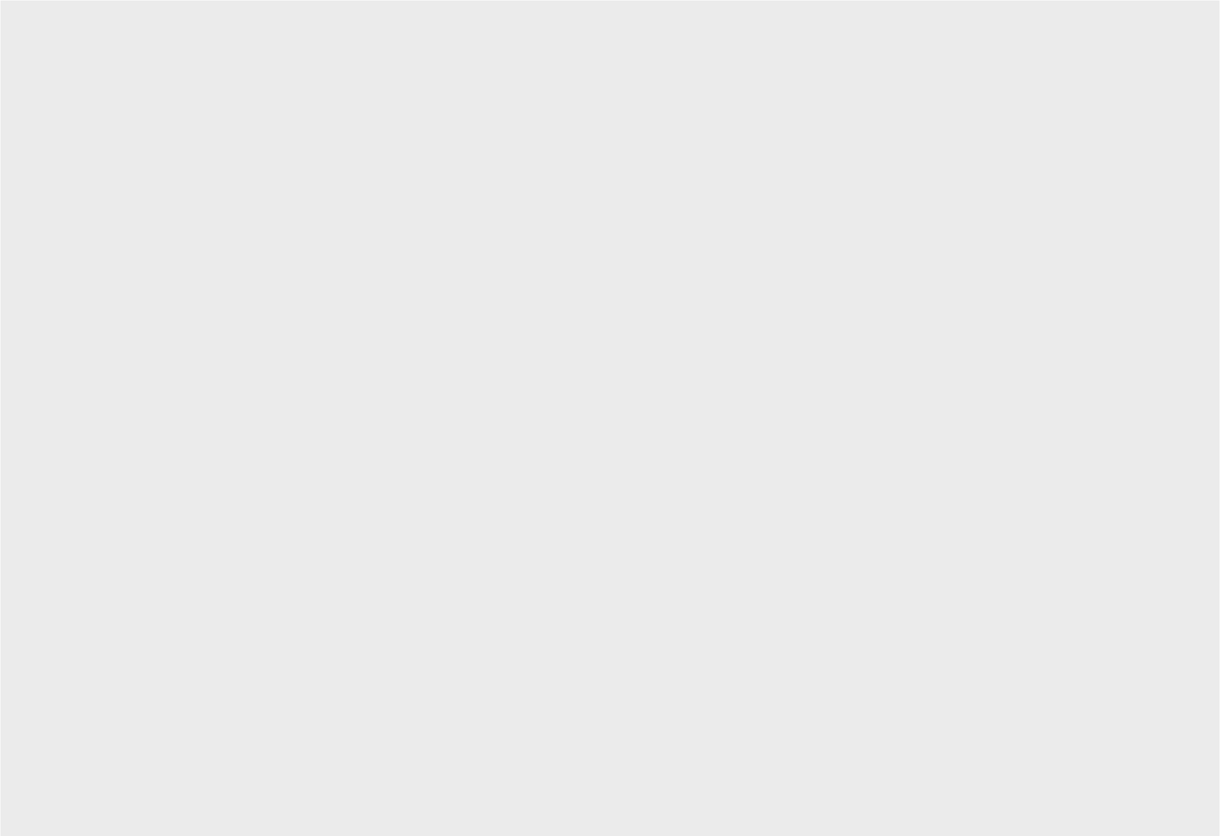
2c. ggplot()

Now let’s try to graph, just using ggplot().

. . . well?

Nothing happens because we did not specify aesthetics or a geometry.

```
####
#02c#
####
iris %>%
  ggplot()
```

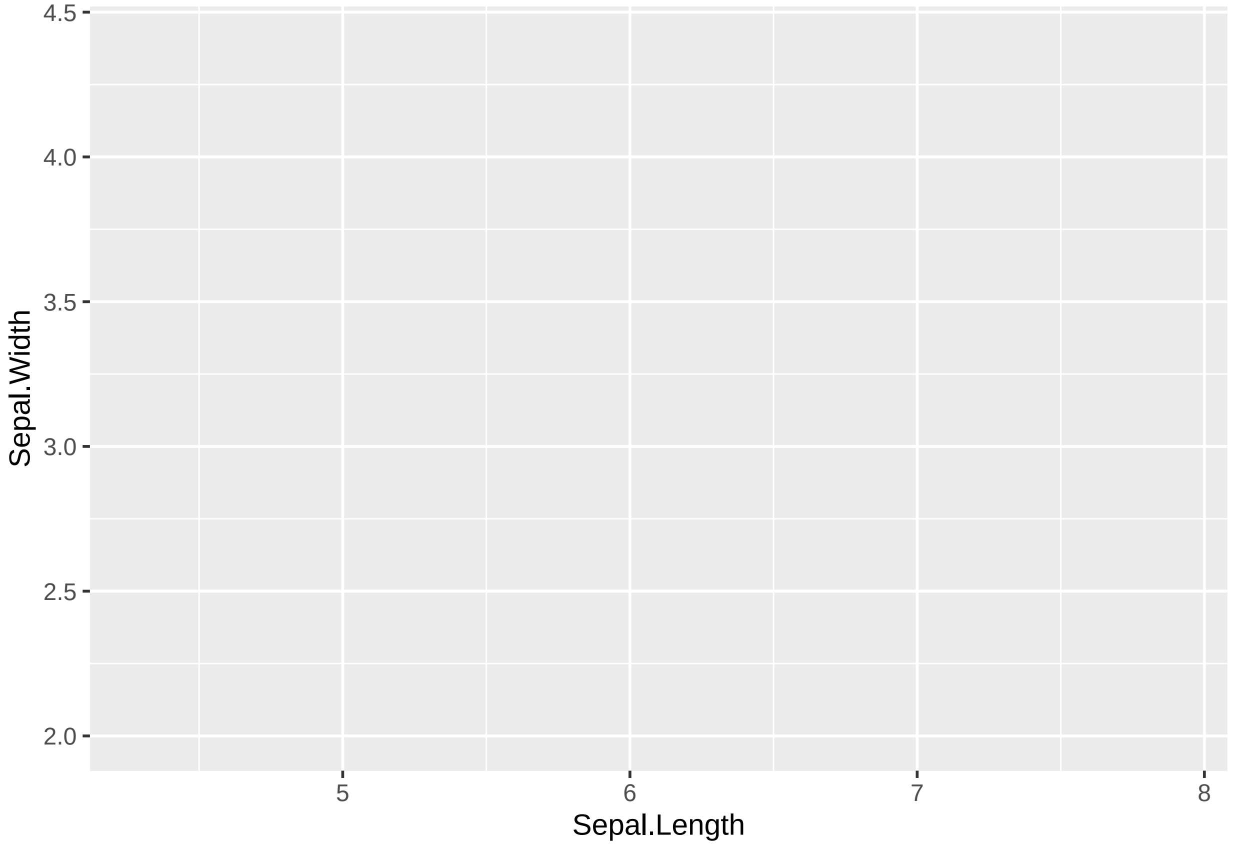


---

3. Mapping x and y Aesthetics with aes() Inside of ggplot()

We’re closer now but we didn’t specify a geometry.

```
####
#03#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width))
```

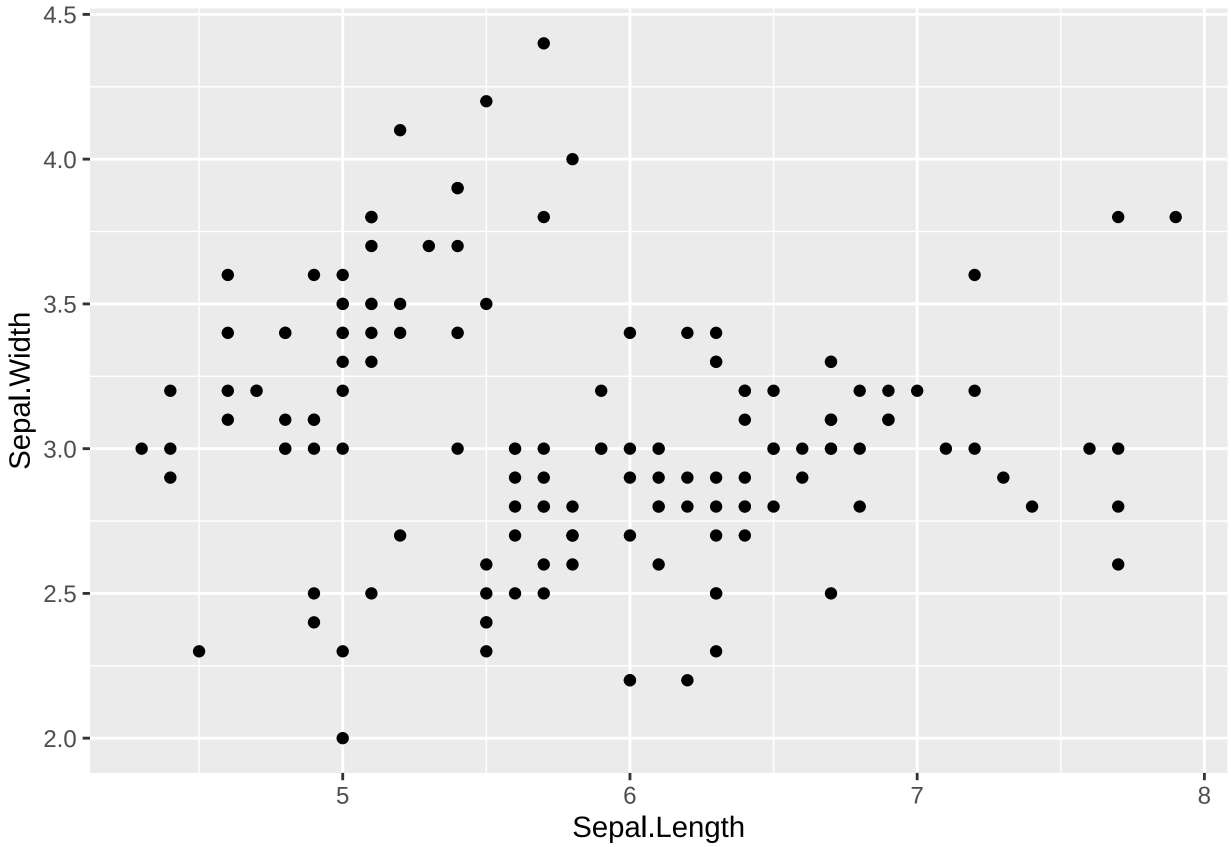


---

4. Specifying a Geometry

Specifying aesthetics and a geometry are the minimum requirement to make a plot using ggplot(). In this case, I specified geom_point(). Notice that I added it to the ggplot() function with "+". "+" in this case acts like %>% for all code involving ggplot.

```
####
#04#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point()
```



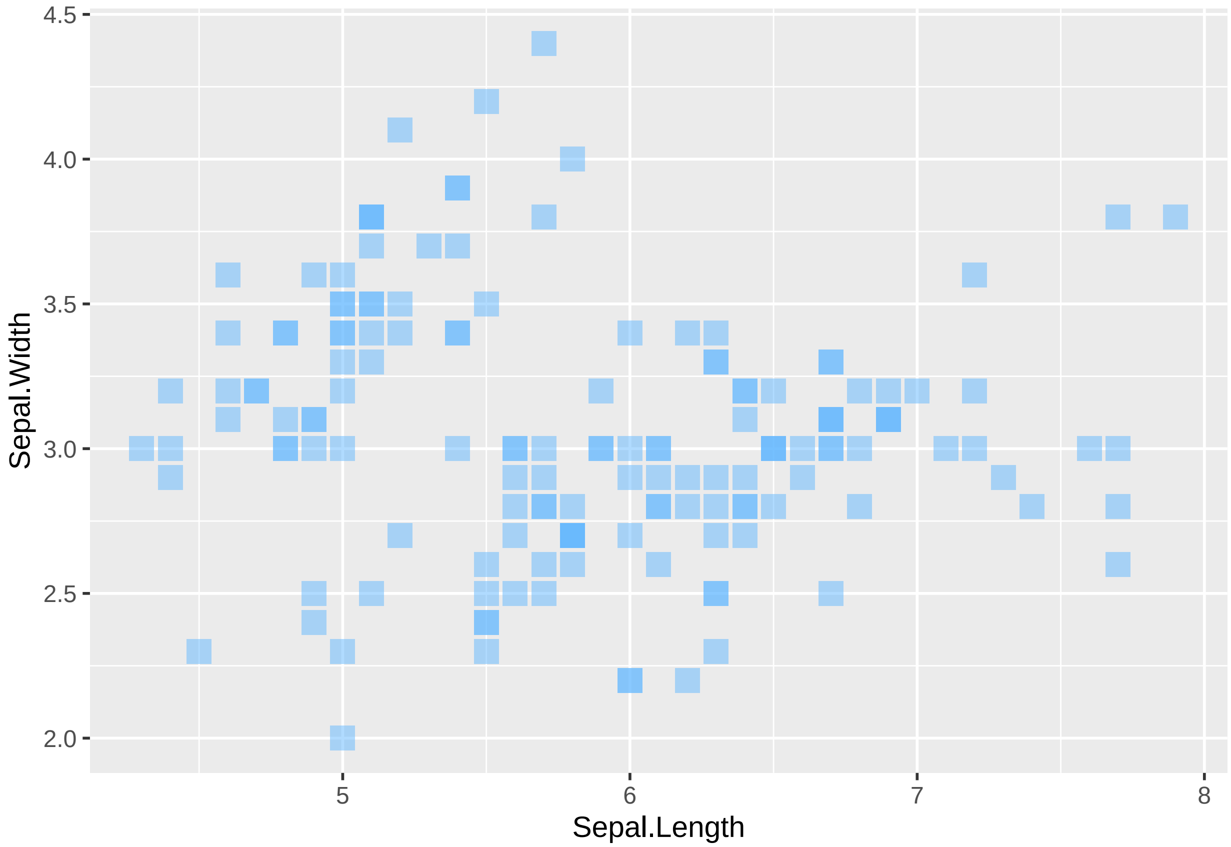
4b. More about Geometry

Each geometry has its own unique aesthetics and content. The aesthetics of a scatter plot require x and y coordinates. This scatter plot allows us to modify:

size=# (for the size of points) color=" " (for the color) shape= " " (for the shape of the points) alpha= a range of 0-1 alpha is transparency where 1 is transparent and 0 is visible (great for overlapping points)

Play around with these and see what you like

```
####
#04b#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point(color='steelblue1', size=4, alpha=0.5, shape="square")
```



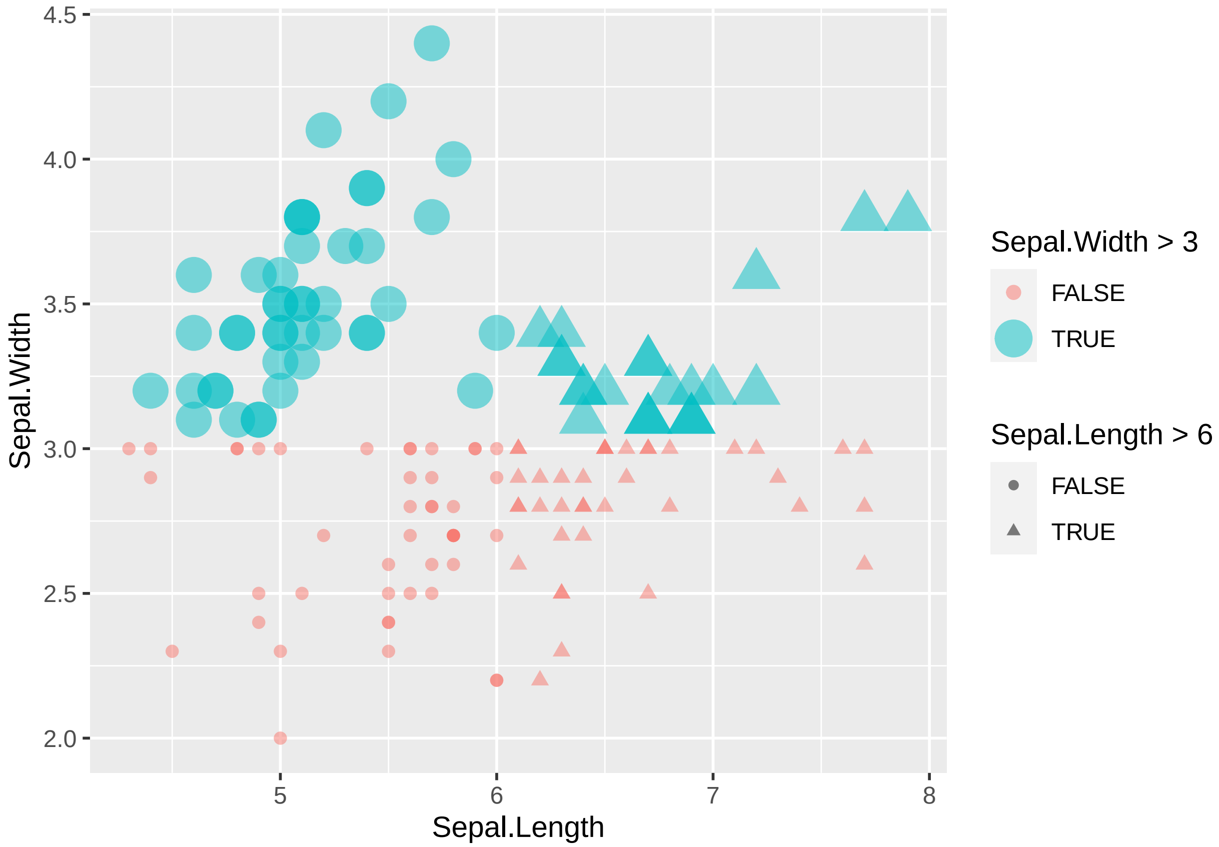
4c. Manipulating Geometry with Aesthetics

Because geometry and aesthetics are somewhat intertwined, you can also make these changes in your mapping.

Size and color do not have to be limited to specifics, but can also be linked to expressions

Note that ggplot automatically creates legends for your specification in mapping.

```
####
#04c#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, color=Sepal.Width > 3, size=Sepal.Width > 3, shape= Sepa
  geom_point(alpha=0.5)
```



4d. Geometries vs Aesthetics

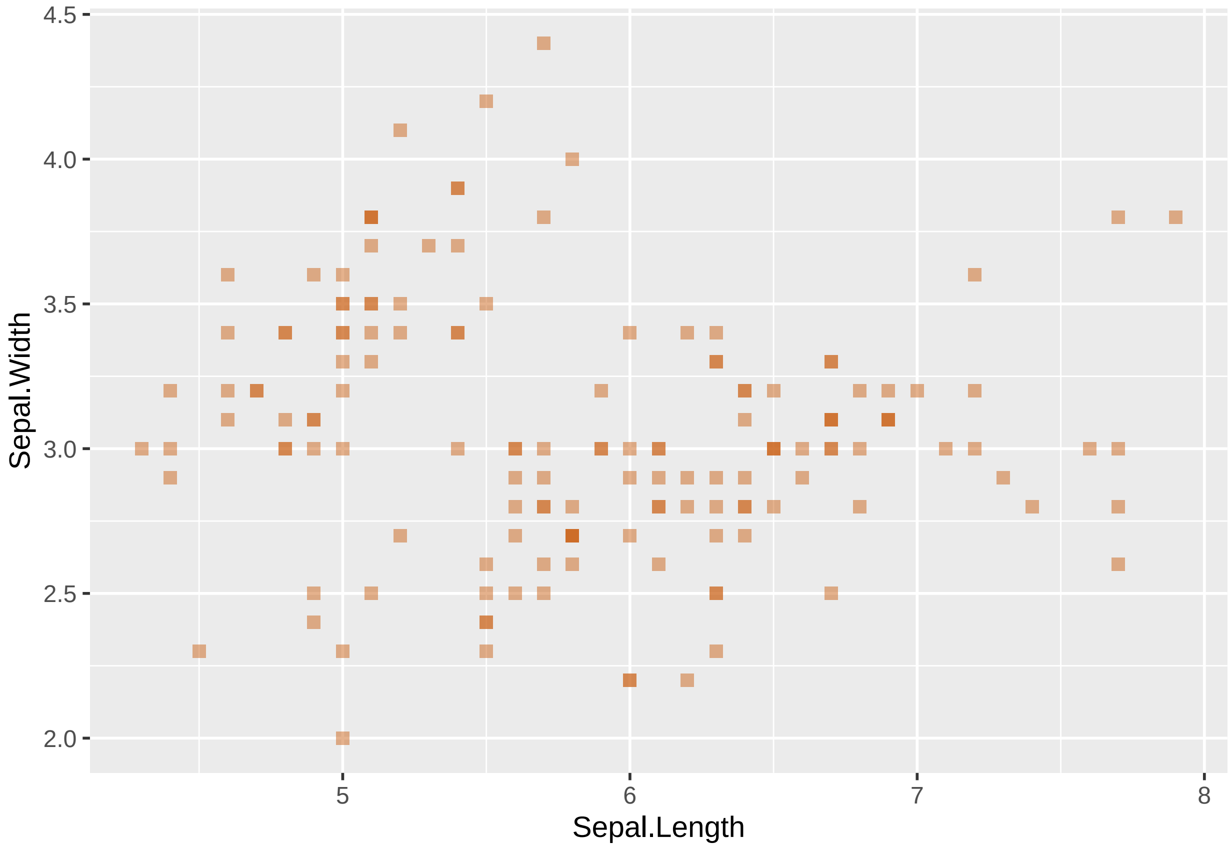
It is important to note that you can customize your graph in both mapping and geometry, but there are advantages to each.

Geometry lets you customize your entire mapping. Aesthetics lets you customize your mapping with respect to your data.

Geometry will override any specifications you make in mapping.

Depending on the situation, you may want your mapping within or outside your geometry.

```
####
#04d#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, color=Sepal.Width > 3, size=Sepal.Width > 3, shape= Sepa
  geom_point(alpha=0.5, color="chocolate3", shape="square", size=2)
```

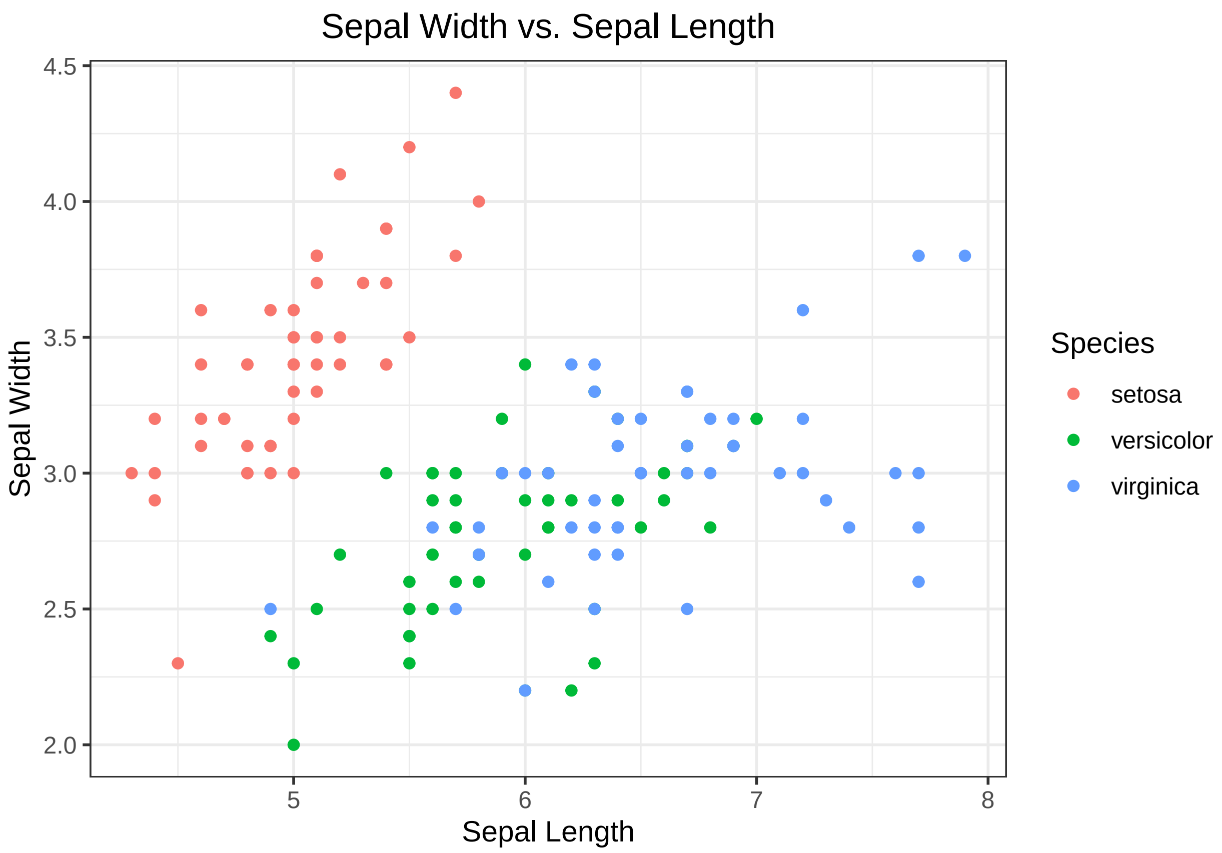


# 5. Coloring by Factor

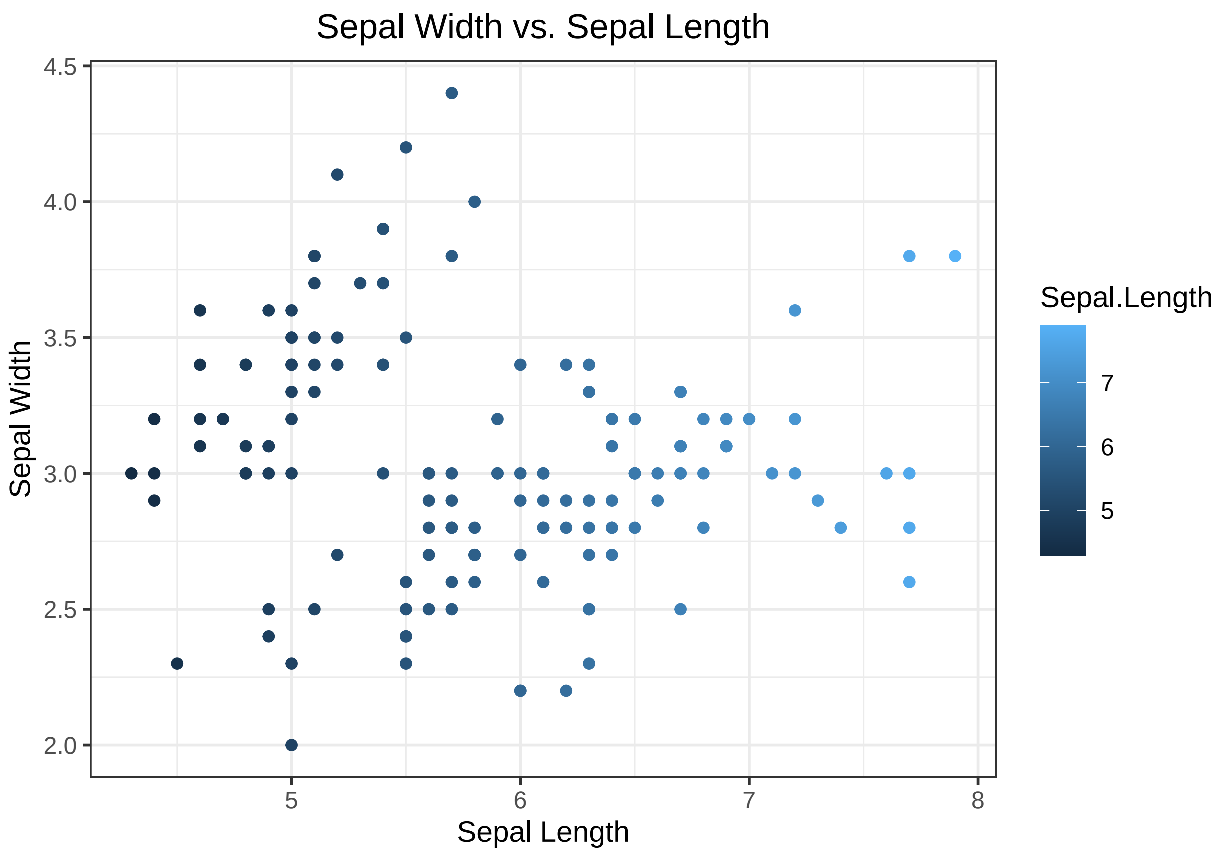
We can color points by a factor, which is typically a categorical variable, to highlight specific trends. This is done by specifying "color" (sometimes abbreviated as "col") within the aes() function. In this case, we specify that the points should be colored according to the factor "Species".

Note: For some geometries and/or variables you may need to use "fill" instead of "color" but overall they function similarly.

```
####
#08#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



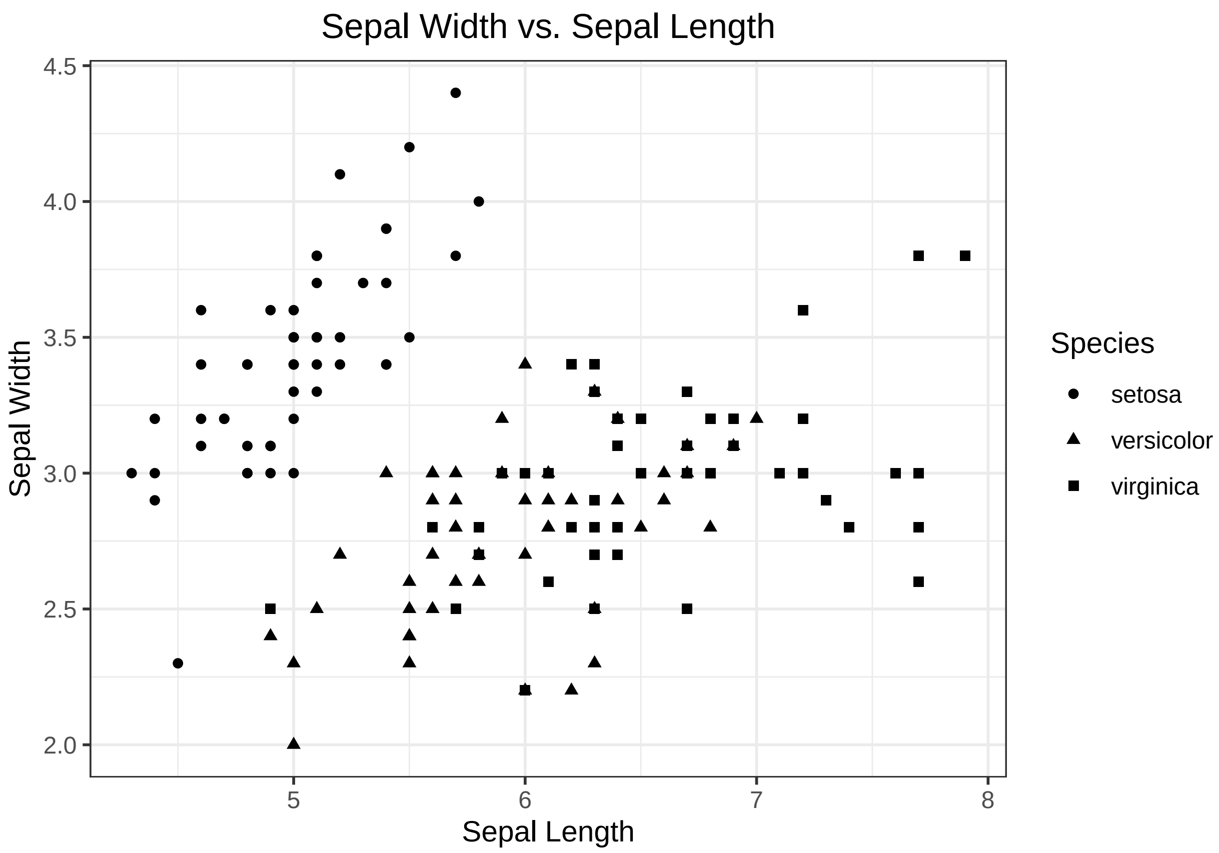
```
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, col = Sepal.Length)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



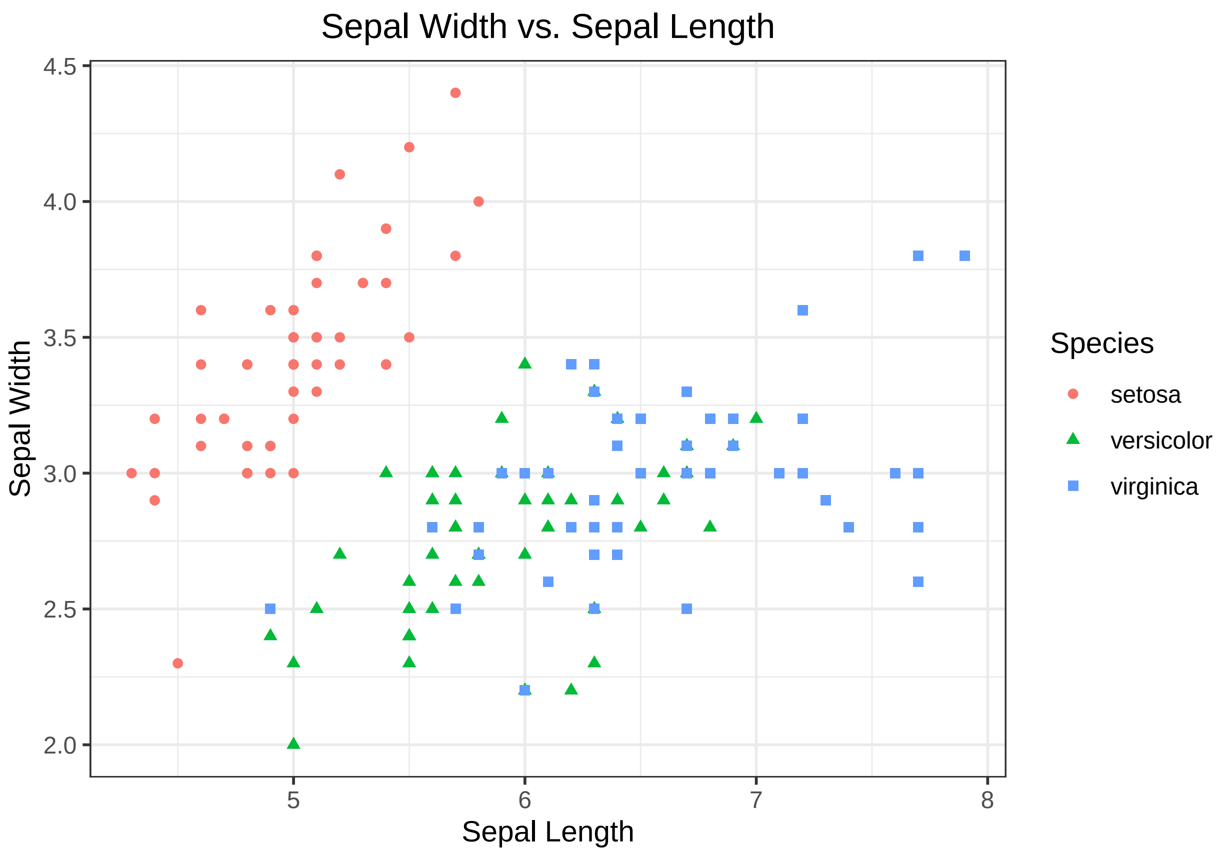
5b. Using Shape to Denote a Factor

Similar to "color" we can also make points different shapes based on a factor. To do so, we specify "shape = Species" inside the aes() function. By specifying "color = Species" and "shape = species", each species gets a unique shape and color.

```
####
#05b#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, shape = Species)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



```
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, color = Species, shape = Species)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



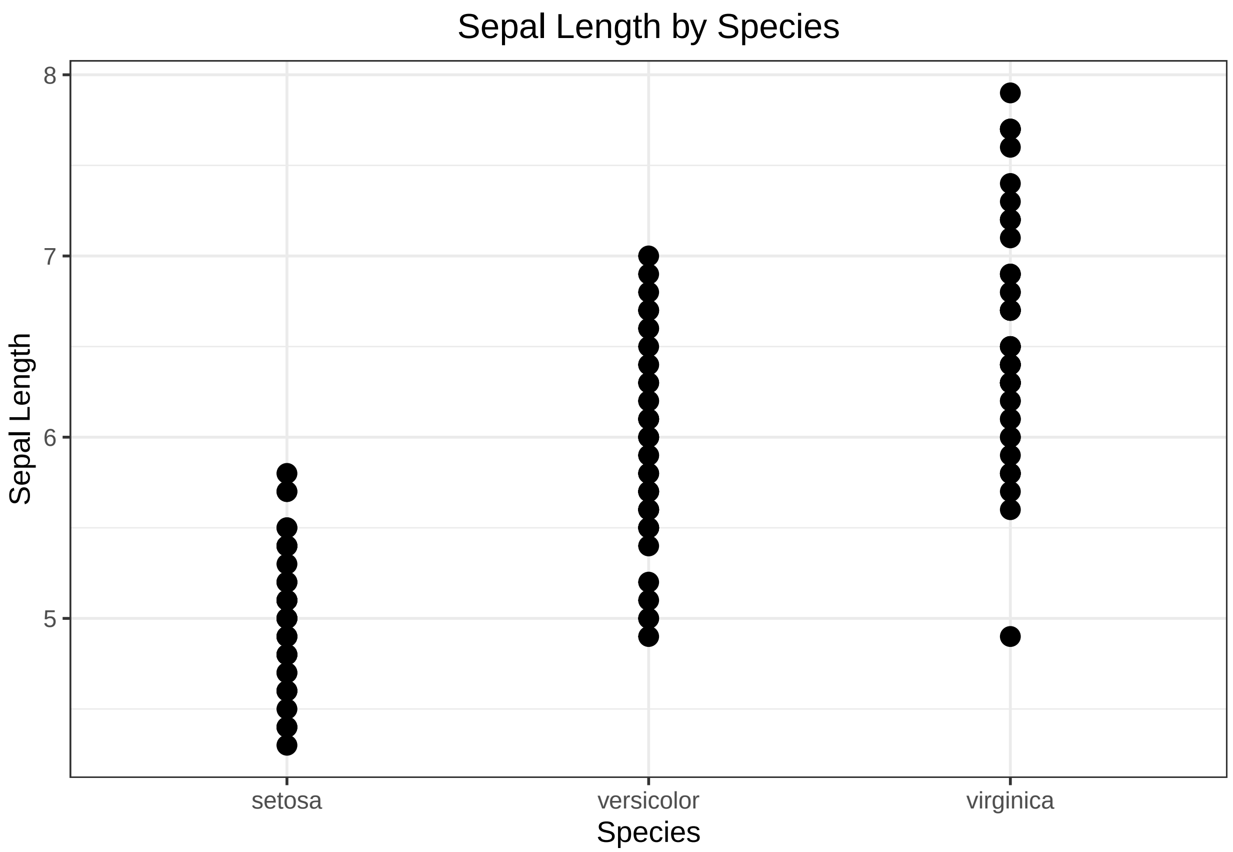
---

6. Plotting a Continuous Variable Against A Categorical Variable

---

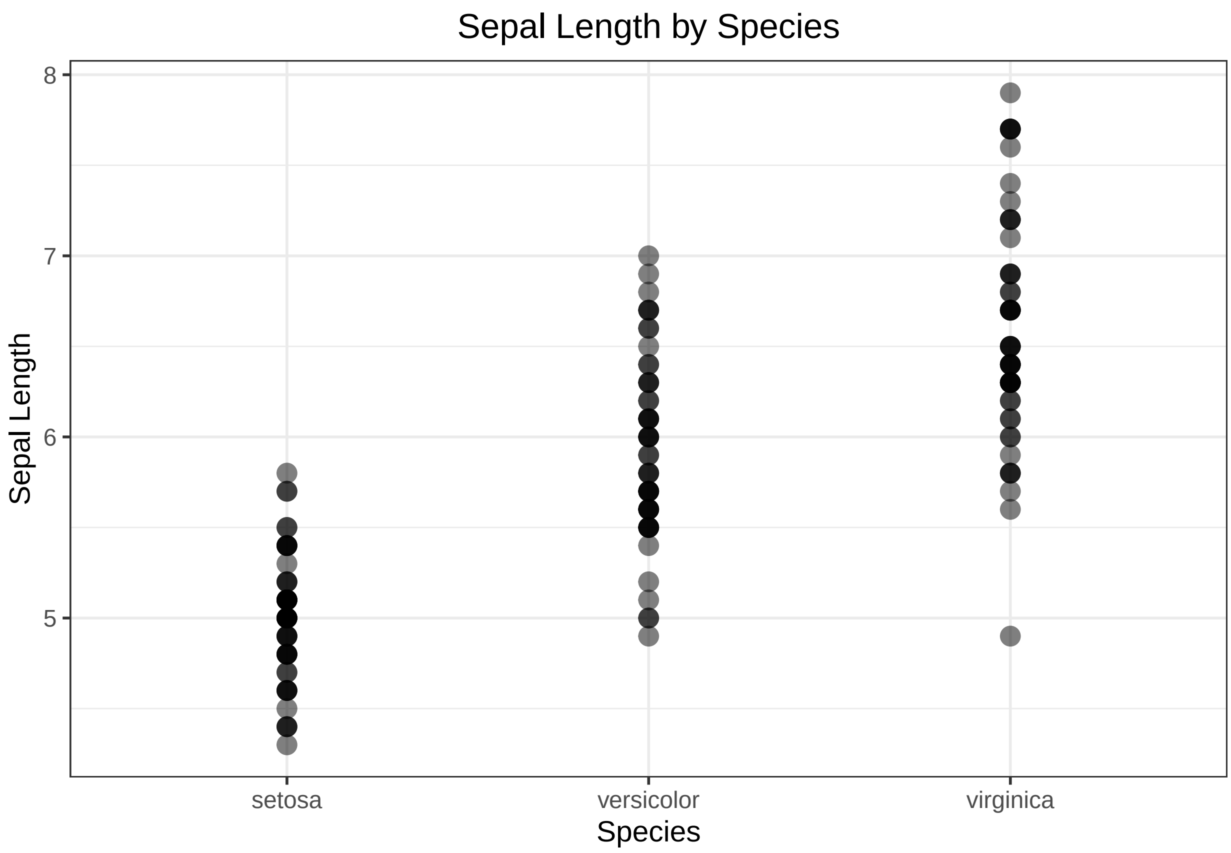
As a Scatter Plot

```
####
#6#
####
iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_point(size=3) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



Some of the points are overlapping. We can try using alpha here for transparency.

```
iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_point(size = 3,
             alpha = .5) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



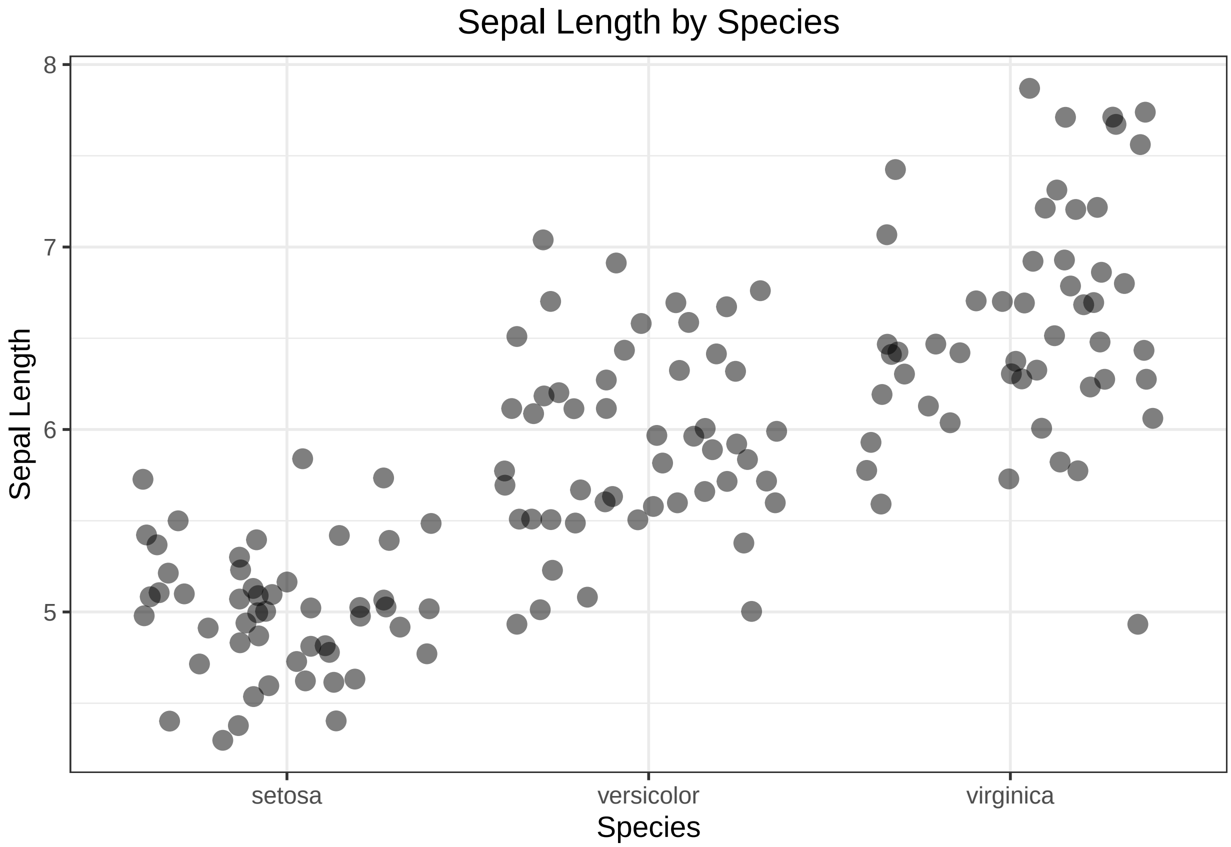
As a Jitter Plot

It looks like we have reached the limitations of a scatter plot for this context.

We will need to explore another geometry. geom_jitter spaces out overlapping points.

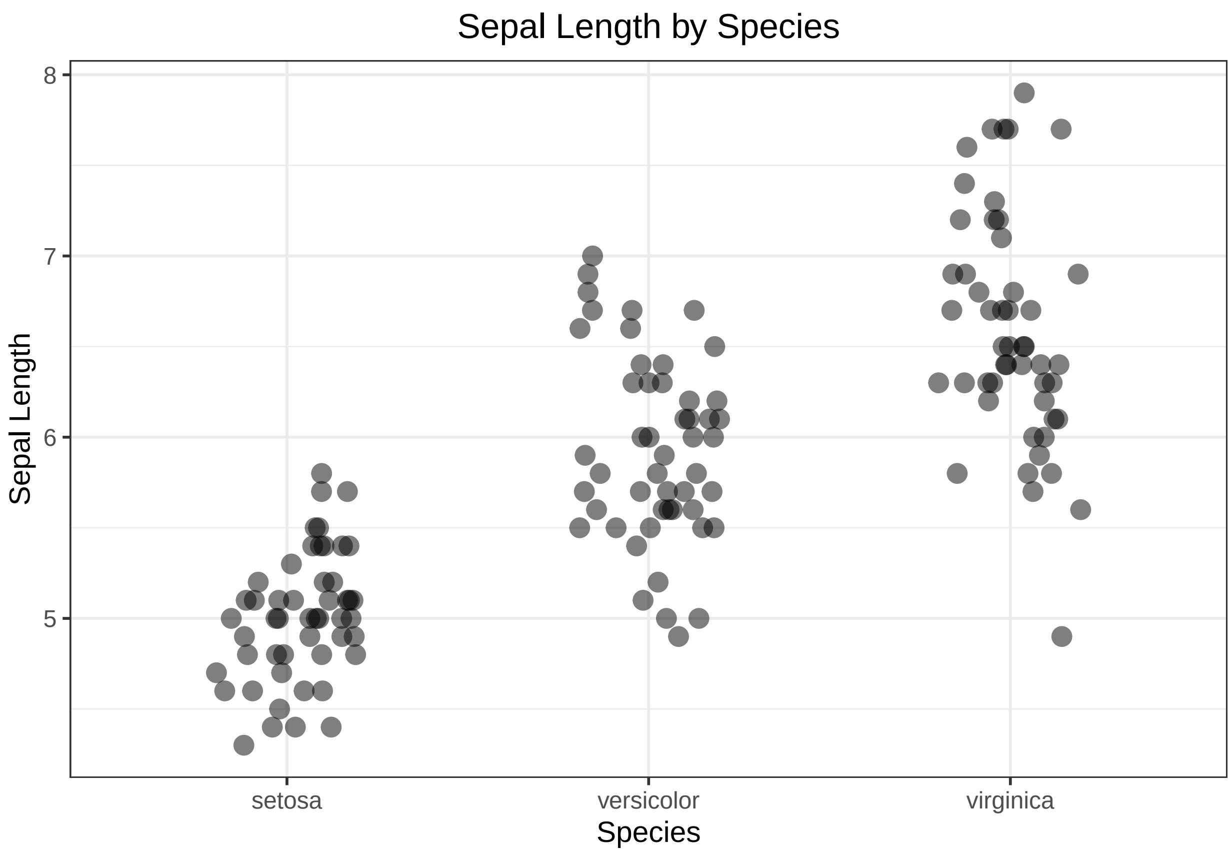
Now we can see individual points better.

```
iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_jitter(size = 3,
              alpha = .5) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



The default "jitter" settings "spread" the points too much for my liking, so I always adjust them using "width" and "height" within the geom_jitter() function.

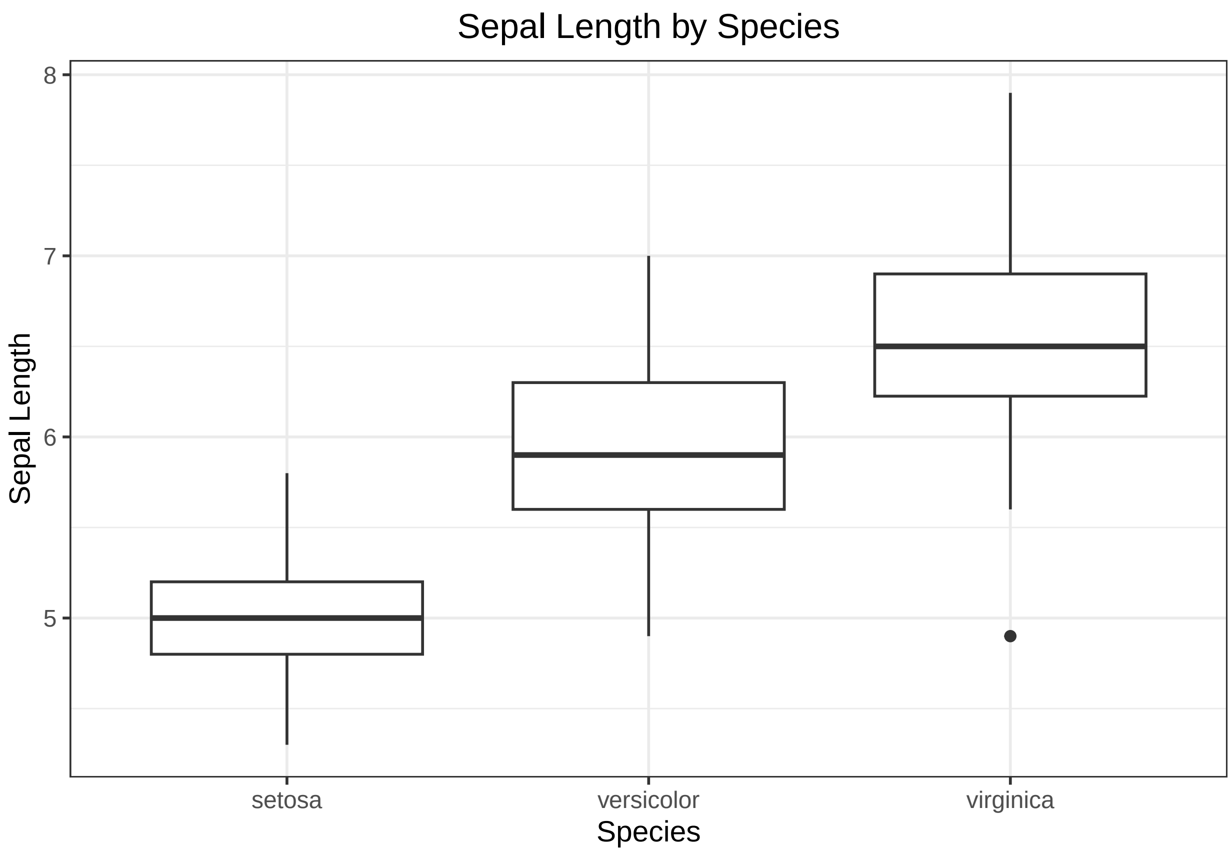
```
iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_jitter(size = 3,
              alpha = .5,
              width = .2,
              height = 0) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



As a Box Plot

We could also use geom_boxplot().

```
iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_boxplot() +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



As a Bar Plot with Error Bars

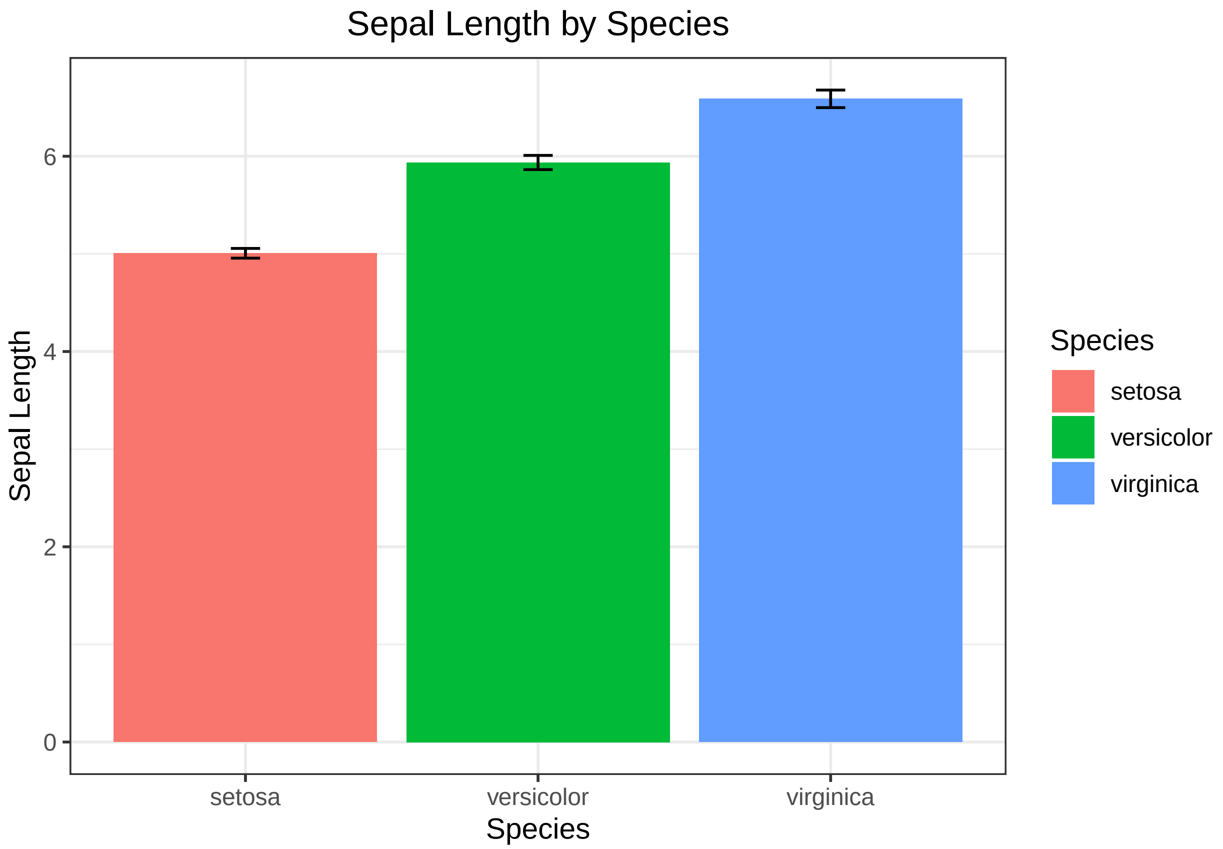
We could use geom_col() and geom_errorbar() to make a bar plot but this requires calculating the Species mean and standard error "by hand" using the group_by() and summarise() functions and then plotting with ggplot().

Note that we must specify "ymin" and "ymax" within the aes() of geom_errorbar(). These are the lower and upper bounds of the error bars. I also adjusted the width of the error bars by specifying "width = .1" within geom_errorbar() but outside of aes().

```
iris_2.0 <- iris %>%
  group_by(Species) %>%
  summarise(mean = mean(Sepal.Length),
            sd = sd(Sepal.Length),
            n = n(),
            se = sd/sqrt(n))
iris_2.0
```

```
## # A tibble: 3 x 5
##   Species     mean    sd     n     se
##   <fct>      <dbl> <dbl> <int>  <dbl>
## 1 setosa      5.01 0.352    50 0.0498
## 2 versicolor  5.94 0.516    50 0.0730
## 3 virginica   6.59 0.636    50 0.0899
```

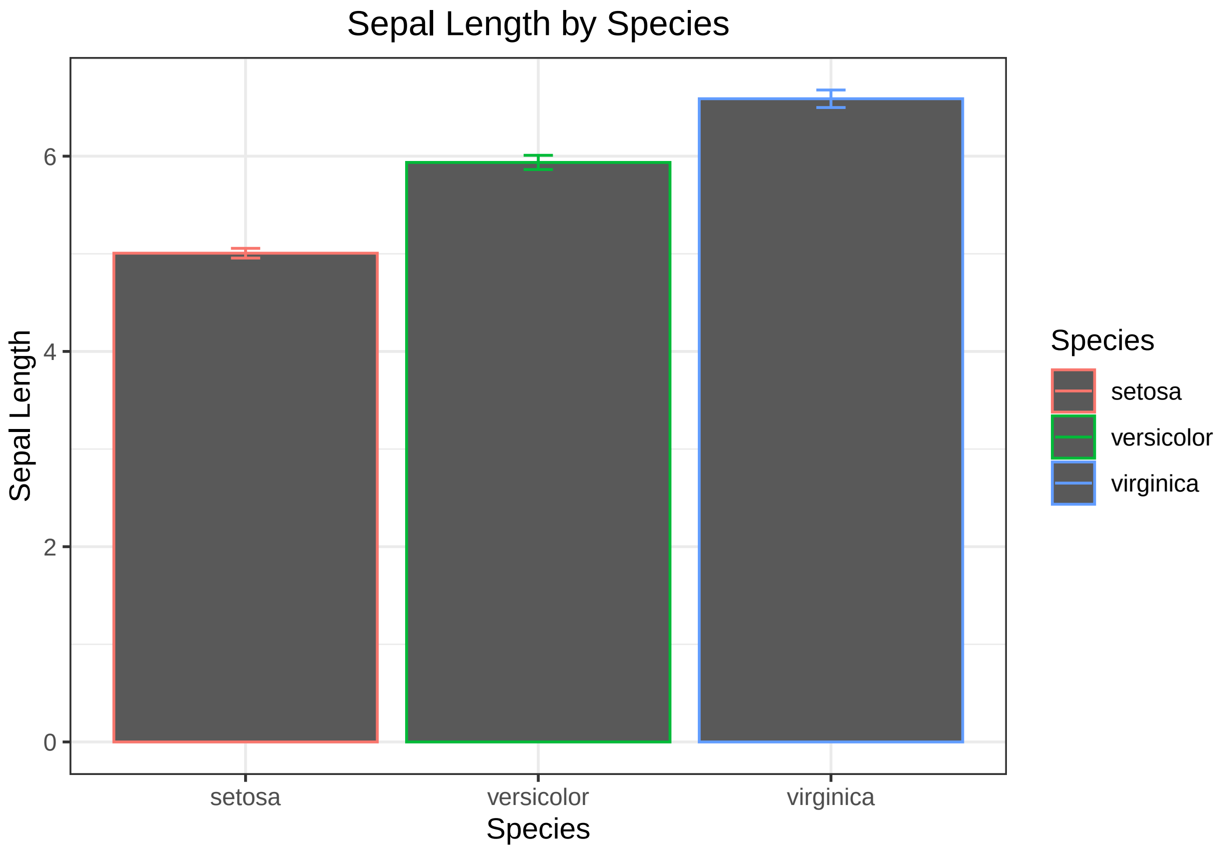
```
iris_2.0 %>%
  ggplot(aes(x = Species, y = mean, fill = Species)) +
  geom_col() +
  geom_errorbar(aes(ymax = mean + se, ymin = mean - se),
                width = .1) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



Specifying “fill” instead of “color” has a similar but different effect on the plot.

Notice here color relates to the outer boundary only.

```
iris %>%
  group_by(Species) %>%
  summarise(mean = mean(Sepal.Length),
            sd = sd(Sepal.Length),
            n = n(),
            se = sd/sqrt(n)) %>%
  ggplot(aes(x = Species, y = mean, color = Species)) +
  geom_col() +
  geom_errorbar(aes(ymax = mean + se, ymin = mean - se),
                width = .1) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



---

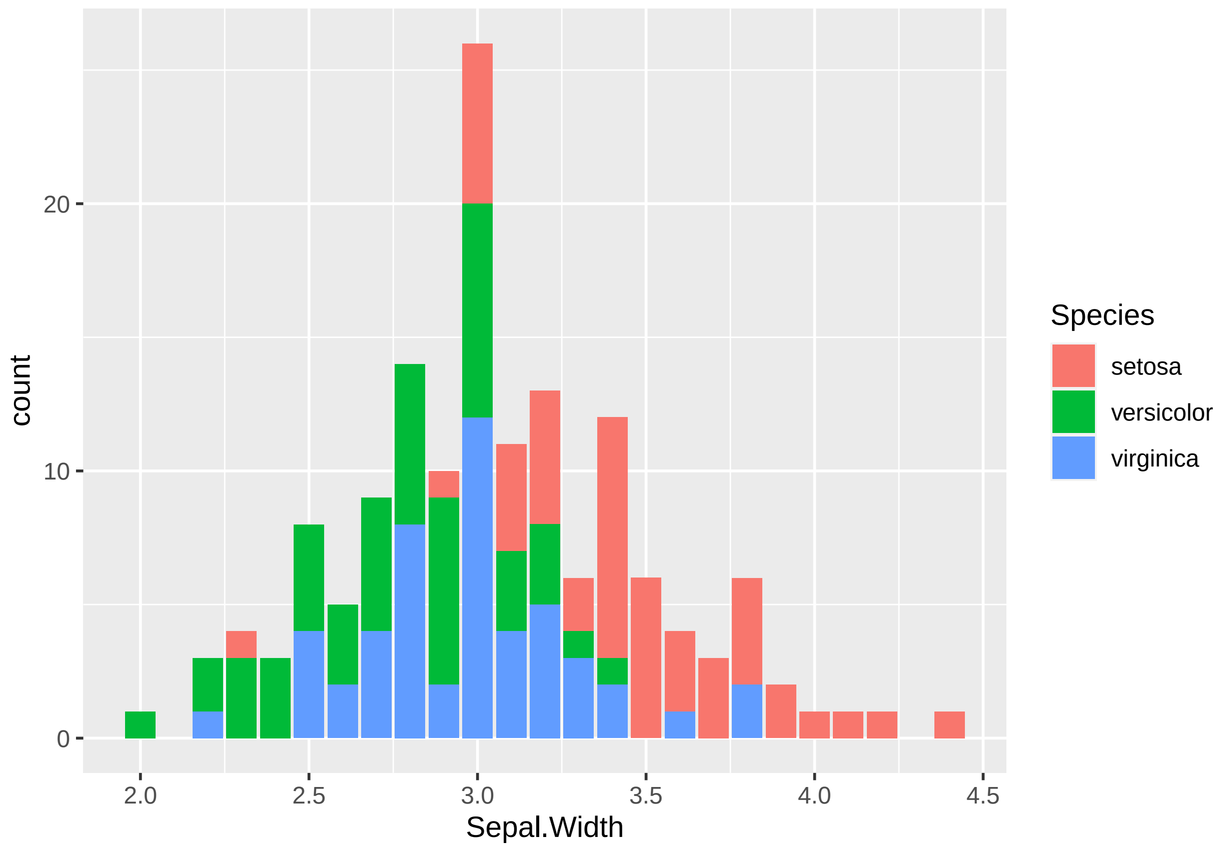
7. Positioning Data

What do we do if we have overlapping data?

There are two ways we can resolve this issue.

In bar plots, this is easily solved by position arguments. If we graph a count of how many leaves were a specific sepal length, the counts overlap between species.

```
####
#7#
####

iris %>%
  ggplot()+
  geom_bar(aes(x=Sepal.Width, fill=Species))
```

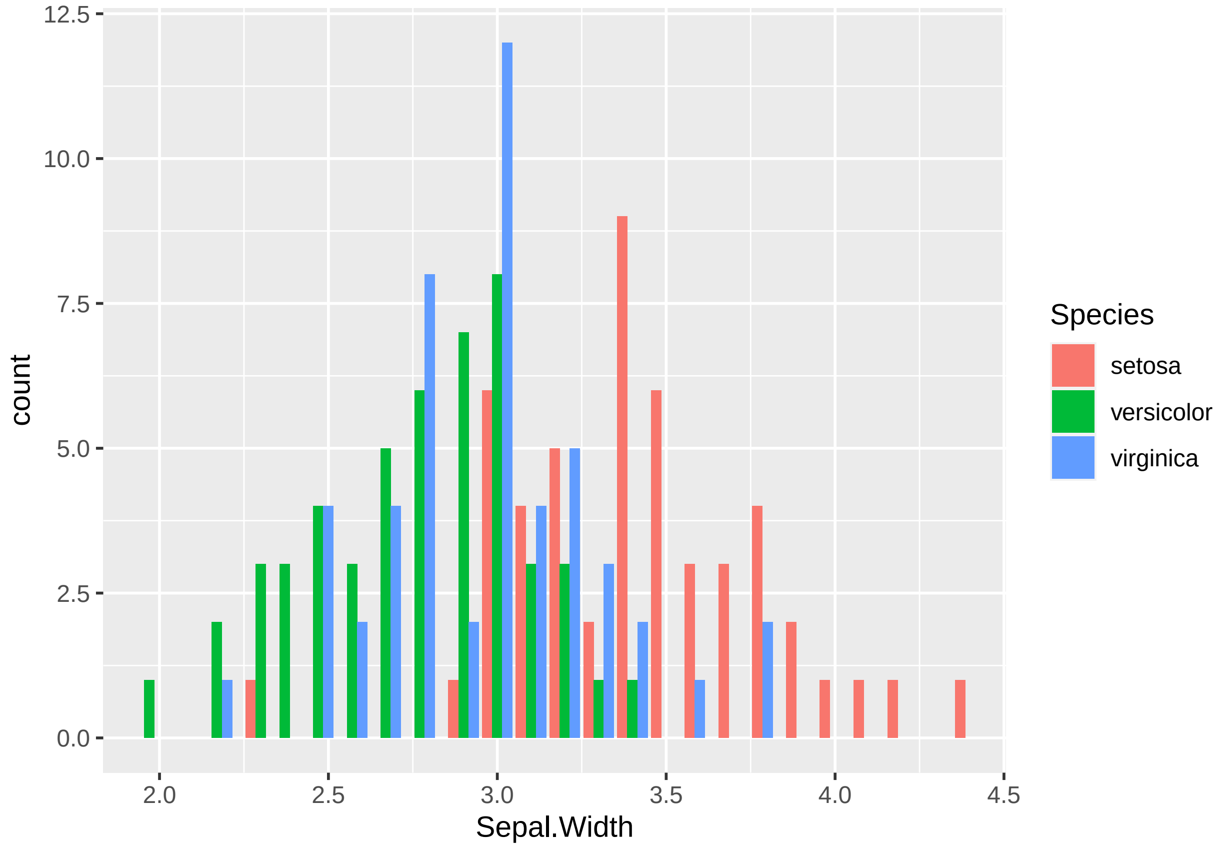


We can re-position the data using position_dodge.

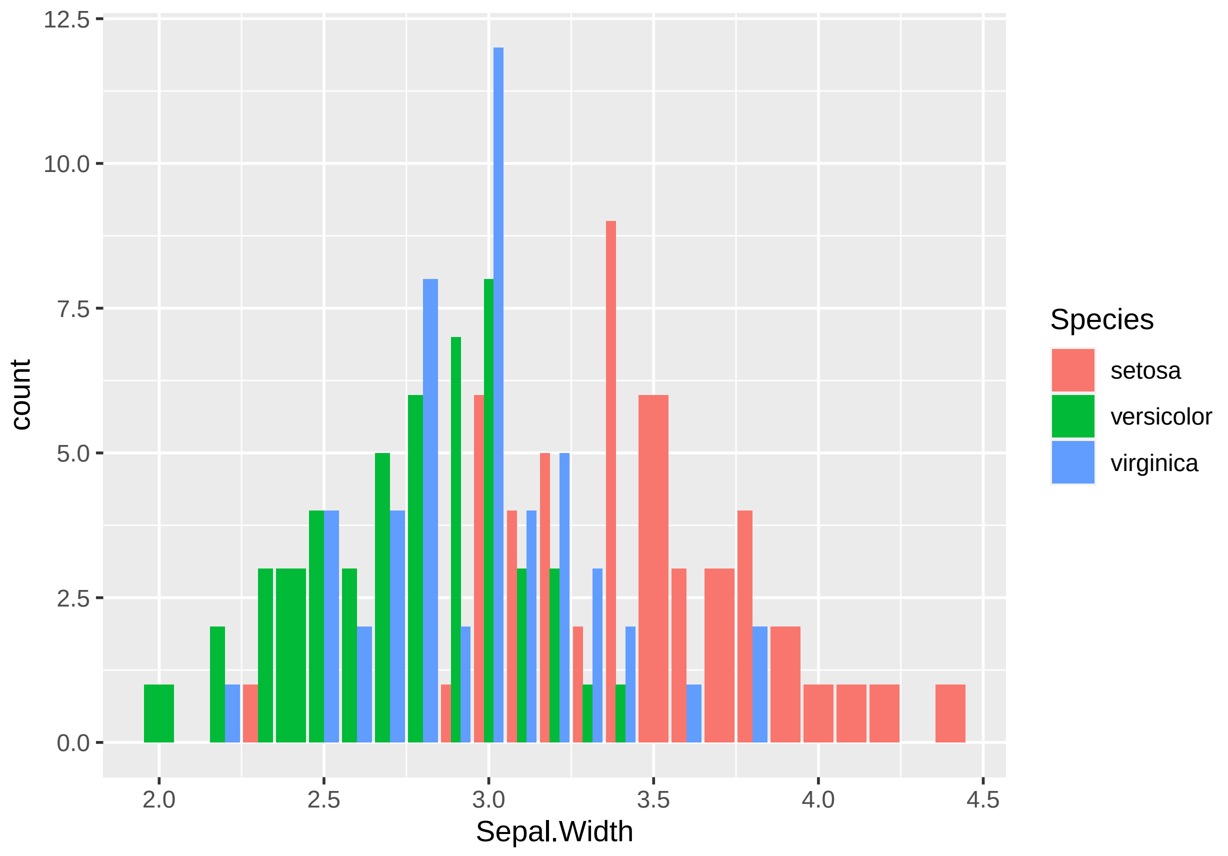
```
iris
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1             5.1         3.5          1.4         0.2     setosa
## 2             4.9         3.0          1.4         0.2     setosa
## 3             4.7         3.2          1.3         0.2     setosa
## 4             4.6         3.1          1.5         0.2     setosa
## 5             5.0         3.6          1.4         0.2     setosa
## 6             5.4         3.9          1.7         0.4     setosa
## 7             4.6         3.4          1.4         0.3     setosa
## 8             5.0         3.4          1.5         0.2     setosa
## 9             4.4         2.9          1.4         0.2     setosa
## 10            4.9         3.1          1.5         0.1     setosa
## 11            5.4         3.7          1.5         0.2     setosa
## 12            4.8         3.4          1.6         0.2     setosa
## 13            4.8         3.0          1.4         0.1     setosa
## 14            4.3         3.0          1.1         0.1     setosa
## 15            5.8         4.0          1.2         0.2     setosa
## 16            5.7         4.4          1.5         0.4     setosa
## 17            5.4         3.9          1.3         0.4     setosa
## 18            5.1         3.5          1.4         0.3     setosa
## 19            5.7         3.8          1.7         0.3     setosa
## 20            5.1         3.8          1.5         0.3     setosa
## 21            5.4         3.4          1.7         0.2     setosa
## 22           5.1         3.7          1.5         0.4     setosa
## 23           4.6         3.6          1.0         0.2     setosa
## 24           5.1         3.3          1.7         0.5     setosa
## 25           4.8         3.4          1.9         0.2     setosa
## 26           5.0         3.0          1.6         0.2     setosa
## 27           5.0         3.4          1.6         0.4     setosa
## 28           5.2         3.5          1.5         0.2     setosa
## 29           5.2         3.4          1.4         0.2     setosa
## 30           4.7         3.2          1.6         0.2     setosa
## 31           4.8         3.1          1.6         0.2     setosa
## 32           5.4         3.4          1.5         0.4     setosa
## 33           5.2         4.1          1.5         0.1     setosa
## 34           5.5         4.2          1.4         0.2     setosa
## 35           4.9         3.1          1.5         0.2     setosa
## 36           5.0         3.2          1.2         0.2     setosa
## 37           5.5         3.5          1.3         0.2     setosa
## 38           4.9         3.6          1.4         0.1     setosa
## 39           4.4         3.0          1.3         0.2     setosa
## 40           5.1         3.4          1.5         0.2     setosa
## 41           5.0         3.5          1.3         0.3     setosa
## 42           4.5         2.3          1.3         0.3     setosa
## 43           4.4         3.2          1.3         0.2     setosa
## 44           5.0         3.5          1.6         0.6     setosa
## 45           5.1         3.8          1.9         0.4     setosa
## 46           4.8         3.0          1.4         0.3     setosa
## 47           5.1         3.8          1.6         0.2     setosa
## 48           4.6         3.2          1.4         0.2     setosa
## 49           5.3         3.7          1.5         0.2     setosa
## 50           5.0         3.3          1.4         0.2     setosa
## 51           7.0         3.2          4.7         1.4 versicolor
## 52           6.4         3.2          4.5         1.5 versicolor
## 53           6.9         3.1          4.9         1.5 versicolor
## 54           5.5         2.3          4.0         1.3 versicolor
## 55           6.5         2.8          4.6         1.5 versicolor
## 56           5.7         2.8          4.5         1.3 versicolor
## 57           6.3         3.3          4.7         1.6 versicolor
## 58           4.9         2.4          3.3         1.0 versicolor
## 59           6.6         2.9          4.6         1.3 versicolor
## 60           5.2         2.7          3.9         1.4 versicolor
## 61           5.0         2.0          3.5         1.0 versicolor
## 62           5.9         3.0          4.2         1.5 versicolor
## 63           6.0         2.2          4.0         1.0 versicolor
## 64           6.1         2.9          4.7         1.4 versicolor
## 65           5.6         2.9          3.6         1.3 versicolor
## 66           6.7         3.1          4.4         1.4 versicolor
## 67           5.6         3.0          4.5         1.5 versicolor
## 68           5.8         2.7          4.1         1.0 versicolor
## 69           6.2         2.2          4.5         1.5 versicolor
## 70           5.6         2.5          3.9         1.1 versicolor
## 71           5.9         3.2          4.8         1.8 versicolor
## 72           6.1         2.8          4.0         1.3 versicolor
## 73           6.3         2.5          4.9         1.5 versicolor
## 74           6.1         2.8          4.7         1.2 versicolor
## 75           6.4         2.9          4.3         1.3 versicolor
## 76           6.6         3.0          4.4         1.4 versicolor
## 77           6.8         2.8          4.8         1.4 versicolor
## 78           6.7         3.0          5.0         1.7 versicolor
## 79           6.0         2.9          4.5         1.5 versicolor
## 80           5.7         2.6          3.5         1.0 versicolor
## 81           5.5         2.4          3.8         1.1 versicolor
## 82           5.5         2.4          3.7         1.0 versicolor
## 83           5.8         2.7          3.9         1.2 versicolor
## 84           6.0         2.7          5.1         1.6 versicolor
## 85           5.4         3.0          4.5         1.5 versicolor
## 86           6.0         3.4          4.5         1.6 versicolor
## 87           6.7         3.1          4.7         1.5 versicolor
## 88           6.3         2.3          4.4         1.3 versicolor
## 89           5.6         3.0          4.1         1.3 versicolor
## 90           5.5         2.5          4.0         1.3 versicolor
## 91           5.5         2.6          4.4         1.2 versicolor
## 92           6.1         3.0          4.6         1.4 versicolor
## 93           5.8         2.6          4.0         1.2 versicolor
## 94           5.0         2.3          3.3         1.0 versicolor
## 95           5.6         2.7          4.2         1.3 versicolor
## 96           5.7         3.0          4.2         1.2 versicolor
## 97           5.7         2.9          4.2         1.3 versicolor
## 98           6.2         2.9          4.3         1.3 versicolor
## 99           5.1         2.5          3.0         1.1 versicolor
## 100          5.7         2.8          4.1         1.3 versicolor
## 101          6.3         3.3          6.0         2.5  virginica
## 102          5.8         2.7          5.1         1.9  virginica
## 103          7.1         3.0          5.9         2.1  virginica
## 104          6.3         2.9          5.6         1.8  virginica
## 105          6.5         3.0          5.8         2.2  virginica
## 106          7.6         3.0          6.6         2.1  virginica
## 107          4.9         2.5          4.5         1.7  virginica
## 108          7.3         2.9          6.3         1.8  virginica
## 109          6.7         2.5          5.8         1.8  virginica
## 110          7.2         3.6          6.1         2.5  virginica
## 111          6.5         3.2          5.1         2.0  virginica
## 112          6.4         2.7          5.3         1.9  virginica
## 113          6.8         3.0          5.5         2.1  virginica
## 114          5.7         2.5          5.0         2.0  virginica
## 115          5.8         2.8          5.1         2.4  virginica
## 116          6.4         3.2          5.3         2.3  virginica
## 117          6.5         3.0          5.5         1.8  virginica
## 118          7.7         3.8          6.7         2.2  virginica
## 119          7.7         2.6          6.9         2.3  virginica
## 120          6.0         2.2          5.0         1.5  virginica
## 121          6.9         3.2          5.7         2.3  virginica
## 122          5.6         2.8          4.9         2.0  virginica
## 123          7.7         2.8          6.7         2.0  virginica
## 124          6.3         2.7          4.9         1.8  virginica
## 125          6.7         3.3          5.7         2.1  virginica
## 126          7.2         3.2          6.0         1.8  virginica
## 127          6.2         2.8          4.8         1.8  virginica
## 128          6.1         3.0          4.9         1.8  virginica
## 129          6.4         2.8          5.6         2.1  virginica
## 130          7.2          3.0          5.8          1.6  virginica
## 131          7.4          2.8          6.1          1.9  virginica
## 132          7.9          3.8          6.4          2.0  virginica
## 133          6.4          2.8          5.6          2.2  virginica
## 134          6.3          2.8          5.1          1.5  virginica
## 135          6.1          2.6          5.6          1.4  virginica
## 136          7.7          3.0          6.1          2.3  virginica
## 137          6.3          3.4          5.6          2.4  virginica
## 138          6.4          3.1          5.5          1.8  virginica
## 139          6.0          3.0          4.8          1.8  virginica
## 140          6.9          3.1          5.4          2.1  virginica
## 141          6.7          3.1          5.6          2.4  virginica
## 142          6.9          3.1          5.1          2.3  virginica
## 143          5.8          2.7          5.1          1.9  virginica
## 144          6.8          3.2          5.9          2.3  virginica
## 145          6.7          3.3          5.7          2.5  virginica
## 146          6.7          3.0          5.2          2.3  virginica
## 147          6.3          2.5          5.0          1.9  virginica
## 148          6.5          3.0          5.2          2.0  virginica
## 149          6.2          3.4          5.4          2.3  virginica
## 150          5.9          3.0          5.1          1.8  virginica
```

```
iris %>%
  ggplot(aes(x=Sepal.Width, fill=Species))+
  geom_bar( position=position_dodge(preserve='single'))
```



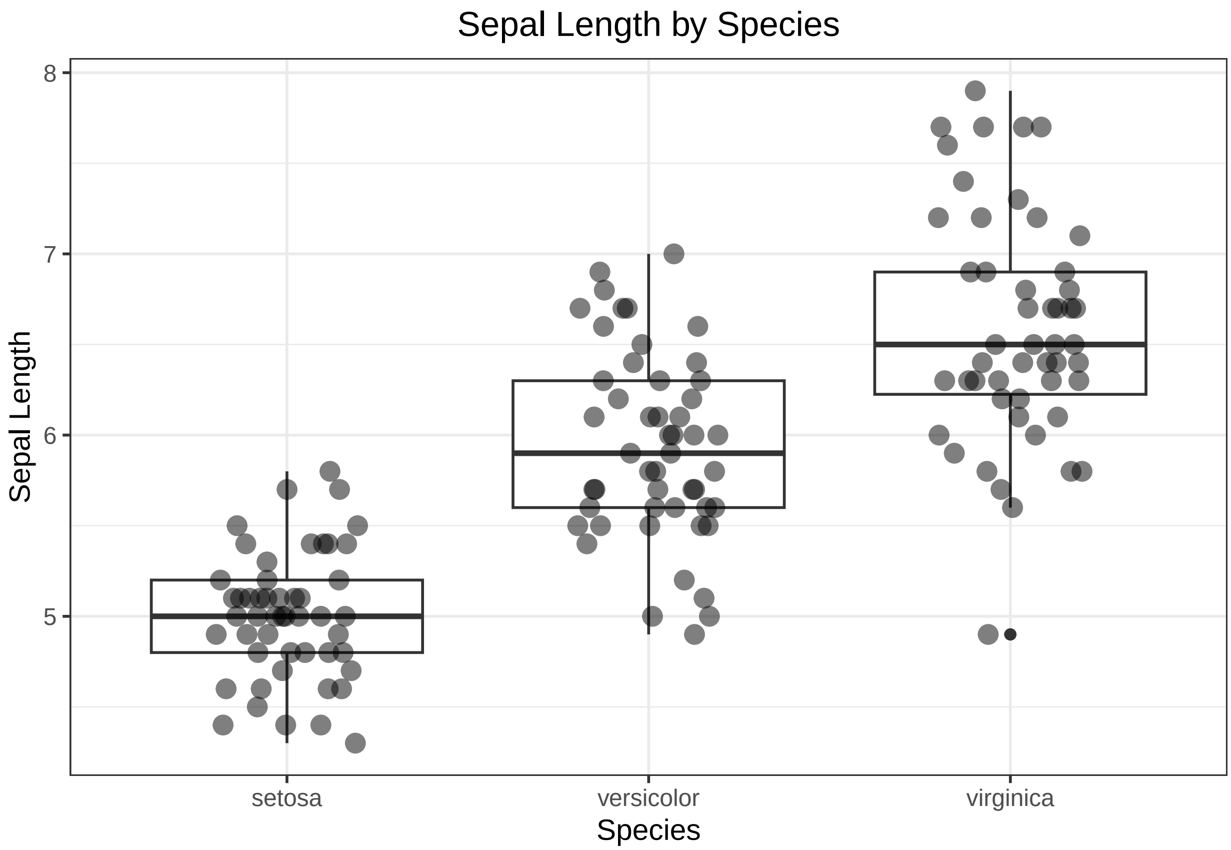
```
iris %>%
  ggplot(aes(x=Sepal.Width, fill=Species))+
  geom_bar( position=position_dodge(preserve='total'))
```



8. Overlaying Geoms: Boxplot with Points Overlain

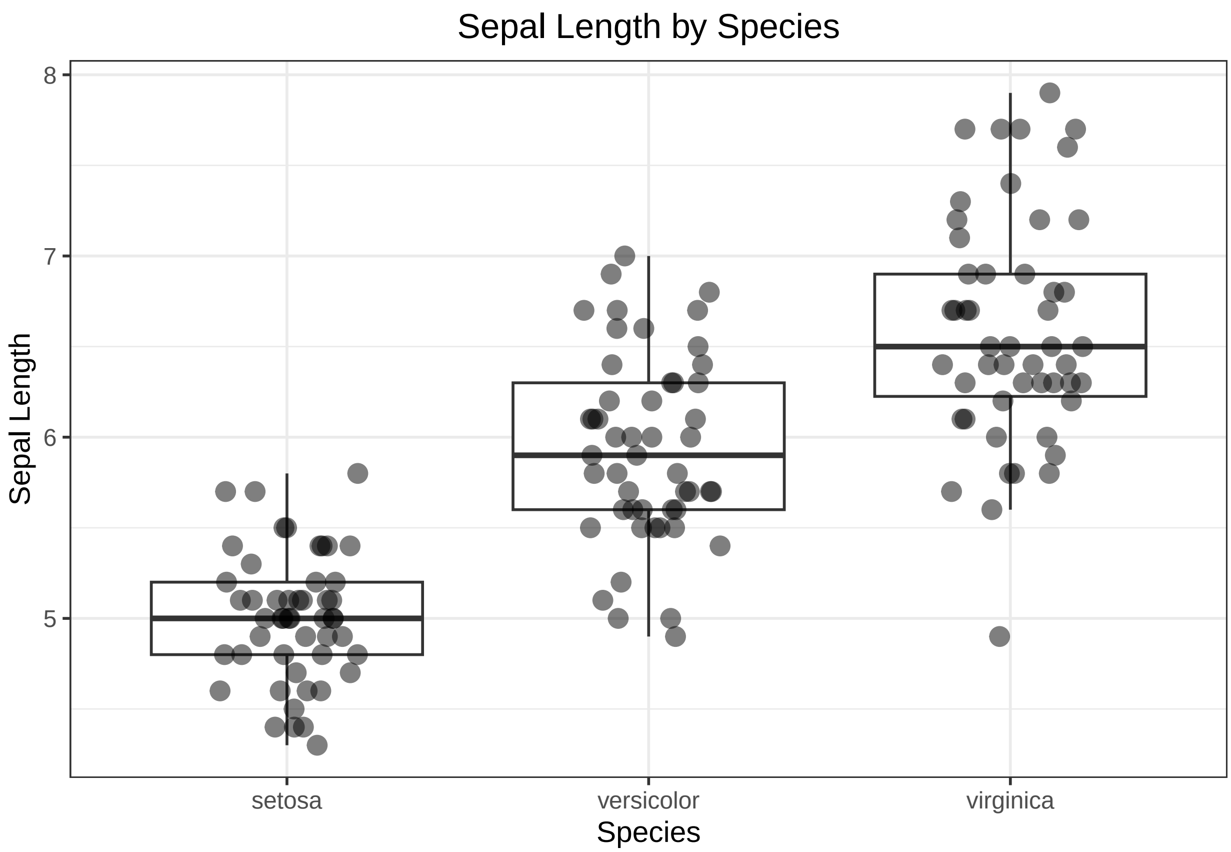
You can layer multiple geometries simply by including them in the same ggplot. Adding points over a boxplot is common form of this. However, because geom_boxplot() plots outliers as points geom_jitter plots all points, some observations are plotted twice. This problem can be seen above for the vriginica species.

```
####
#8#
####
iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_boxplot() +
  geom_jitter(width = .2,
              height = 0,
              alpha = .5,
              size = 3) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



To correct this, we can specify “outlier.shape = NA” withing geom_boxplot() as seen below.

```
iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(size = 3,
              width = .2,
              height = 0,
              alpha = .5) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```

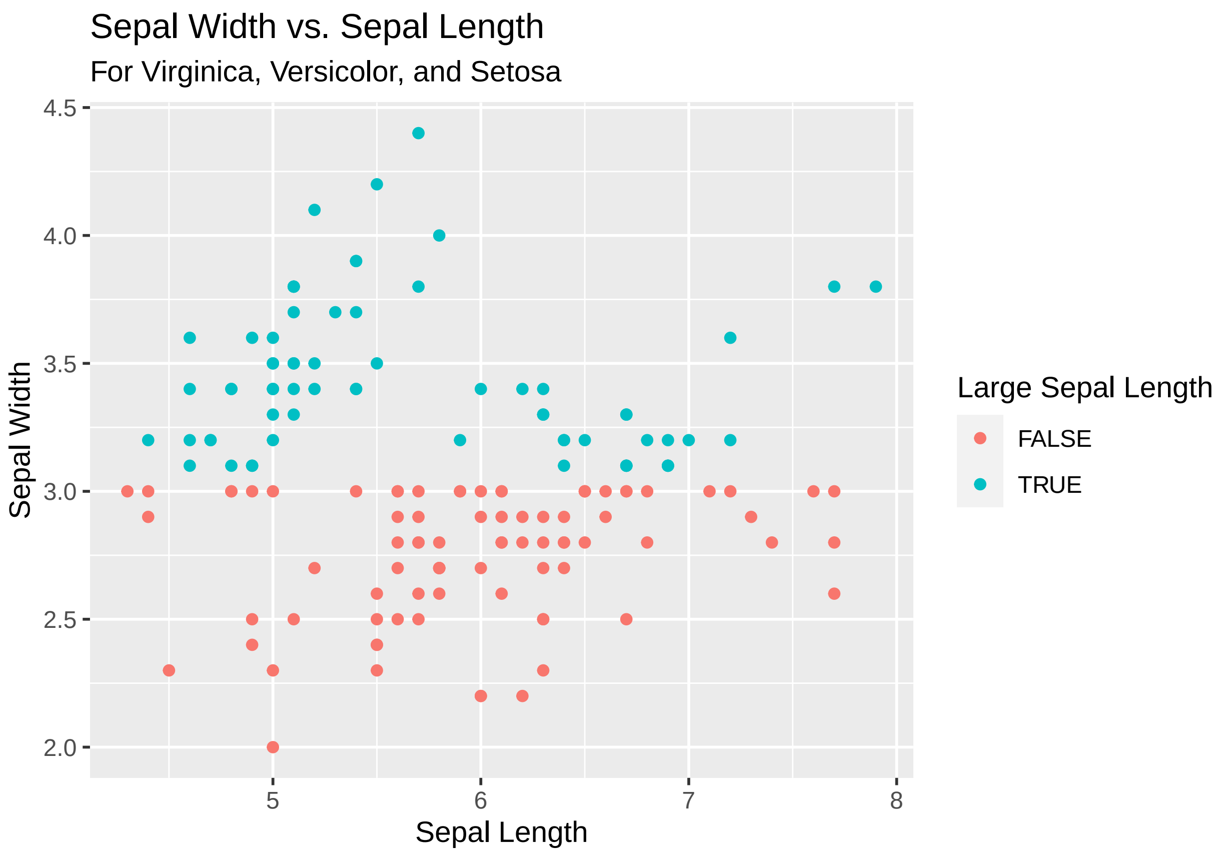


9. Adding Labels and a Title

Adding the labs() function with "+" lets us give the plot x and y labels as well as a title and subtitle

You can also change the label for any component that has a built in legend.

```
####
#09#
####
iris %>%
  ggplot() +
  geom_point(aes(x = Sepal.Length, y = Sepal.Width, color=Sepal.Width > 3)) +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length",
       subtitle= "For Virginica, Versicolor, and Setosa",
       color= "Large Sepal Length")
```

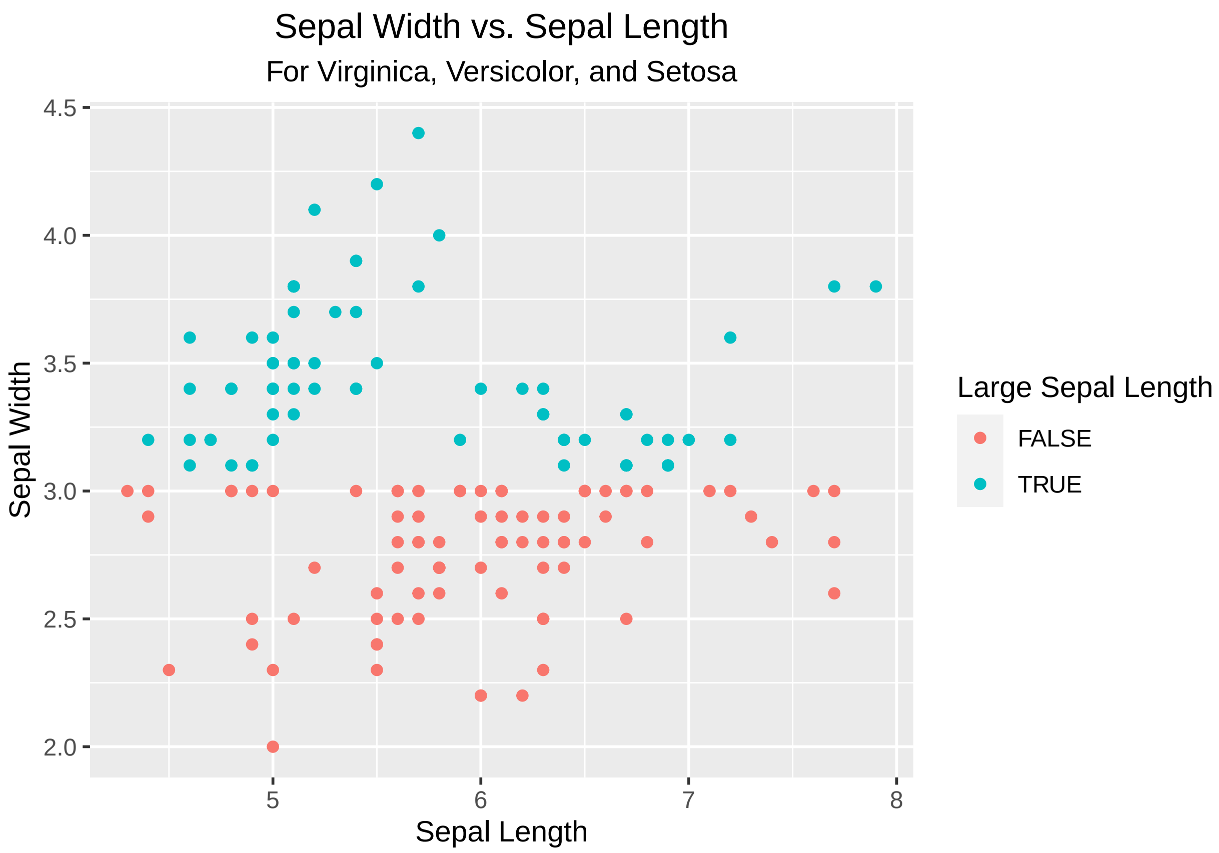


---

10. Centering the Title

Many parameters can be adjusted in the theme function. Overall, its fairly unintuitive but typically, someone has done what you would like to do and posted the solution on the internet so it's worth looking it up. I never remember how to center the title so I always have to look it up and copy the code.

```
####
#10#
####
iris %>%
  ggplot() +
  geom_point(aes(x = Sepal.Length, y = Sepal.Width, color=Sepal.Width > 3)) +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length",
       subtitle= "For Virginica, Versicolor, and Setosa",
       color= "Large Sepal Length") +
  theme(plot.title = element_text(hjust = 0.5), plot.subtitle = element_text(hjust = 0.5))
```



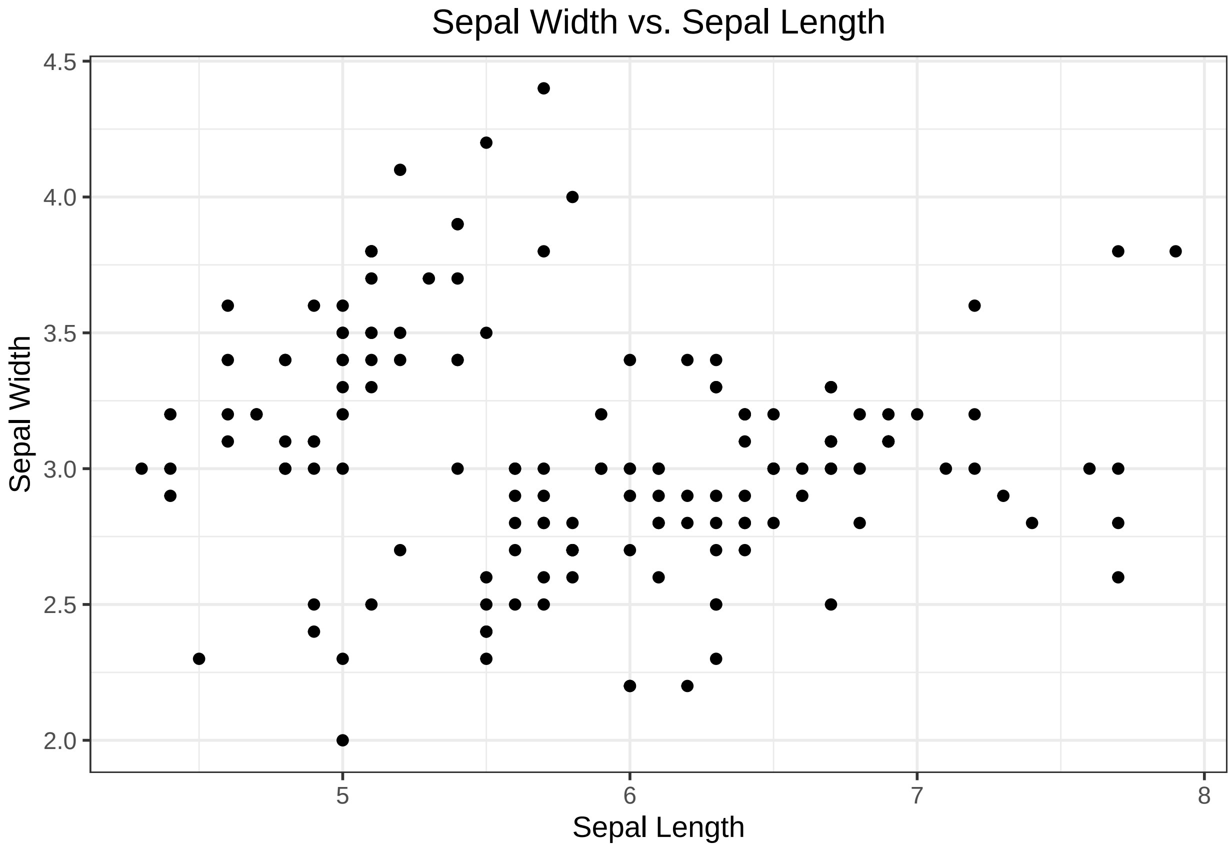
---

11. Choosing a Theme

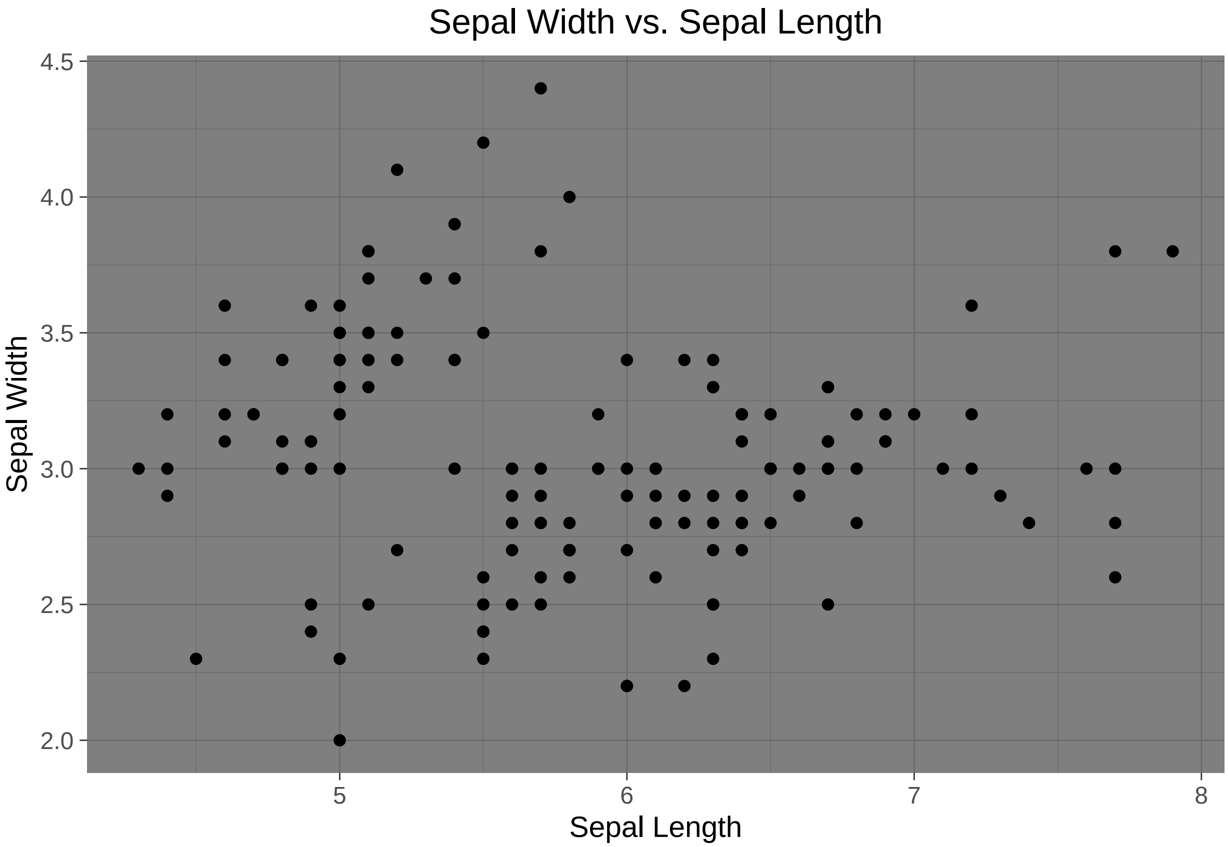
There are many themes in ggplot(). The theme can be added using "+" and specifying a theme, such as "theme_bw()". Here are 4 examples. Note that the "theme()" function comes after "theme_bw()".

The 4th example shows that it is a lot more than just the backdrop.

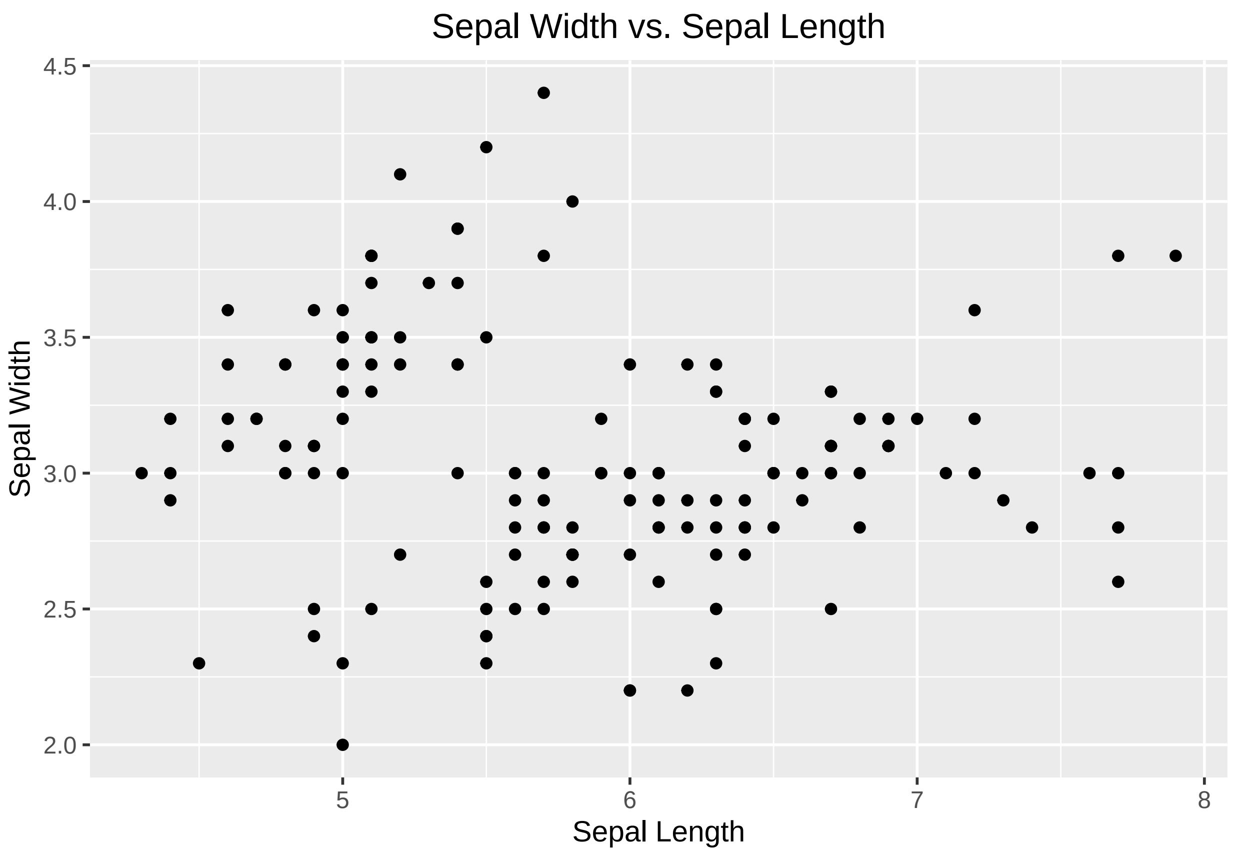
```
####
#11#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



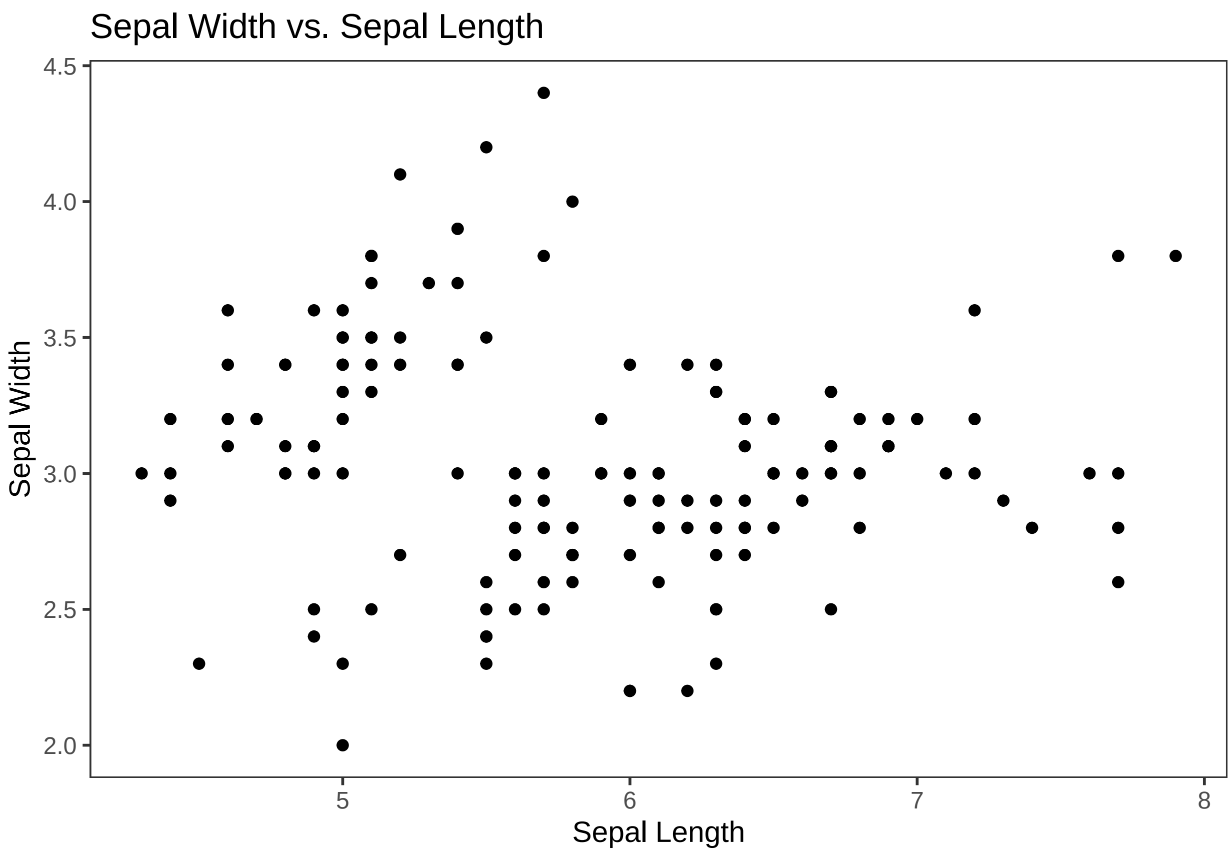
```r
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_dark() +
  theme(plot.title = element_text(hjust = 0.5))
```



```
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_gray() +
  theme(plot.title = element_text(hjust = 0.5))
```



```r
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_test()
```



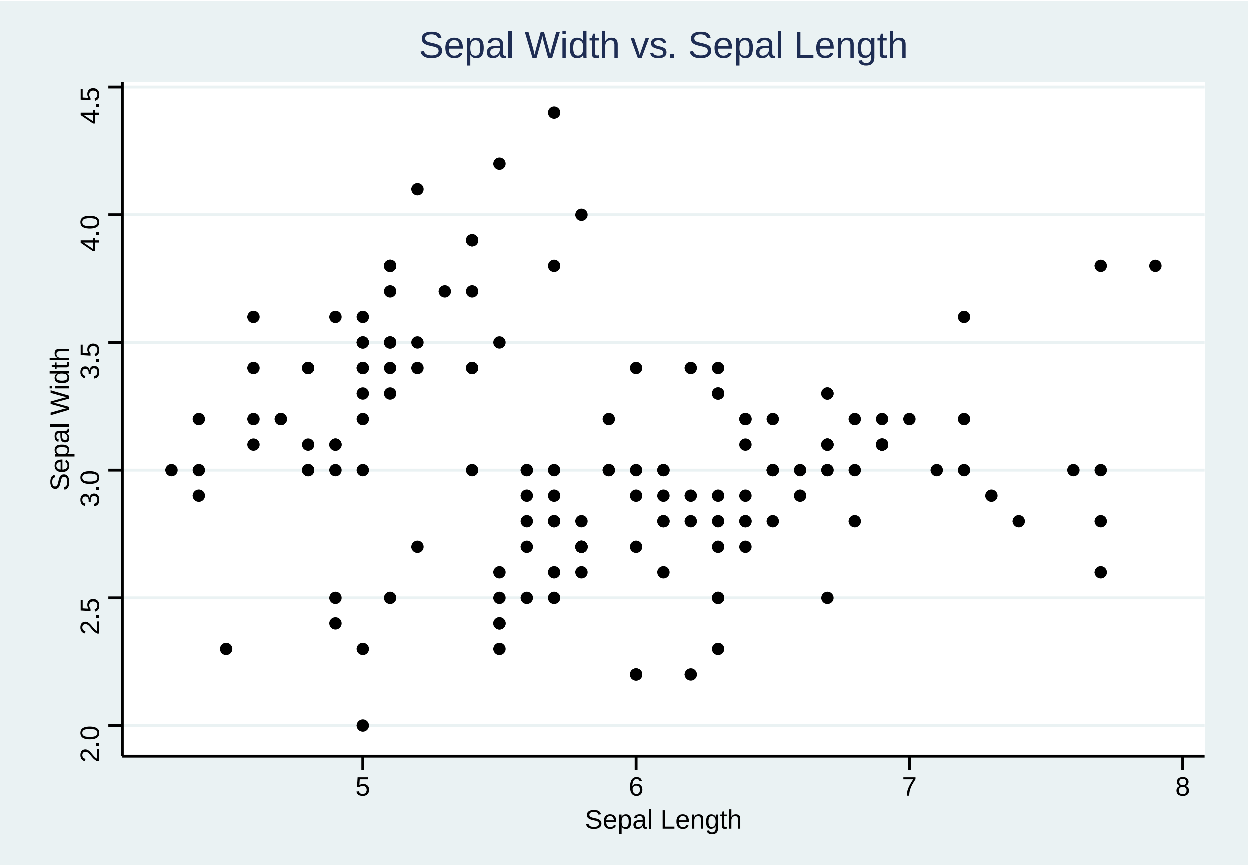
# 11b. More Themes!

There are many many many more themes available than just those listed.

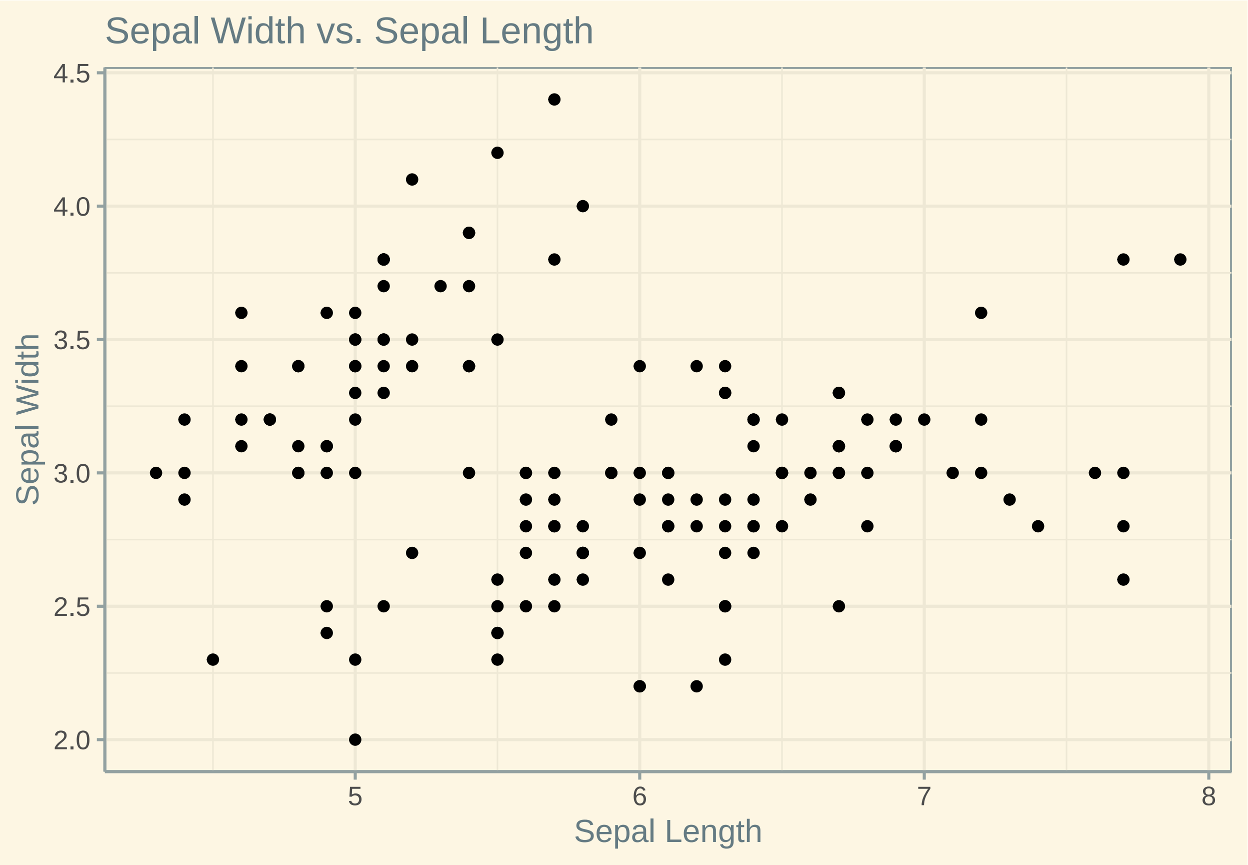
ggthemes has a ton more options

It goes beyond the scope of the class, but one can even generate their own themes.

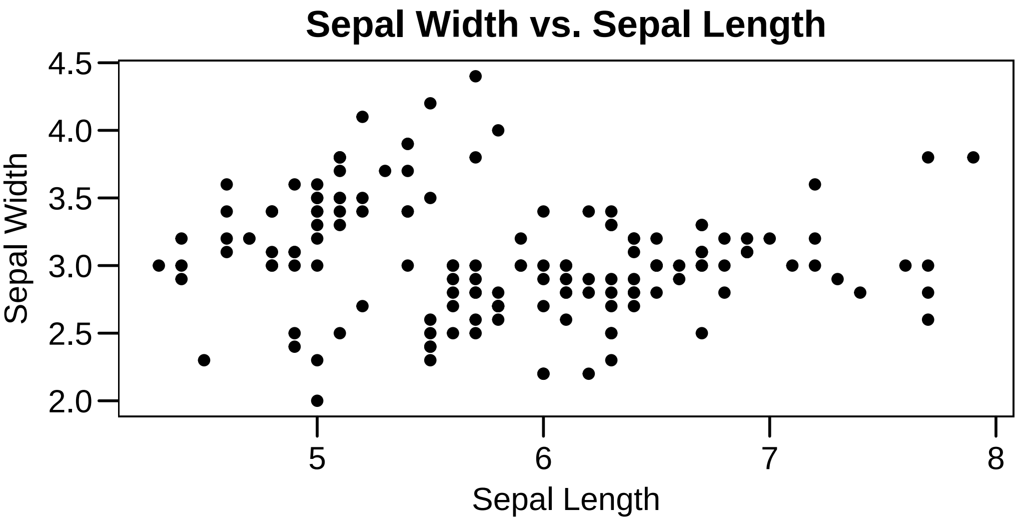
```
####
#11b#
####

#install.packages("ggthemes")
library(ggthemes)


iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_stata()
```



```
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_solarized()
```



```r
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_par()
```



12. Scales and Coordinates

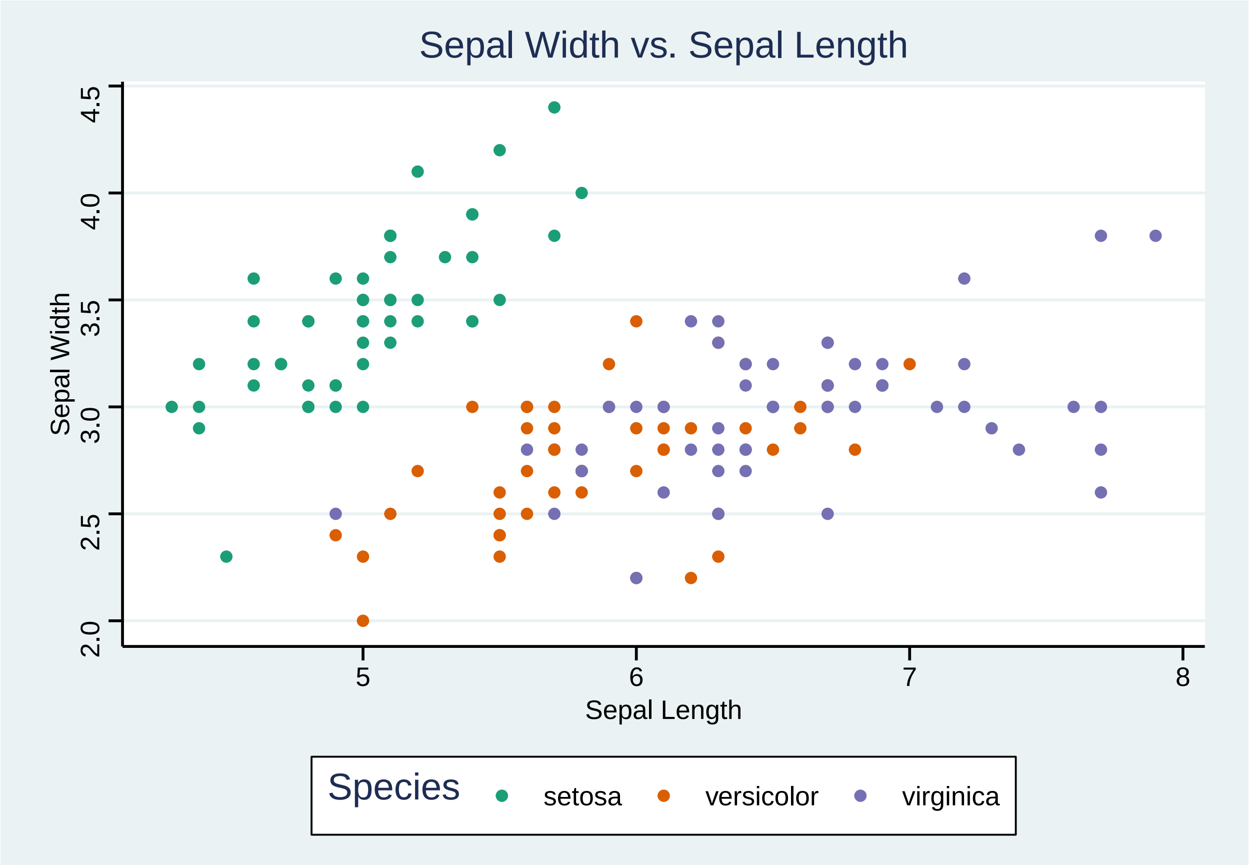
Scales

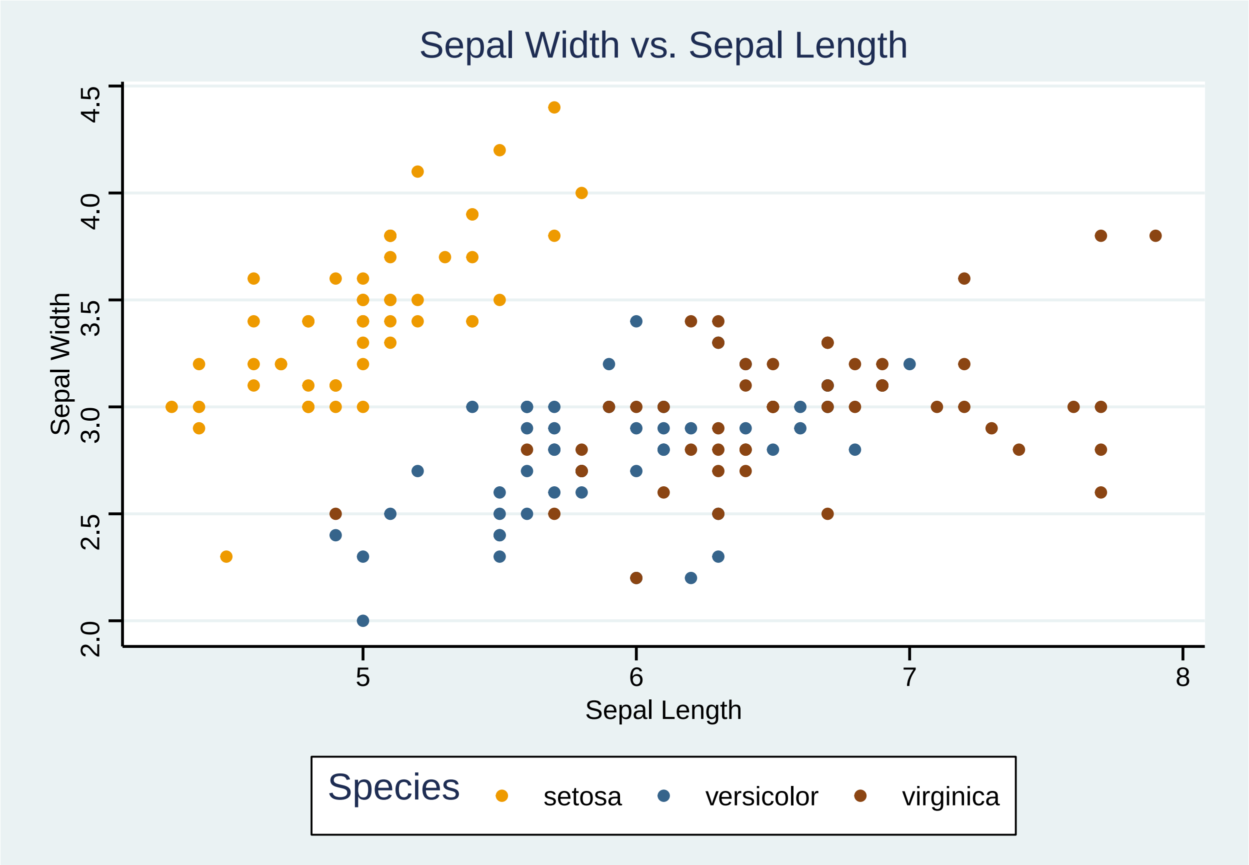
Scales allow us to define our mapping.

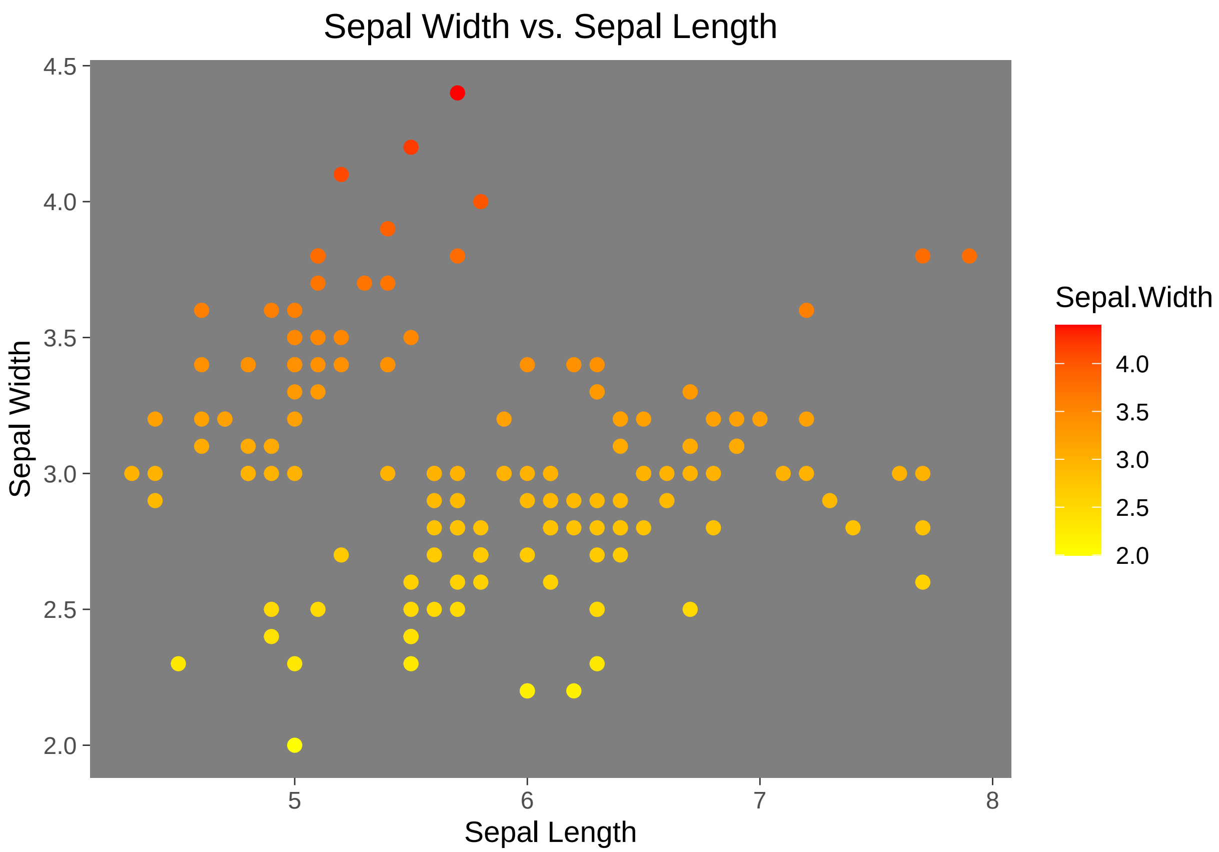
We can use scale_color_gradient or scale_color_brewer to take a continuous variable and make a color scheme that is both easy and pleasant on the eyes.

However, scale_color_brewer is only for categorical variables, not continuous ones.

You can also use scale_color_manual to make your own colorset.


Sepal Width vs. Sepal Length
Sepal Width
2.0
2.5
3.0
3.5
4.0
4.5
Sepal Length
5
6
7
8
Species
● setosa
● versicolor
● virginica


Sepal Width vs. Sepal Length
Sepal Width
2.0
2.5
3.0
3.5
4.0
4.5
5
6
7
8
Sepal Length
Species
● setosa
● versicolor
● virginica

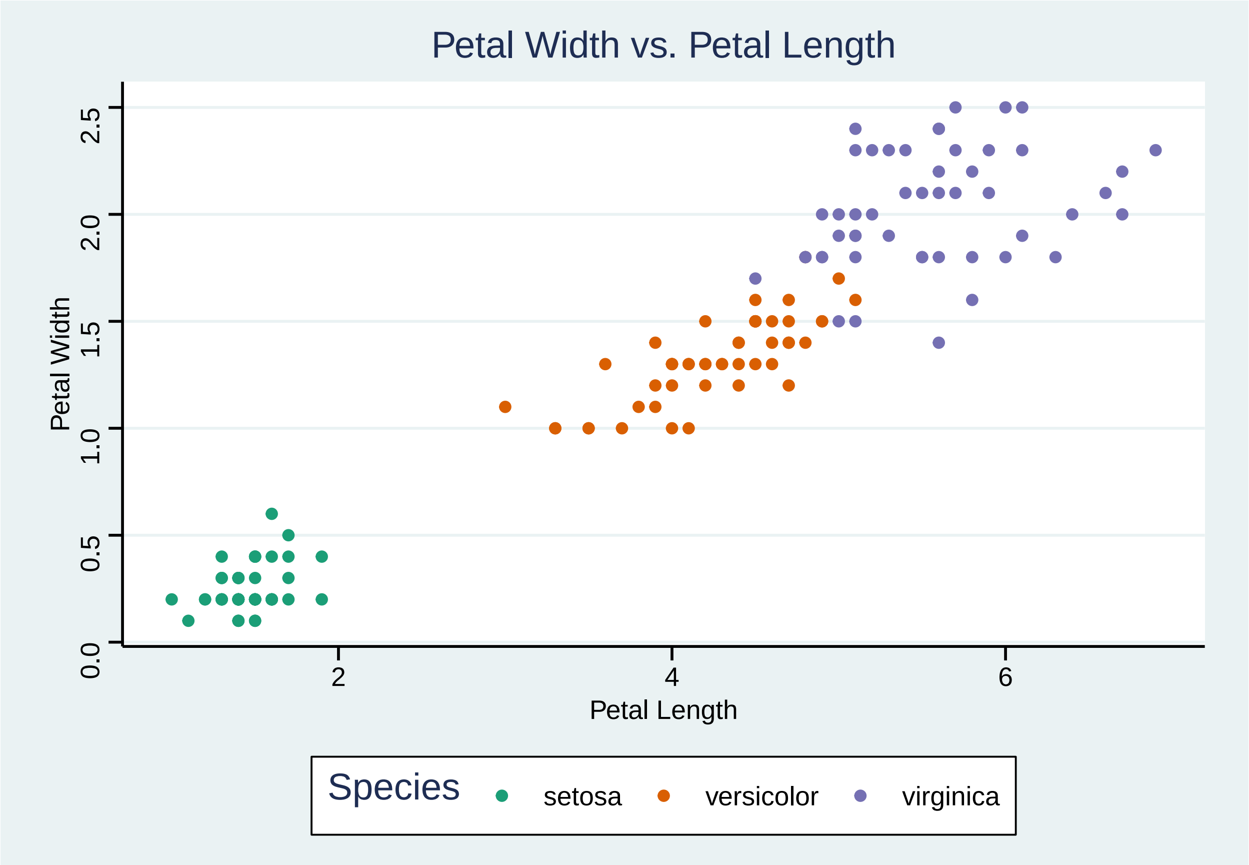


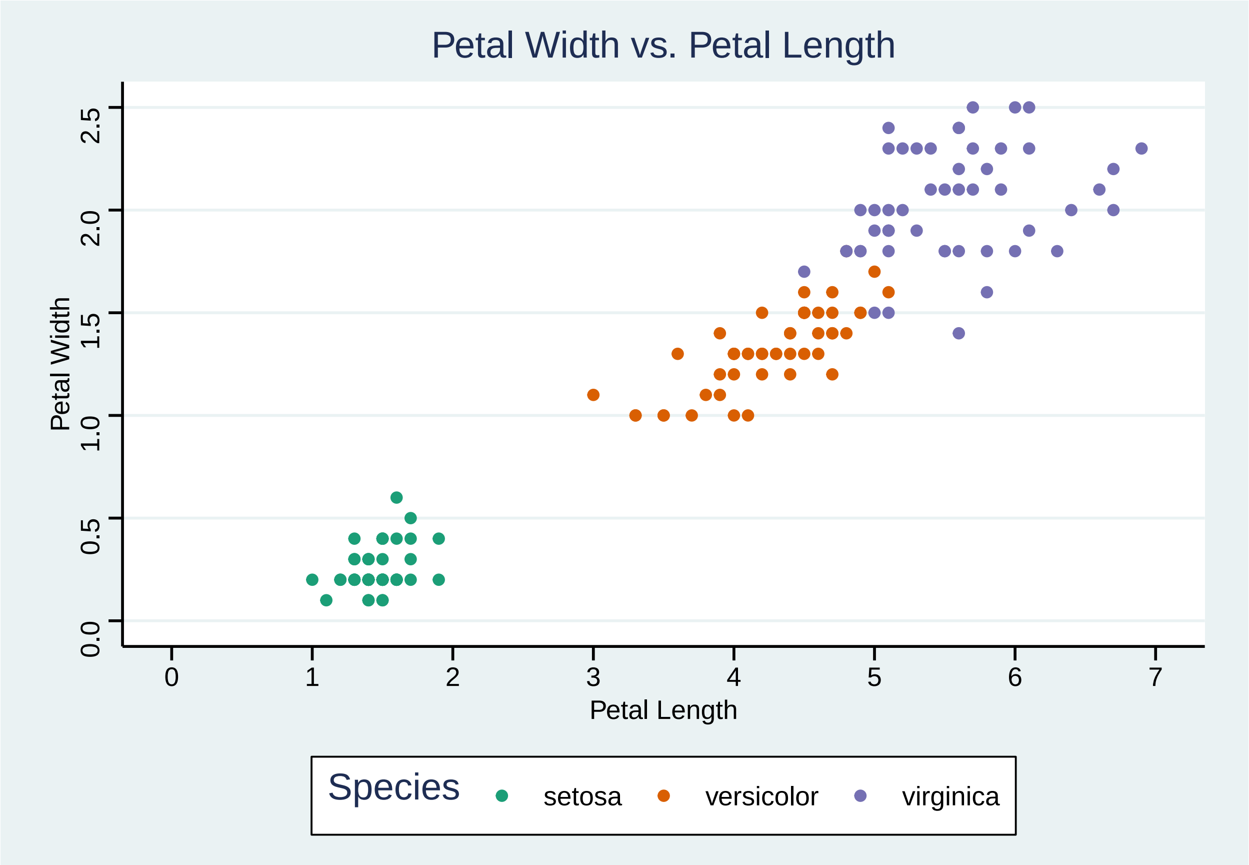
12b. Scales for your axis

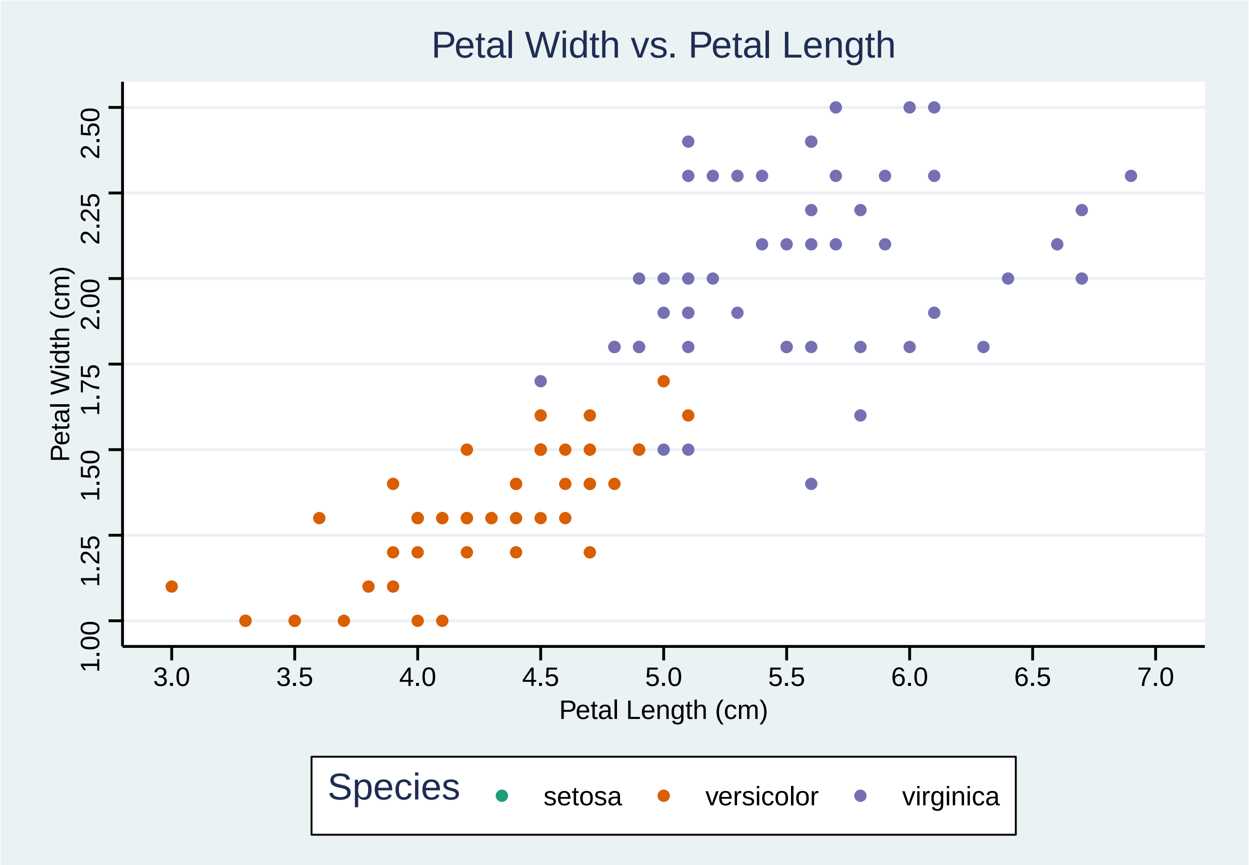
Scales

You can also use scales to modify your axis. If we graph petal length and petal width, the standard x axis looks very sparse. What if we don't care about the setosa data and are more interested in the spread of versicolor and virginica data?

We can use scales to zoom in on that data and provided more detailed tick marks to tease out the details.


Petal Width vs. Petal Length
Petal Width
0.0
0.5
1.0
1.5
2.0
2.5
2
4
6
Petal Length
Species
● setosa
● versicolor
● virginica


Petal Width vs. Petal Length
Petal Width
0.0
0.5
1.0
1.5
2.0
2.5
0
1
2
3
4
5
6
7
Petal Length
Species
● setosa
● versicolor
● virginica



Appendix

```
library(datasets)
library(tidyverse)
library(knitr)
library(ggplot2)

####
#01#
####
plot(c(1,2,3,4,5),c(1,4,9,16,25), xlab="X-Axis", ylab="Y-Axis")


####
#02#
####
data()

####
#02b#
####
```

```r
?iris

head(iris)
summary(iris)


####
#02c#
####
iris %>%
  ggplot()

####
#03#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width))

####
#04#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point()

####
#04b#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point(color='steelblue1', size=4, alpha=0.5, shape="square")

####
#04c#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, color=Sepal.Width > 3, size=Sepal.Width > 3, shape= Sepa
  geom_point(alpha=0.5)

####
#04d#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, color=Sepal.Width > 3, size=Sepal.Width > 3, shape= Sepa
  geom_point(alpha=0.5, color="chocolate3", shape="square", size=2)

####
#08#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, col = Sepal.Length)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

####
#05b#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, shape = Species)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, color = Species, shape = Species)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
####
#6#
####
iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_point(size=3) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_point(size = 3,
             alpha = .5) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))


iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_jitter(size = 3,
              alpha = .5) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))


iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_jitter(size = 3,
              alpha = .5,
              width = .2,
              height = 0) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))


iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_boxplot() +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))


iris_2.0 <- iris %>%
  group_by(Species) %>%
  summarise(mean = mean(Sepal.Length),
            sd = sd(Sepal.Length),
            n = n(),
            se = sd/sqrt(n))
iris_2.0

iris_2.0 %>%
  ggplot(aes(x = Species, y = mean, fill = Species)) +
  geom_col() +
  geom_errorbar(aes(ymax = mean + se, ymin = mean - se),
                width = .1) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

iris %>%
  group_by(Species) %>%
  summarise(mean = mean(Sepal.Length),
            sd = sd(Sepal.Length),
            n = n(),
            se = sd/sqrt(n)) %>%
  ggplot(aes(x = Species, y = mean, color = Species)) +
  geom_col() +
  geom_errorbar(aes(ymax = mean + se, ymin = mean - se),
                width = .1) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))


####
#7#
####

iris %>%
  ggplot()+
  geom_bar(aes(x=Sepal.Width, fill=Species))


iris
iris %>%
  ggplot(aes(x=Sepal.Width, fill=Species))+
  geom_bar( position=position_dodge(preserve='single'))

iris %>%
  ggplot(aes(x=Sepal.Width, fill=Species))+
  geom_bar( position=position_dodge(preserve='total'))

####
#8#
####
iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_boxplot() +
  geom_jitter(width = .2,
              height = 0,
              alpha = .5,
              size = 3) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))


iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(size = 3,
              width = .2,
              height = 0,
              alpha = .5) +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

####
#09#
####
iris %>%
  ggplot() +
  geom_point(aes(x = Sepal.Length, y = Sepal.Width, color=Sepal.Width > 3)) +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length",
       subtitle= "For Virginica, Versicolor, and Setosa",
       color= "Large Sepal Length")

####
#10#
####
iris %>%
  ggplot() +
  geom_point(aes(x = Sepal.Length, y = Sepal.Width, color=Sepal.Width > 3)) +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length",
       subtitle= "For Virginica, Versicolor, and Setosa",
       color= "Large Sepal Length") +
  theme(plot.title = element_text(hjust = 0.5), plot.subtitle = element_text(hjust = 0.5))

####
#11#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))

iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_dark() +
  theme(plot.title = element_text(hjust = 0.5))

iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_gray() +
  theme(plot.title = element_text(hjust = 0.5))
```

```r
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_test()

####
#11b#
####

#install.packages("ggthemes")
library(ggthemes)


iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_stata()

iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_solarized()

iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_par()
####
#12#
####

iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, color=Species)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_stata() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_brewer("Species", palette="Dark2")

iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, color=Species)) +
  geom_point() +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_stata() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_manual(breaks=c("setosa", "versicolor", "virginica"), values=c("orange2", "steelblue4", "

iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, color=Sepal.Width)) +
  geom_point(size=2) +
  labs(x = "Sepal Length", y = "Sepal Width",
       title = "Sepal Width vs. Sepal Length") +
  theme_dark() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_gradient(low = "yellow", high = "red")
####
#12b#
####

iris %>%
  ggplot(aes(x = Petal.Length, y = Petal.Width, color=Species)) +
  geom_point() +
  labs(x = "Petal Length", y = "Petal Width",
       title = "Petal Width vs. Petal Length") +
  theme_stata() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_brewer("Species", palette="Dark2")

iris %>%
  ggplot(aes(x = Petal.Length, y = Petal.Width, color=Species)) +
  geom_point() +
  labs(x = "Petal Length", y = "Petal Width",
       title = "Petal Width vs. Petal Length") +
  theme_stata() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_brewer("Species", palette="Dark2") +
  scale_x_continuous (breaks=c(0, 1, 2, 3, 4, 5, 6, 7), limits = c(0,7)) +
  scale_y_continuous(limits=c(0,2.5))

iris %>%
  ggplot(aes(x = Petal.Length, y = Petal.Width, color=Species)) +
  geom_point() +
  labs(x = "Petal Length", y = "Petal Width",
       title = "Petal Width vs. Petal Length") +
  theme_stata() +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_color_brewer("Species", palette="Dark2") +
  scale_x_continuous (name= "Petal Length (cm)", breaks=c(3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0),
  scale_y_continuous(name="Petal Width (cm)", breaks = c(1.0, 1.25, 1.50, 1.75, 2.0, 2.25, 2.50), limits
```