

# CM580A3: Plotting with ggplot()

## 1. ggplot()

---

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

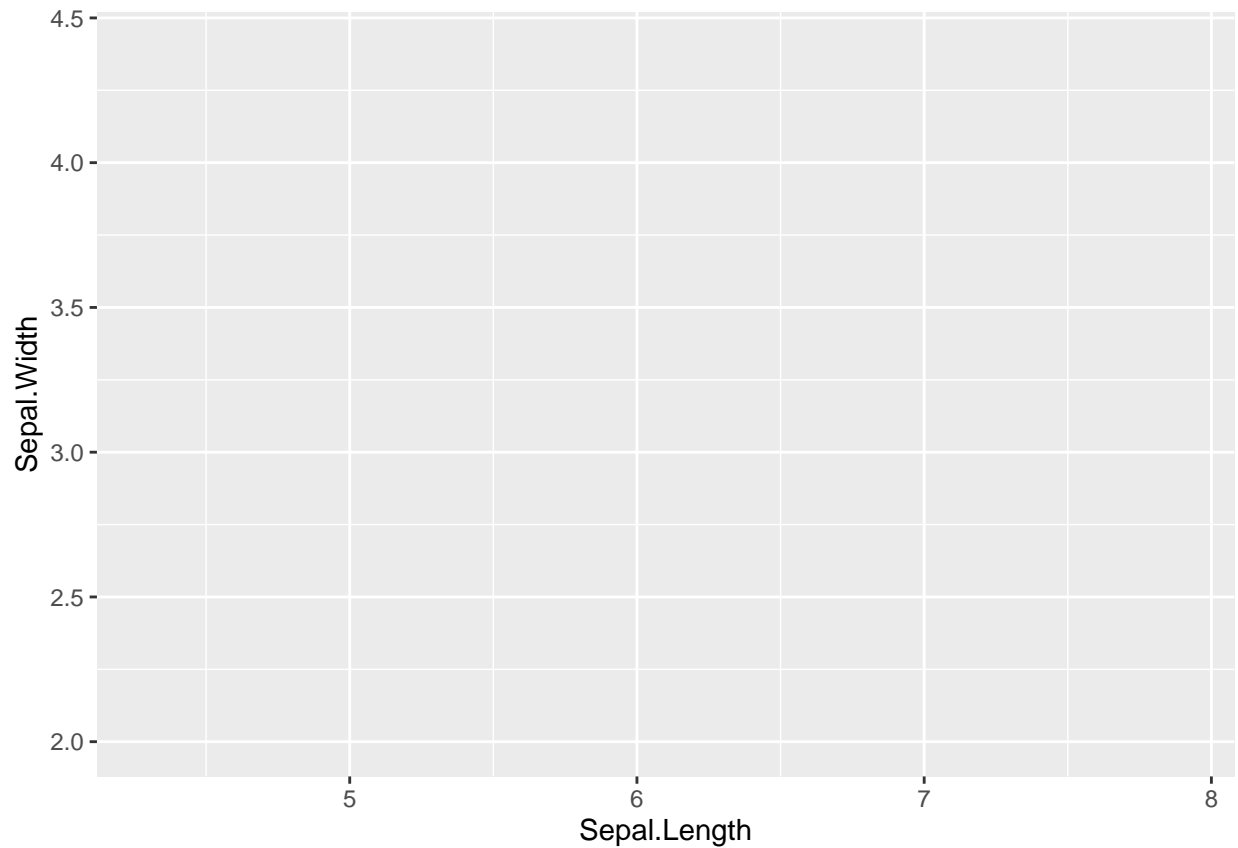
```
####  
#01#  
####  
iris %>%  
  ggplot()
```

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

---

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

```
####  
#02#  
####  
iris %>%  
  ggplot(aes(x = Sepal.Length, y = Sepal.Width))
```

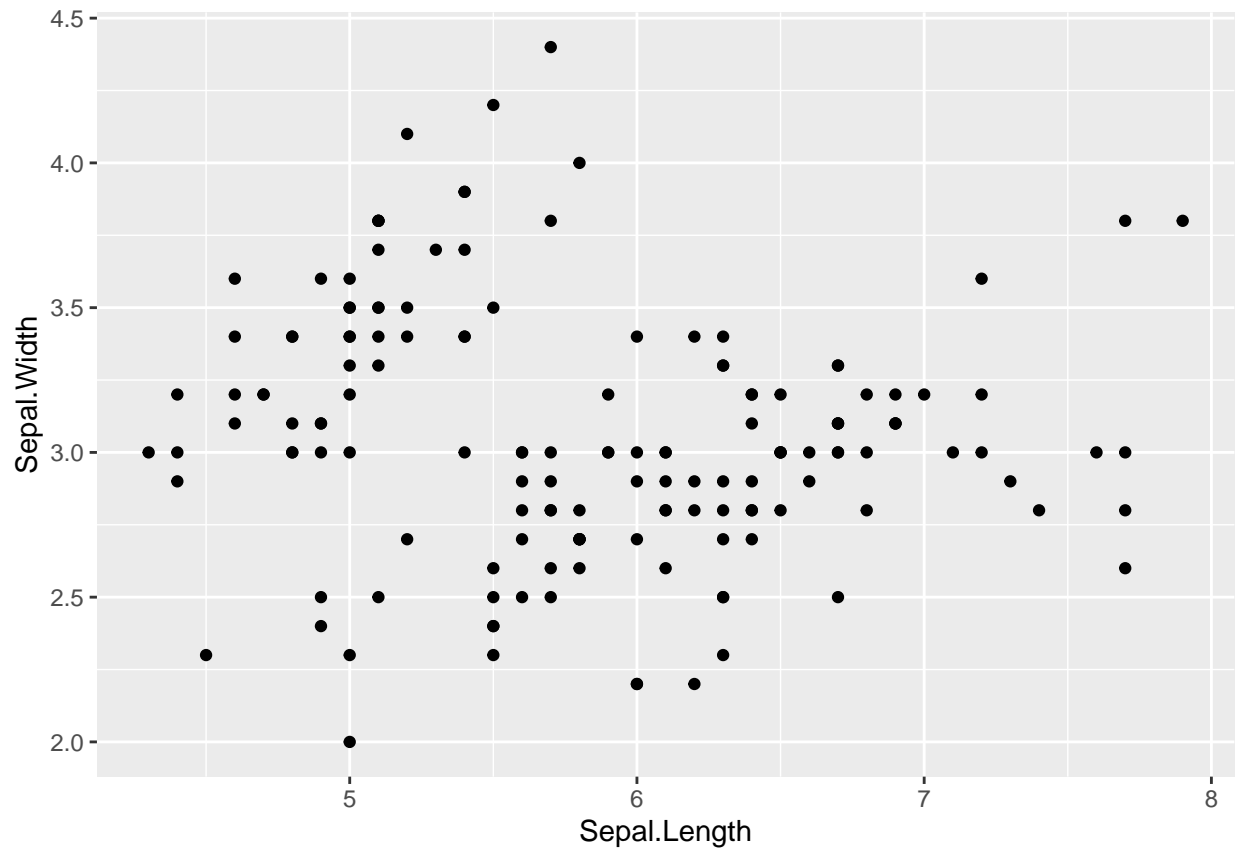


## 3. Specifying a Geometry

---

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 “+”.

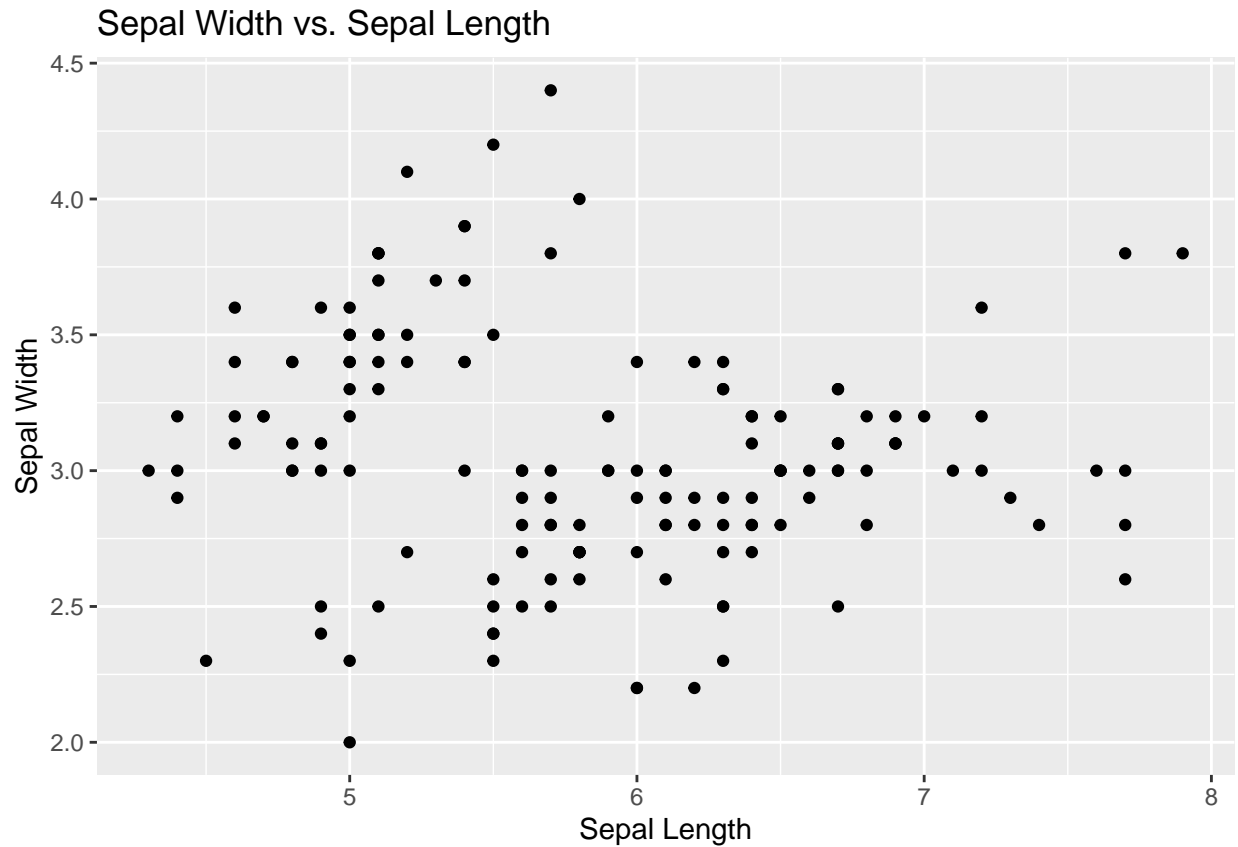
```
####
#03#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point()
```



## 4. Adding Labels and a Title

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

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

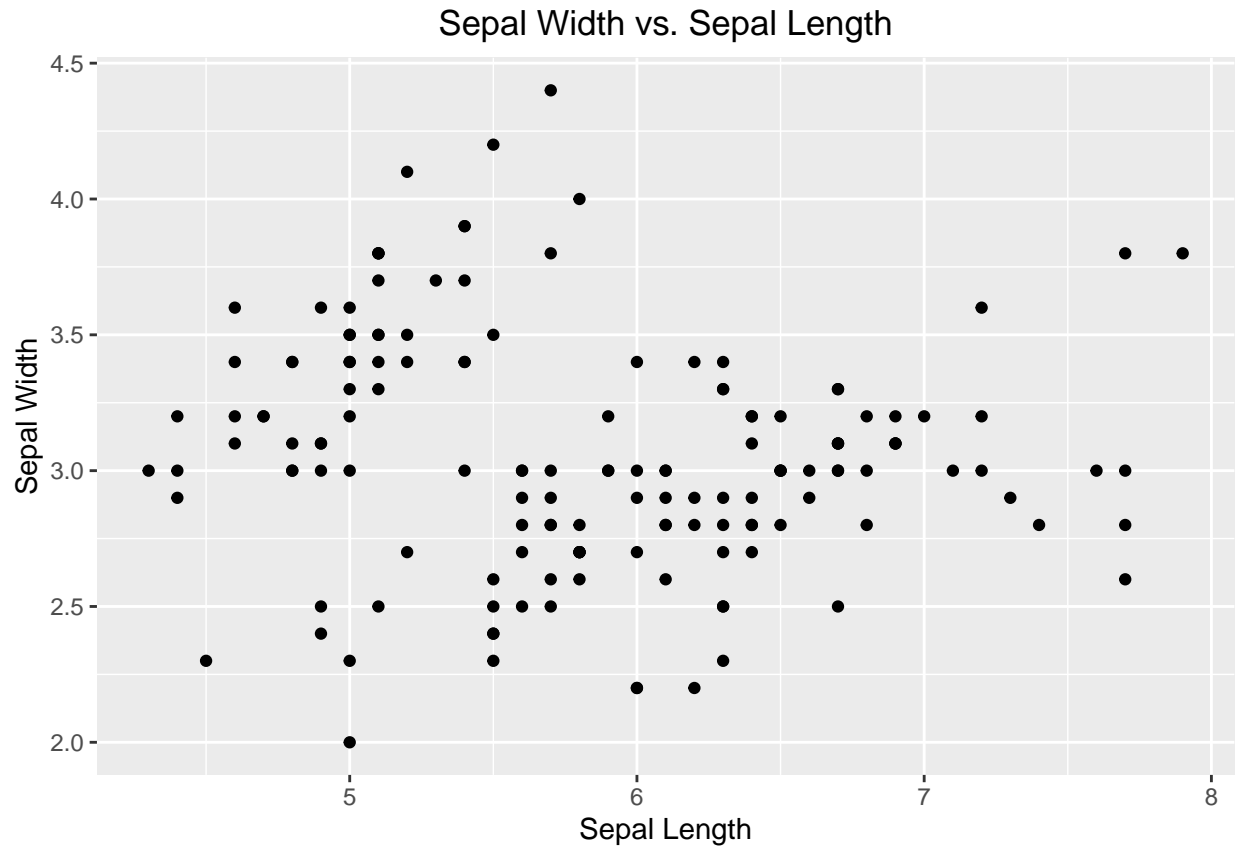


## 5. Centering the Title

---

Many parameters can be adjusted in the theme function. Overall, it's 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.

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



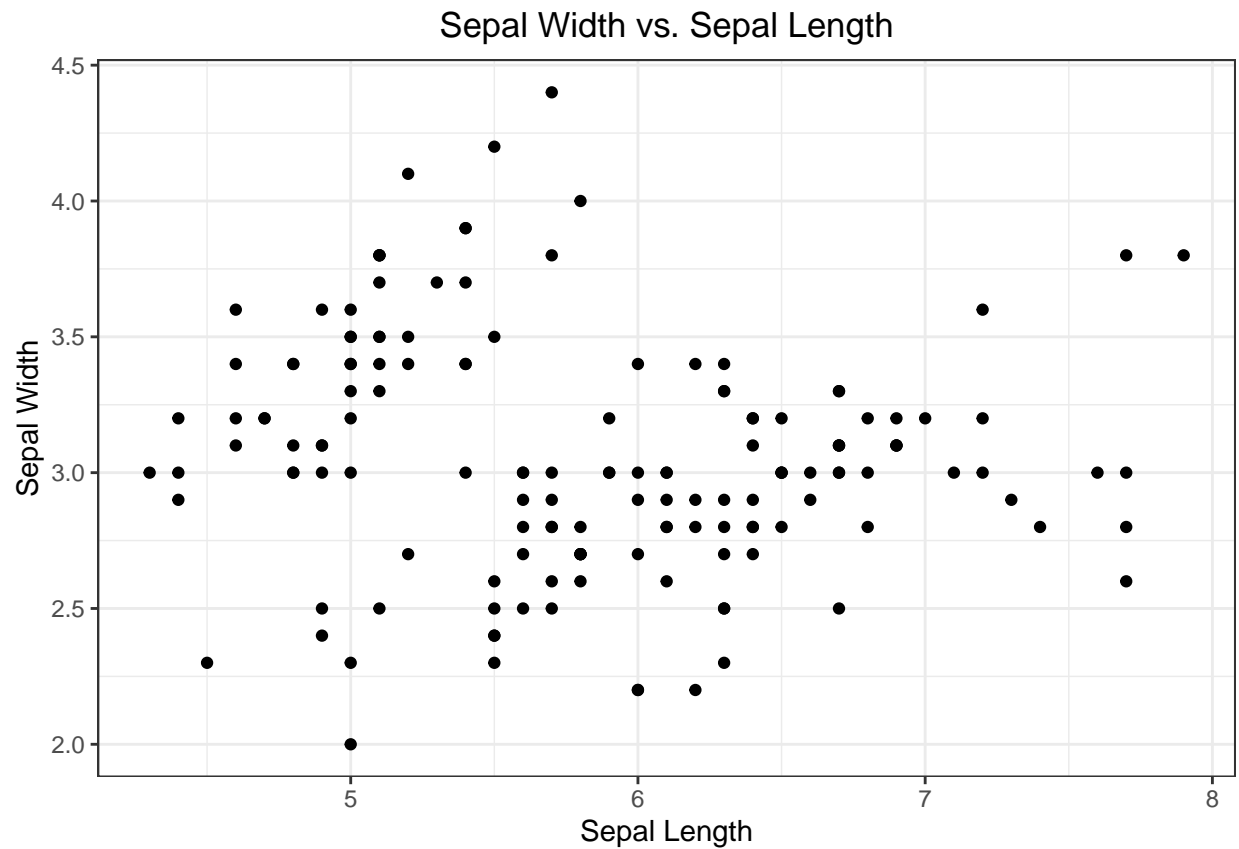
---

## 6. Choosing a Theme

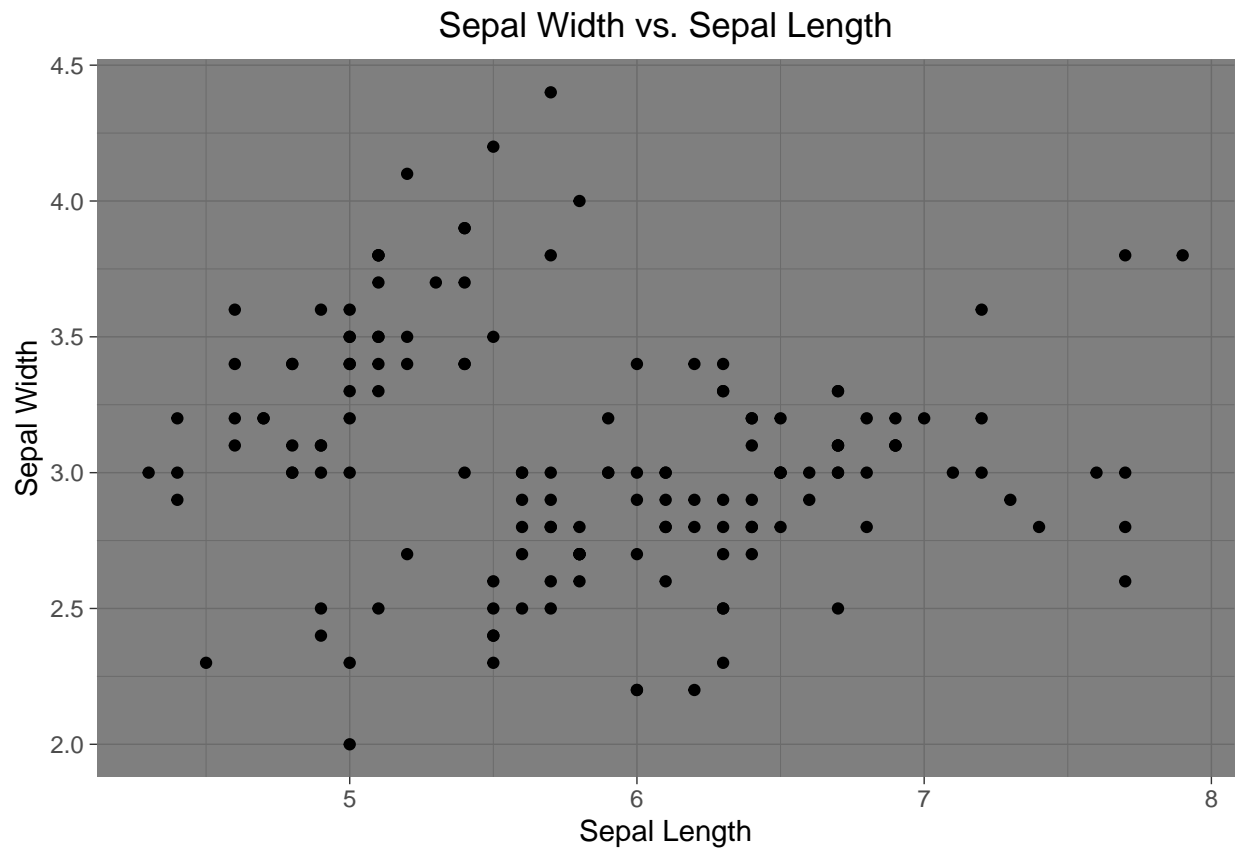
---

There are many themes in `ggplot()`. Here is a link to see a complete list : <https://ggplot2.tidyverse.org/reference/ggtheme.html>. The theme can be added using “+” and specifying a theme, such as “`theme_bw()`”. Here are 3 examples. Note that the “`theme()`” function comes after “`theme_bw()`”.

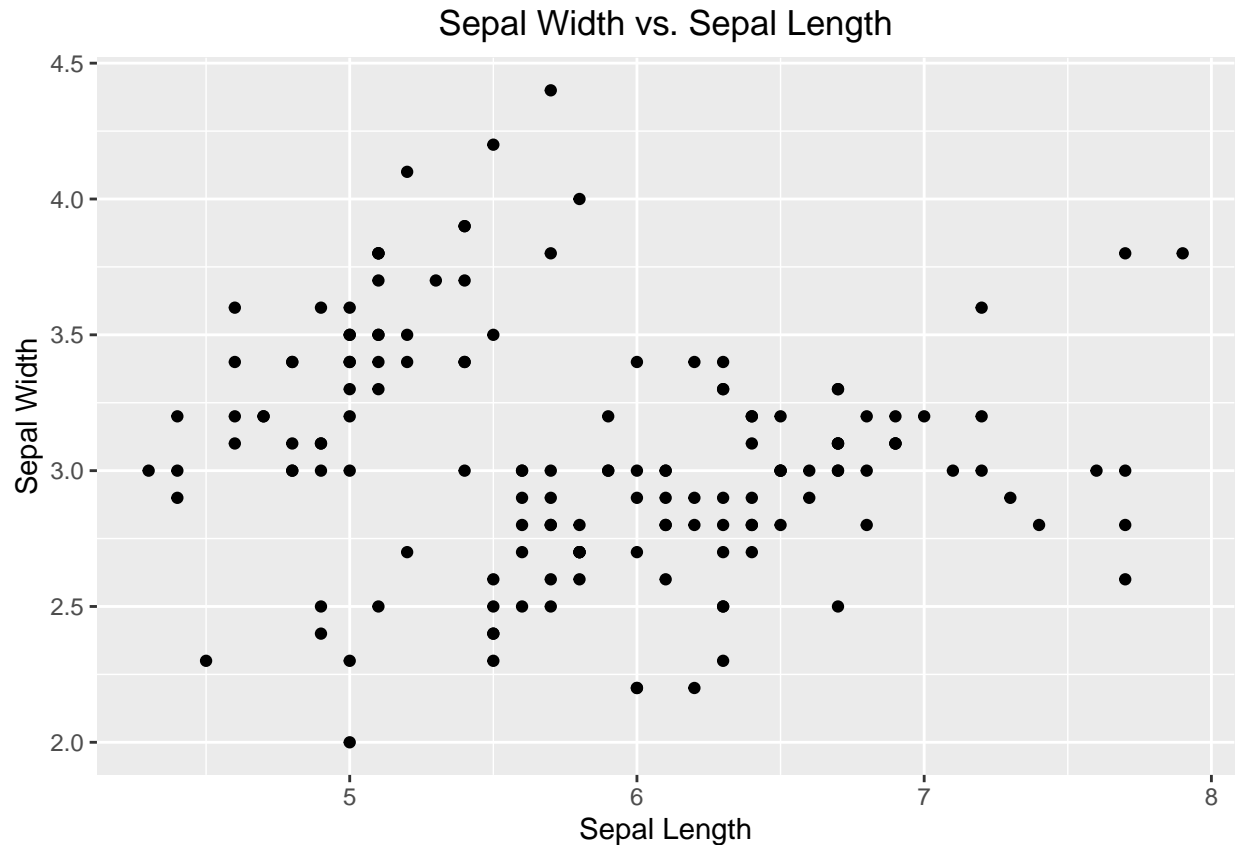
```
####  
#06#  
####  
iris %>%  
  ggplot(aes(x = Sepal.Length, y = Sepal.Width)) +  
  geom_point() +  
  labs(x = "Sepal Length", y = "Sepal Width",  
       title = "Sepal Width vs. Sepal Length") +  
  theme_bw() +  
  theme(plot.title = element_text(hjust = 0.5))
```



```
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))
```



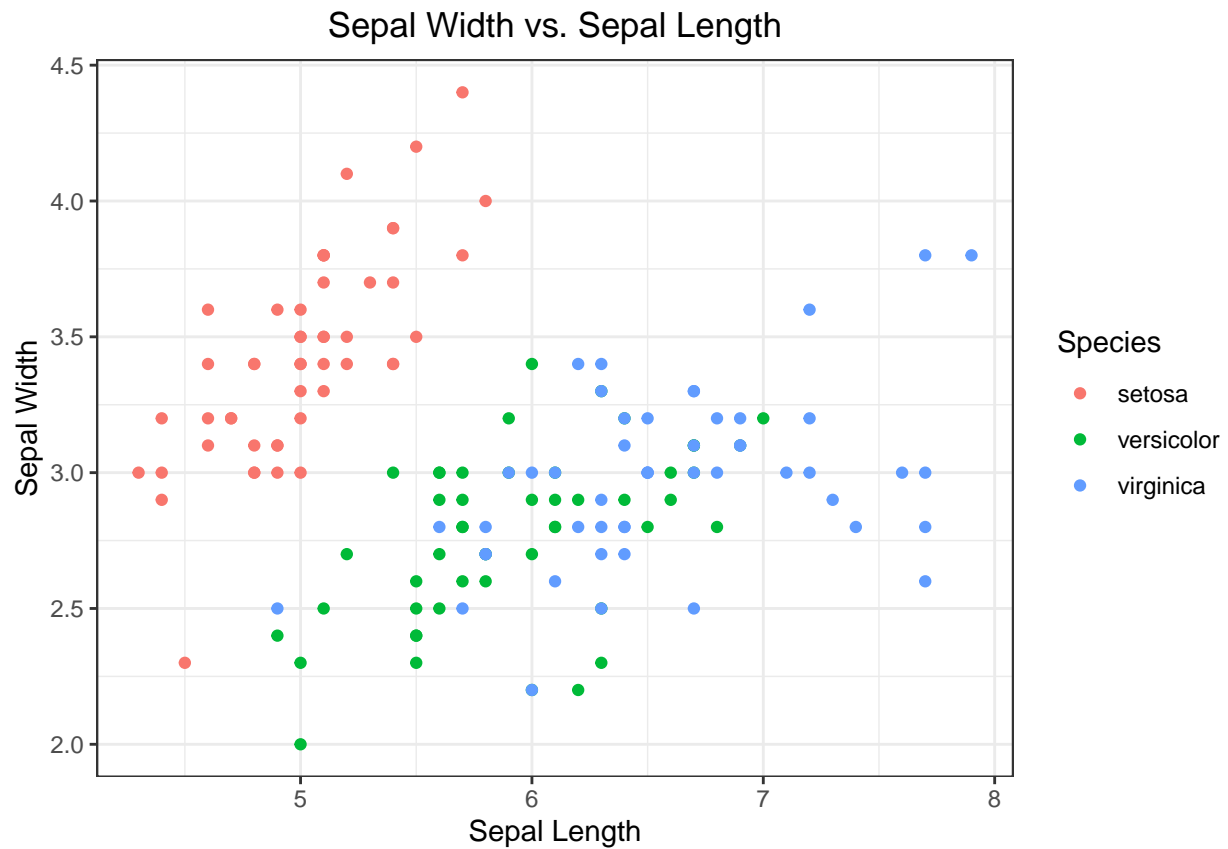
## 7. Coloring by Factor

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”.

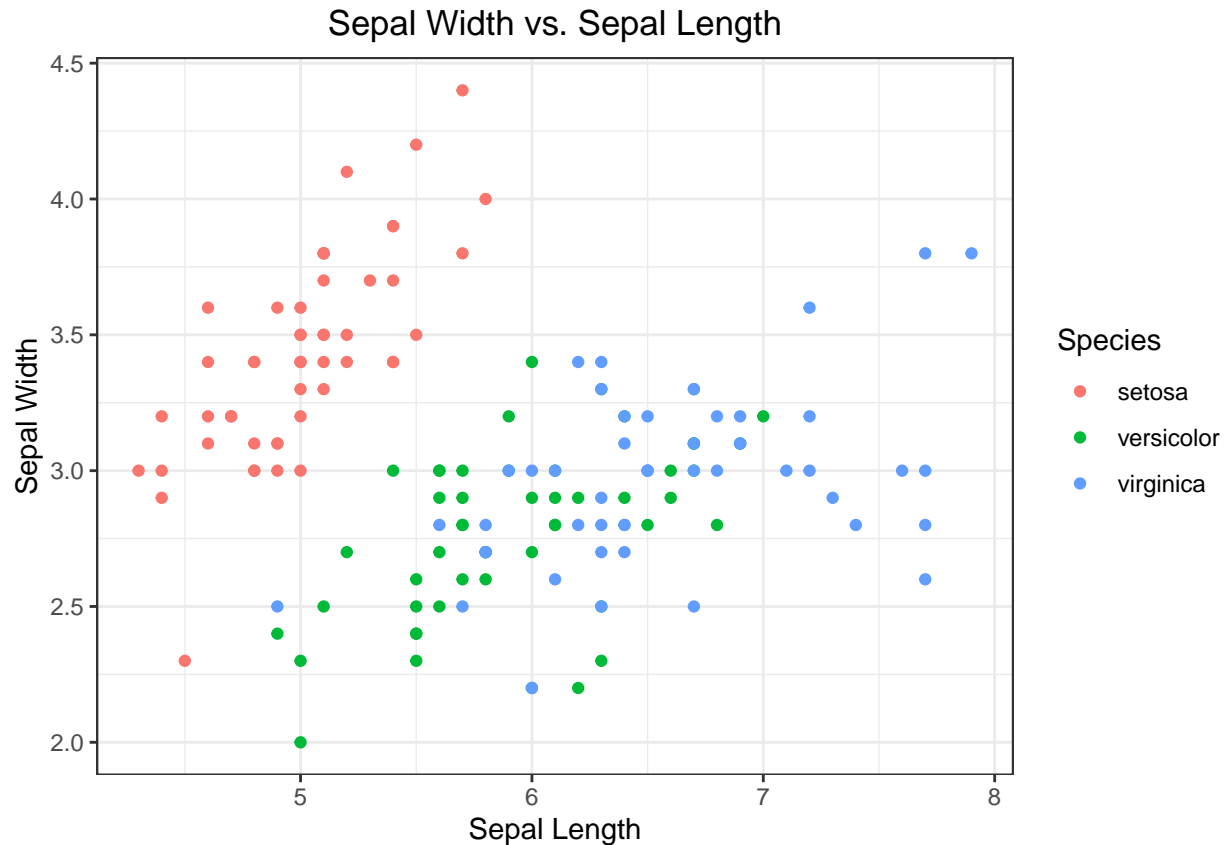
Note: For some geometries and/or variables you may need to use “fill” instead of “color” but overall they function similarly.

```
####
#07#
####
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))
```





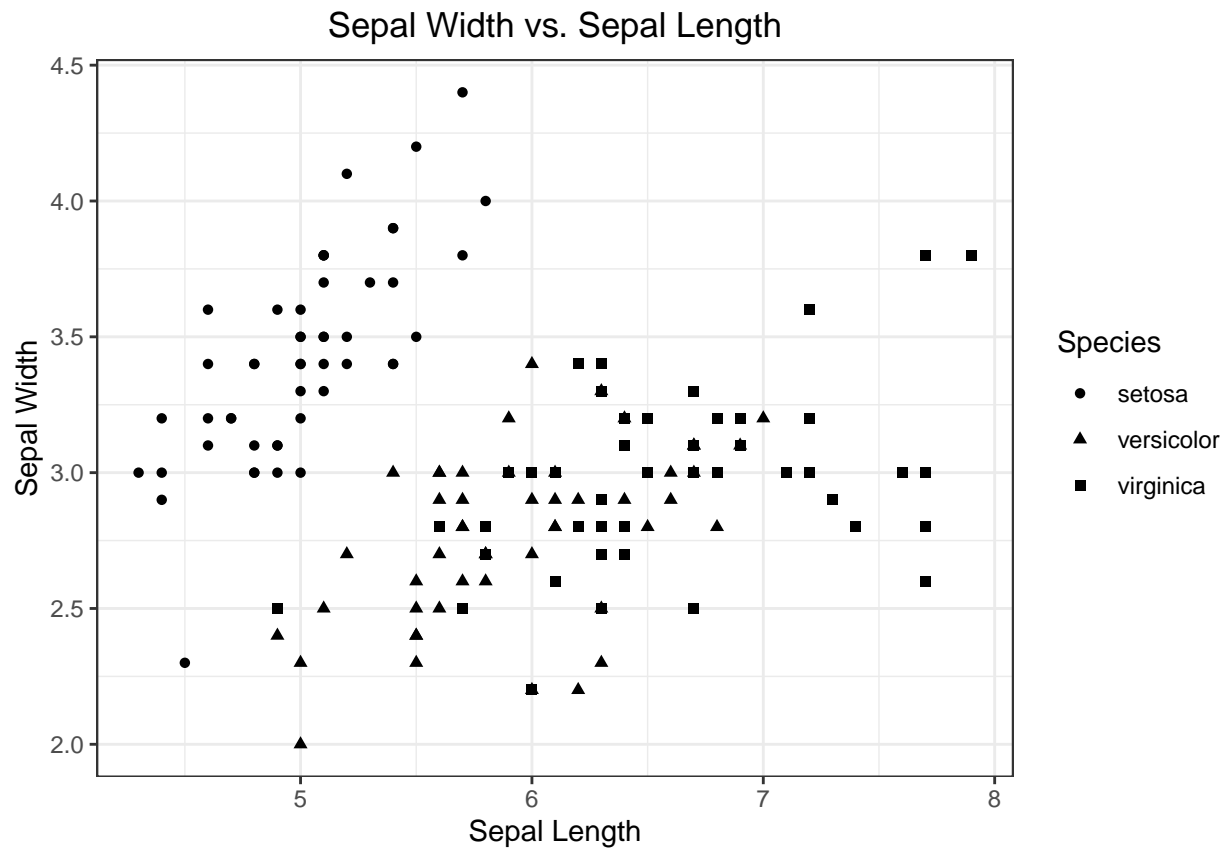
```
iris %>%  
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, col = Species)) +  
  geom_point() +  
  labs(x = "Sepal Length", y = "Sepal Width",  
        title = "Sepal Width vs. Sepal Length") +  
  theme_bw() +  
  theme(plot.title = element_text(hjust = 0.5))
```



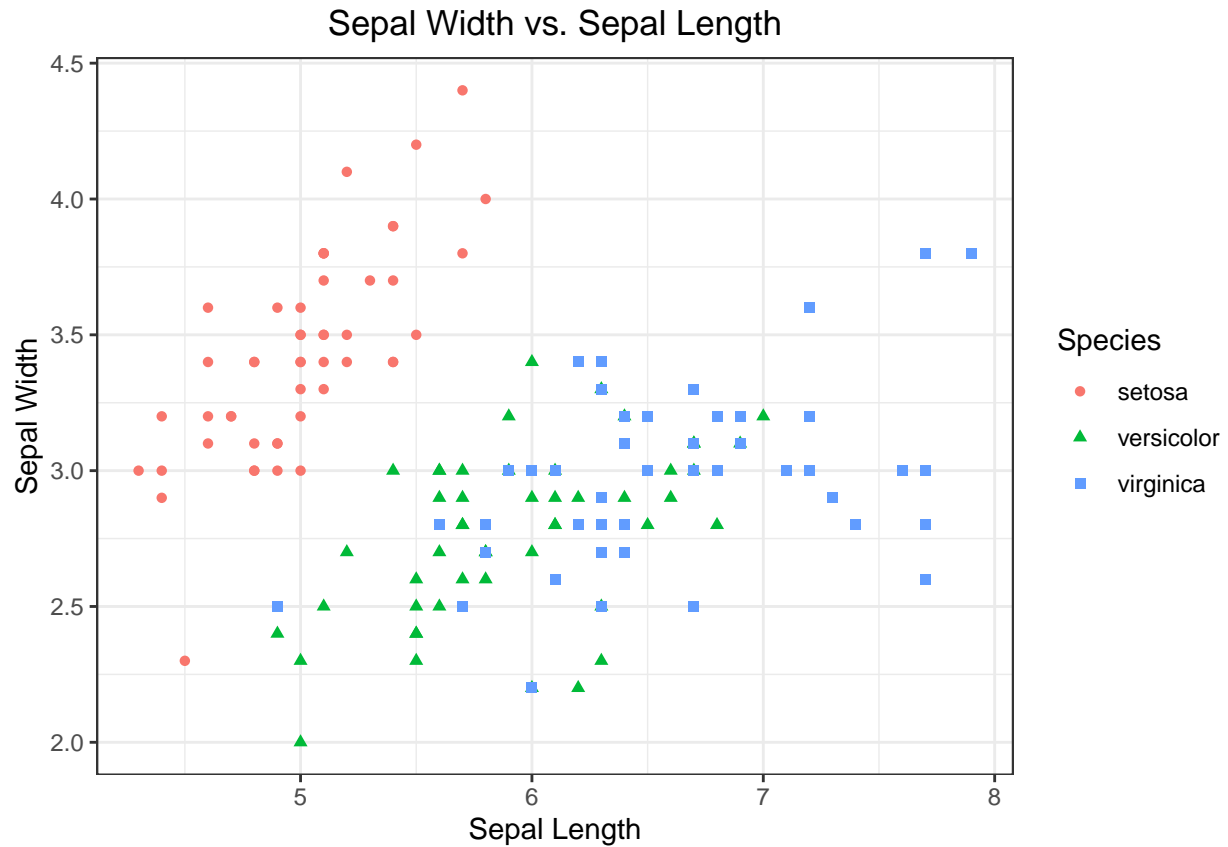
## 8. Using Shape to Denote a Factor

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.

```
####
#08#
####
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))
```



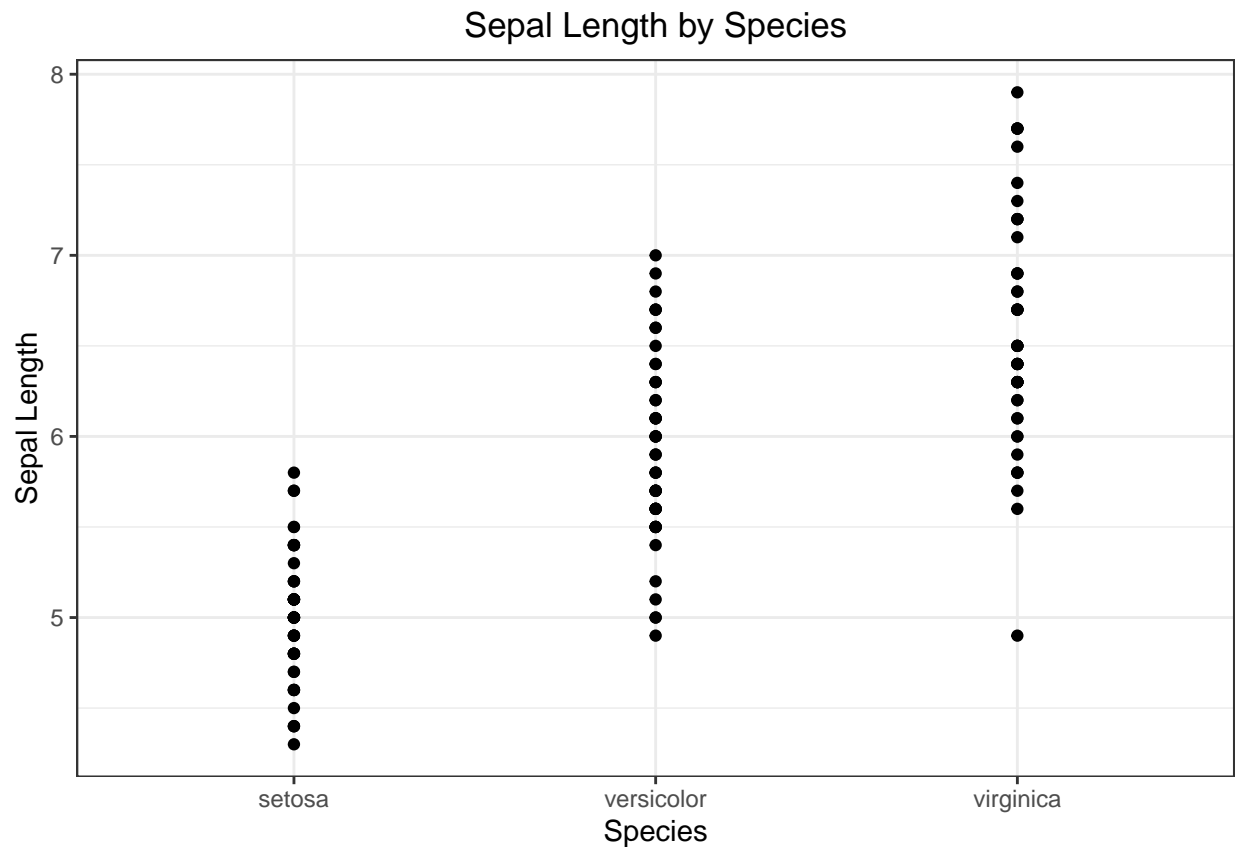
```
iris %>%  
  ggplot(aes(x = Sepal.Length, y = Sepal.Width, color = Species, shape = Species)) +  
  geom_point() +  
  labs(x = "Sepal Length", y = "Sepal Width",  
        title = "Sepal Width vs. Sepal Length") +  
  theme_bw() +  
  theme(plot.title = element_text(hjust = 0.5))
```



## 9. Plotting a Continuous Variable Against A Categorical Variable

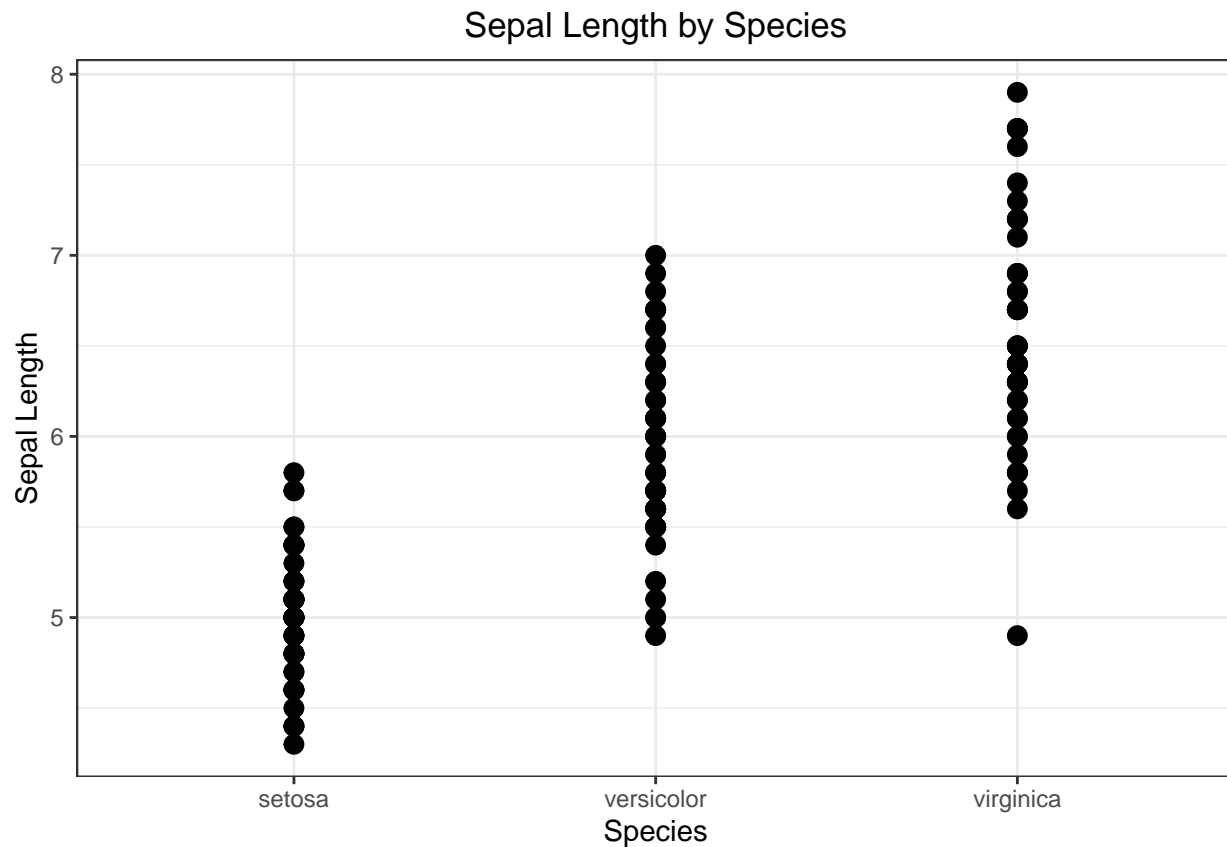
As a Scatter Plot

```
####
#09#
####
iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_point() +
  labs(x = "Species", y = "Sepal Length",
       title = "Sepal Length by Species") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```



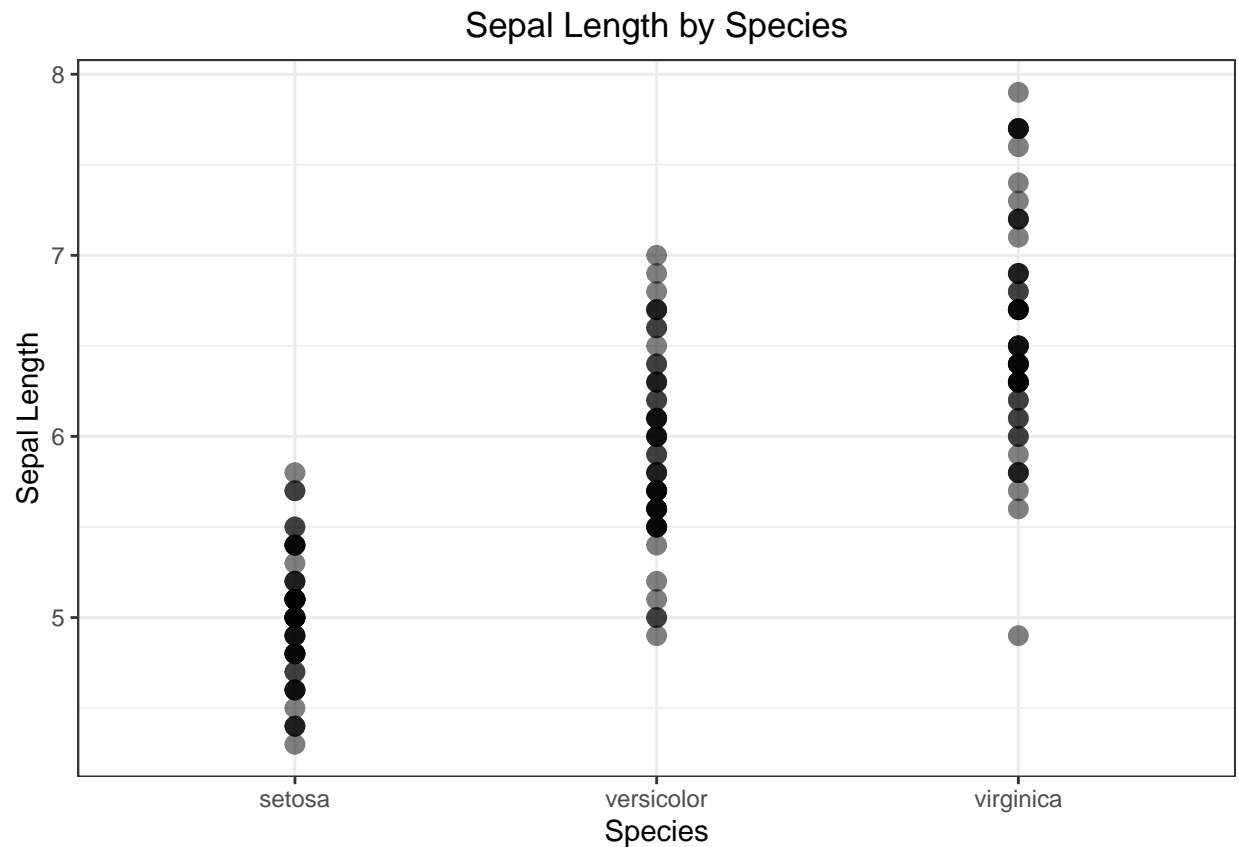
The size of points can be changed by specifying “size” withing the `geom_point()` function.

```
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))
```



Similarly, we can adjust transparency with “alpha”. We can see that some points overlap so maybe `geom_jitter()` is a better option

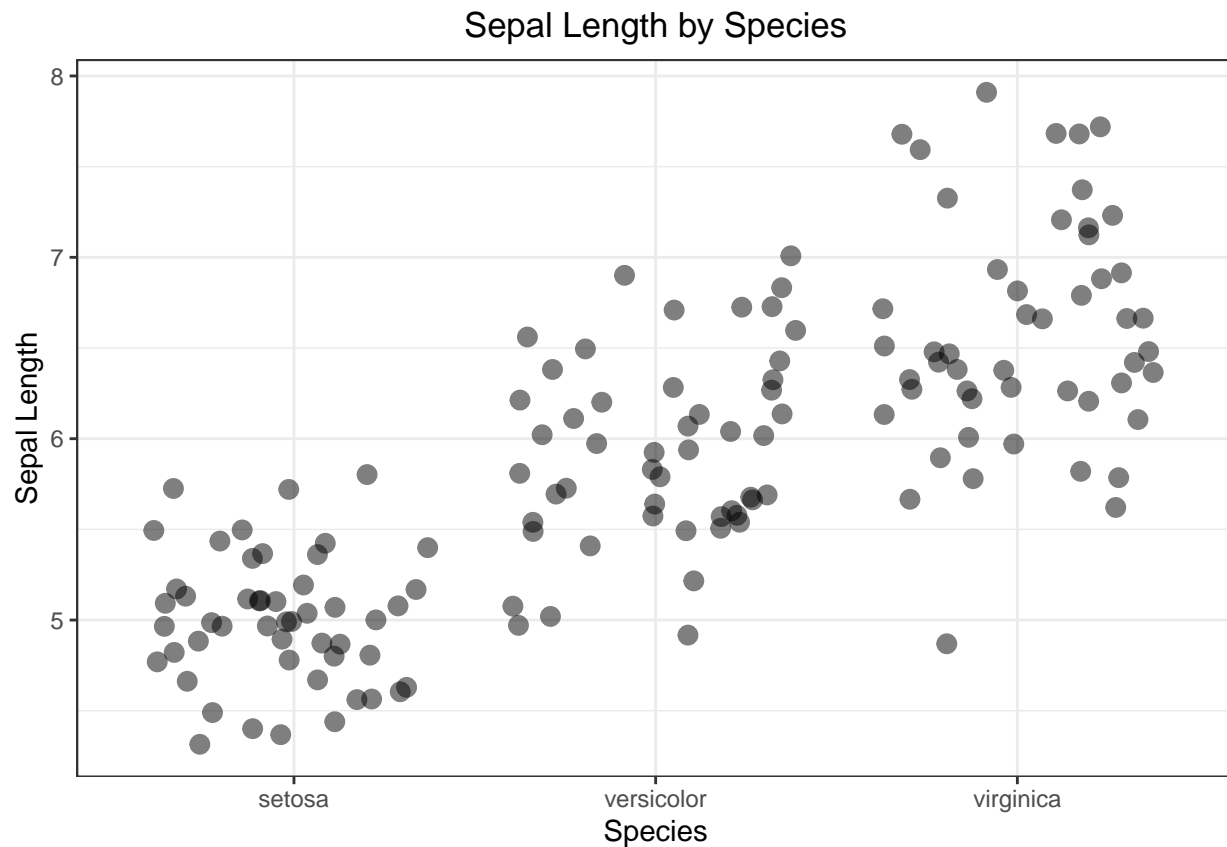
```
iris %>%  
  ggplot(aes(x = Species, y = Sepal.Length)) +  
  geom_point(size = 3,  
             alpha = .5) +  
  labs(x = "Species", y = "Sepal Length",  
       title = "Sepal Length by Species") +  
  theme_bw() +  
  theme(plot.title = element_text(hjust = 0.5))
```



## As a Jitter Plot

Now we can see individual points better.

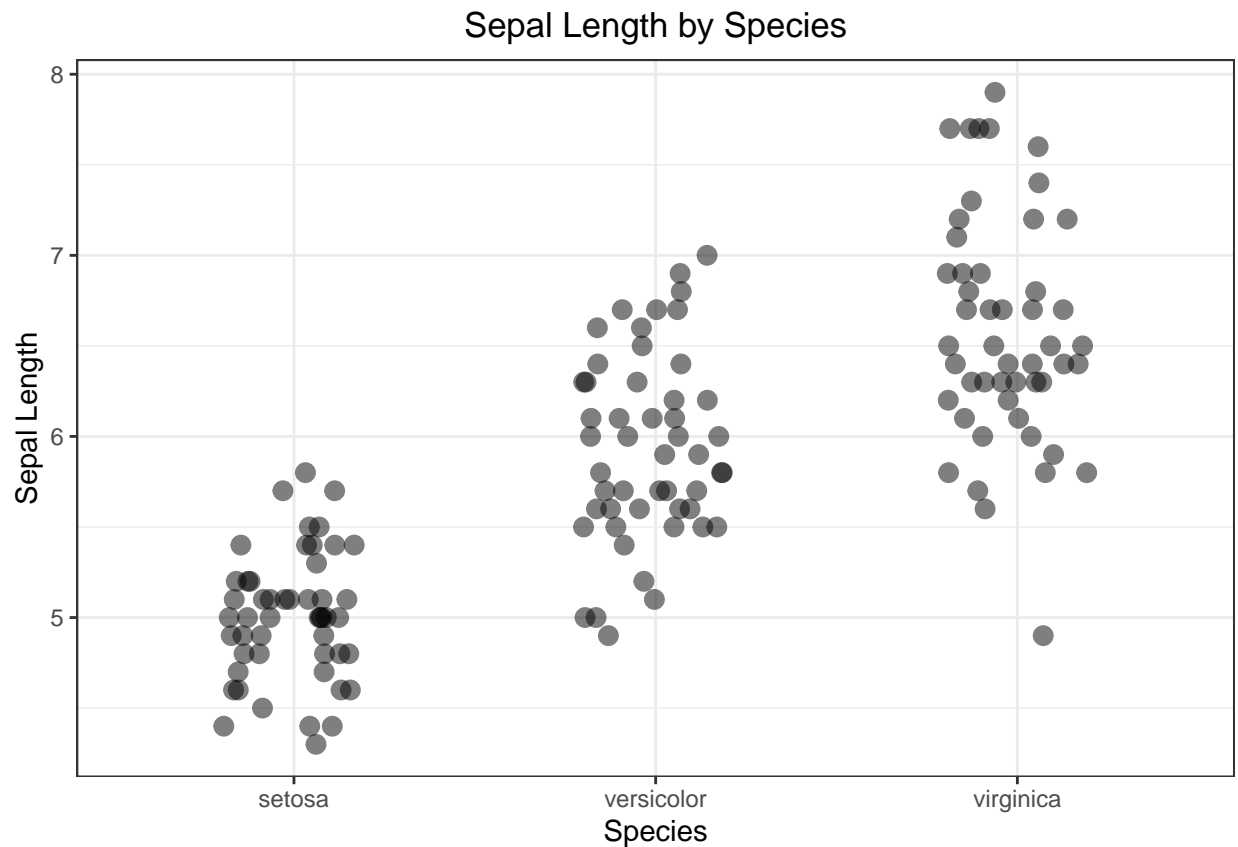
```
iris %>%  
  ggplot(aes(x = Species, y = Sepal.Length)) +  
  geom_jitter(size = 3,  
              alpha = .5) +  
  labs(x = "Species", y = "Sepal Length",  
        title = "Sepal Length by Species") +  
  theme_bw() +  
  theme(plot.title = element_text(hjust = 0.5))
```



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 %>%
  group_by(Species) %>%
  summarise(mean = mean(Sepal.Length),
            sd = sd(Sepal.Length),
            n = n(),
            se = sd/sqrt(n))
```

```
## # 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
```

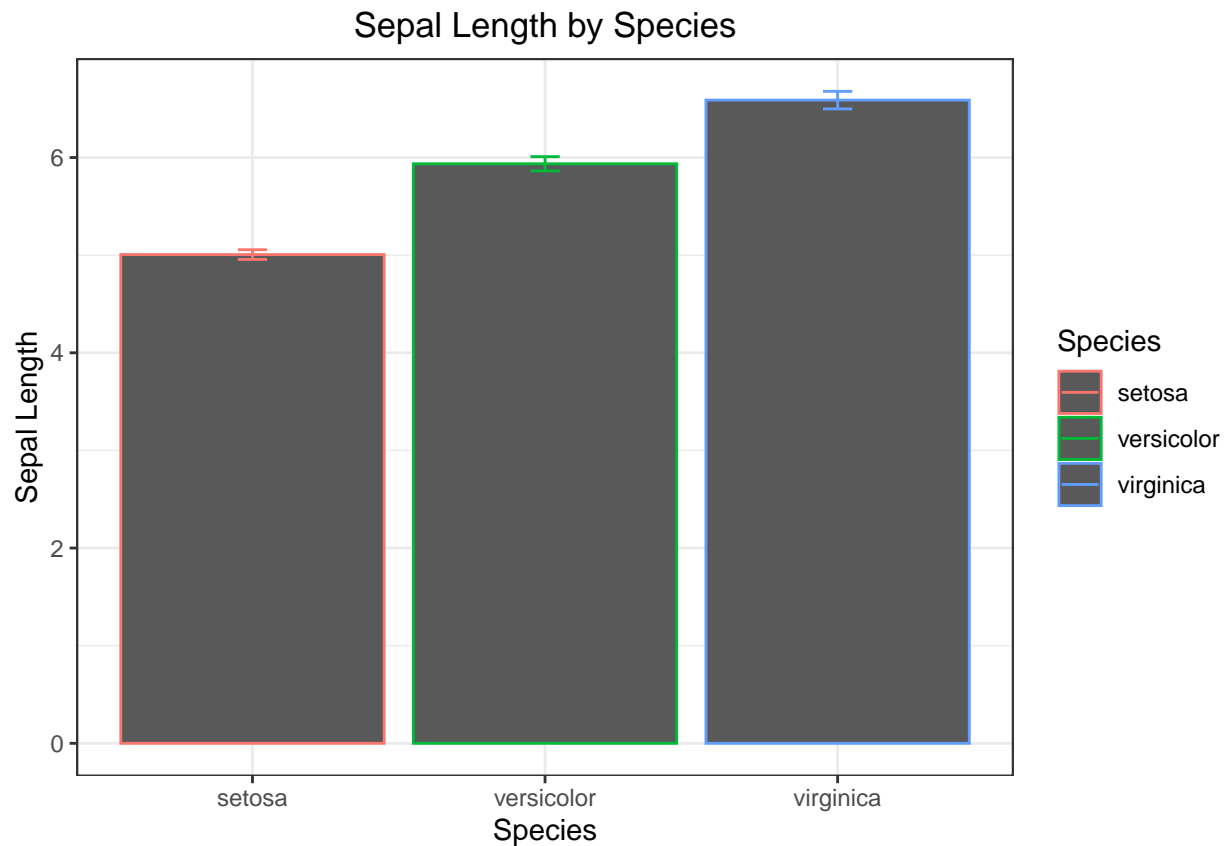
```
iris %>%
  group_by(Species) %>%
  summarise(mean = mean(Sepal.Length),
            sd = sd(Sepal.Length),
            n = n(),
            se = sd/sqrt(n)) %>%
  ggplot(aes(x = Species, y = mean, fill = Species)) +
  geom_col() +
  geom_errorbar(aes(ymax = mean + se, ymin = mean - se),
               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.

```
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))
```

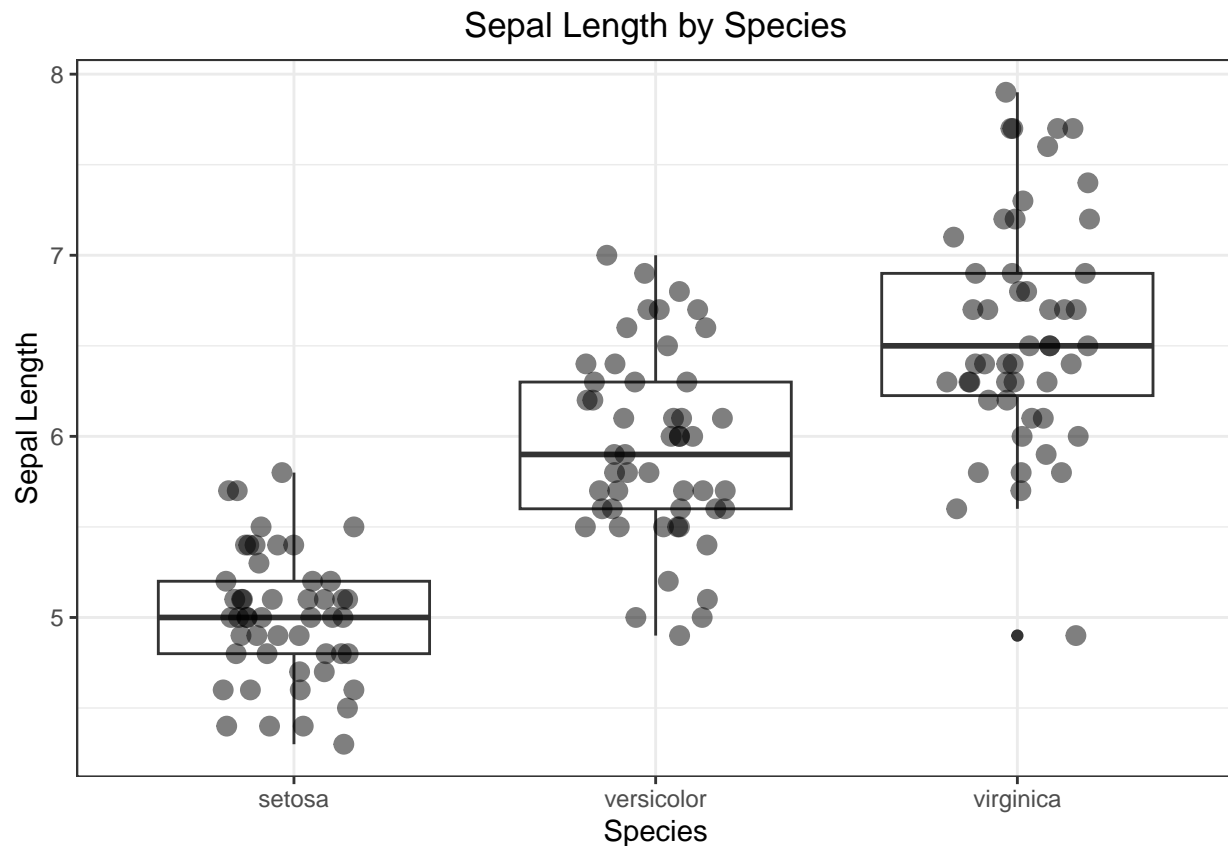


## 10. Overlaying Geoms: Boxplot with Points Overlain

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 virginica species.

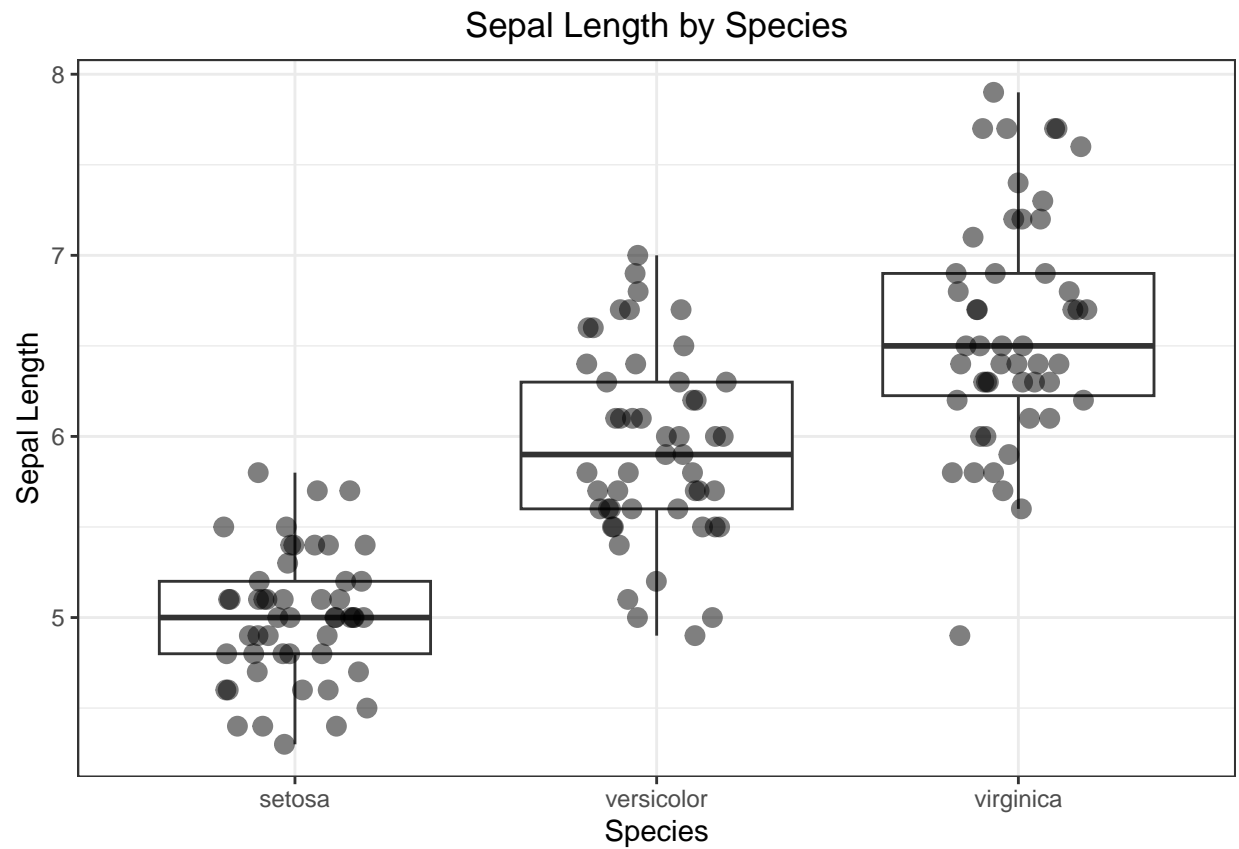
```
####
#10#
####
iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_boxplot() +
```

```
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))
```



## Appendix

```
library(datasets)
library(tidyverse)
library(knitr)
data(iris)

####
#01#
####
iris %>%
  ggplot()

####
#02#
####
iris %>%
  ggplot(aes(x = Sepal.Length, y = Sepal.Width))

####
#03#
```

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

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

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

####
#06#
####
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))

####
#07#
```

```
####
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 = 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))

####
#08#
####
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))

####
#09#
####
iris %>%
  ggplot(aes(x = Species, y = Sepal.Length)) +
  geom_point() +
  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) +
  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 %>%
  group_by(Species) %>%
  summarise(mean = mean(Sepal.Length),
            sd = sd(Sepal.Length),
            n = n(),
            se = sd/sqrt(n))
```

```
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, 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))

####
#10#
####
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))

```