

Advanced ggplot

Debbie Yee
R Mini Workshop

Department of Psychological and Brain Sciences
Washington University in St. Louis

September 30, 2016

1 Loading required packages

```
library(ggplot2)
library(grid)
library(gridExtra)
library(lattice)
library(psych)
library(ggthemes)
library(extrafont)
```

2 Iris dataset at a glance

For this tutorial, we're going to use the "iris" dataset, which gives the measurements of sepal length and width and petal length and width for 50 flowers from each of 3 species of iris. There are 150 rows in this dataframe.

```
# To look at the description of the iris dataset
help("iris")

# prints the first 6 rows of the iris dataset
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

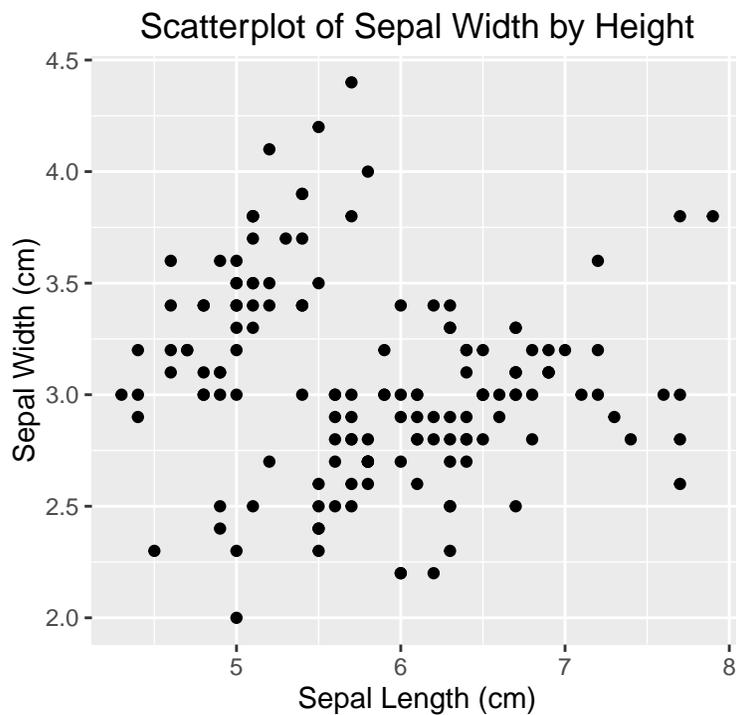
# describe is a function from the psych package that provides some descriptive
# statistics about the column variables in the iris dataframe
describe(iris)

##           vars  n mean  sd median trimmed  mad min max range  skew
## Sepal.Length  1 150 5.84 0.83  5.80  5.81 1.04 4.3 7.9  3.6  0.31
## Sepal.Width   2 150 3.06 0.44  3.00  3.04 0.44 2.0 4.4  2.4  0.31
## Petal.Length  3 150 3.76 1.77  4.35  3.76 1.85 1.0 6.9  5.9 -0.27
## Petal.Width   4 150 1.20 0.76  1.30  1.18 1.04 0.1 2.5  2.4 -0.10
## Species*     5 150 2.00 0.82  2.00  2.00 1.48 1.0 3.0  2.0  0.00
##           kurtosis  se
## Sepal.Length -0.61 0.07
## Sepal.Width   0.14 0.04
## Petal.Length -1.42 0.14
## Petal.Width -1.36 0.06
## Species*     -1.52 0.07
```

3 Scatterplot of Sepal Widths and Sepal Lengths

Here, let us start with a simple scatterplot. On the x-axis, we will plot the Sepal Length, while on the y-axis, we will plot the Sepal Width. We have used `xlab()` and `ylab()` to label the x and y axes, and use `ggtitle()` to label the title in your plots. This should be a nice throwback to all of those graphs you made in high school back in the day. But the life lesson is pretty relevant: YOU SHOULD ALWAYS LABEL YOUR AXES.

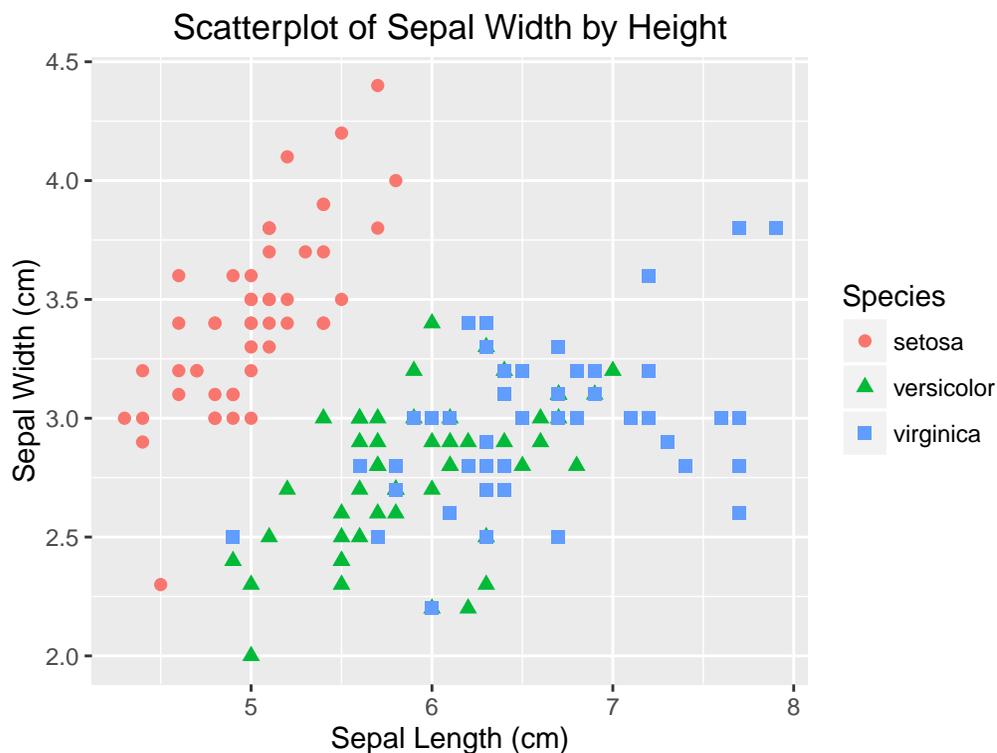
```
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) +  
  geom_point() +  
  xlab("Sepal Length (cm)") + ylab("Sepal Width (cm)") +  
  ggtitle("Scatterplot of Sepal Width by Height")
```



Grouping your data by category

Okay, so you're probably thinking, "this plot is nice, but not super informative." You know that you have three clusters of data based on Species, so I want my graph to show that. So one thing that you could do is label the dots on your graph by species by modifying your `geom_point()` function. Additionally, for good measure, you decide to also make the shapes different, in case anyone who reads these graphs are colorblind.

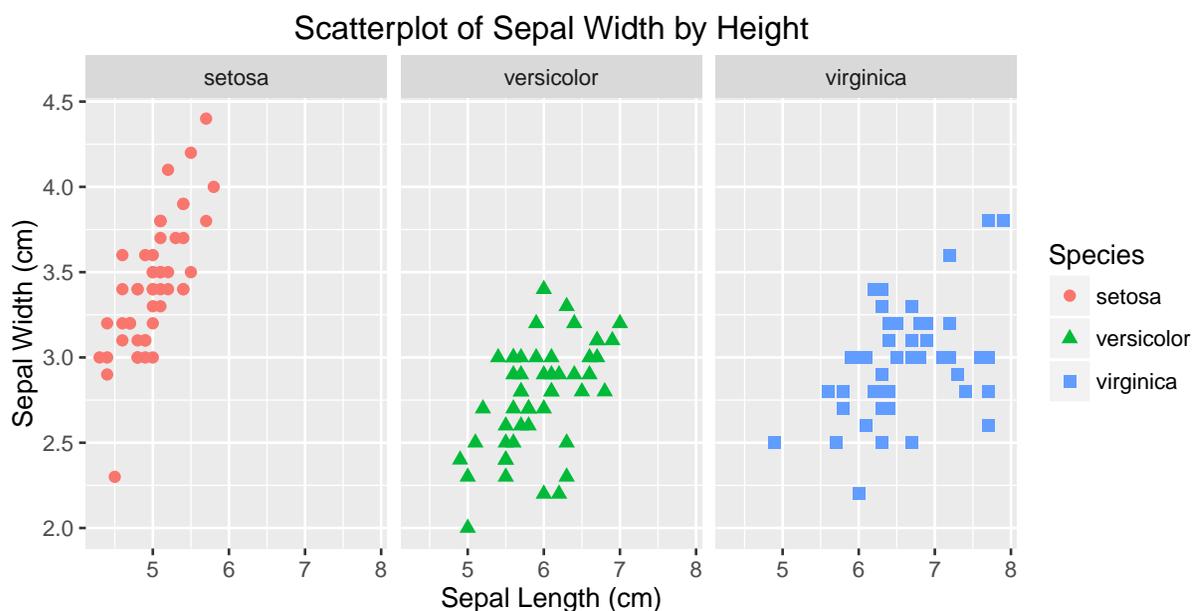
```
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) +  
  geom_point(aes(color=Species, shape=Species), size=2) +  
  xlab("Sepal Length (cm)") + ylab("Sepal Width (cm)") +  
  ggtitle("Scatterplot of Sepal Width by Height")
```



4 Faceting your graphs

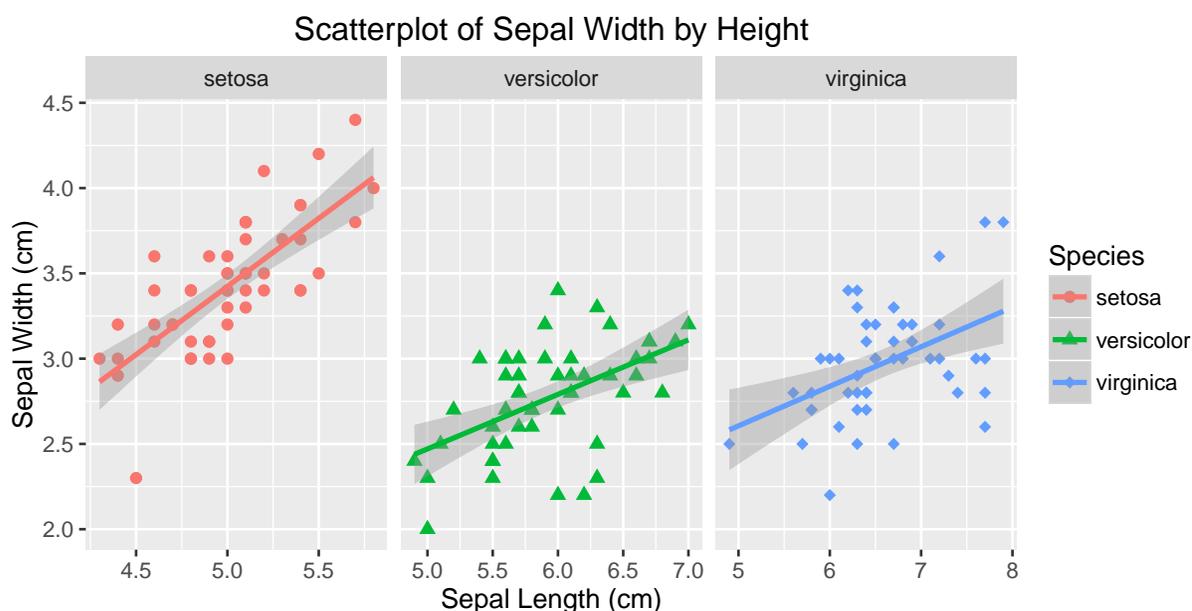
So you look at your graphs and realize that the data are too clustered together. So you decide you want to facet your graphs, so that you can have separate graphs for each of your scatterplots. You can add the `facet_grid()` function as another layer in your `ggplot`, and it separates your scatterplots by species, which you think will be much more useful for your poster or publication. There is a related function called `facet_wrap()` that does something similar, but has different syntax and wraps the grids based on the number of columns or rows you specify.

```
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) +  
  geom_point(aes(color=Species, shape=Species), size=2) +  
  xlab("Sepal Length (cm)") + ylab("Sepal Width (cm)") +  
  ggtitle("Scatterplot of Sepal Width by Height") +  
  facet_grid(.~Species)
```



To make things even more fancy, we can add some regression lines with confident intervals. And maybe you have a thing against square shapes, so you decide that diamonds are far superior than squares and would make your data look even more beautiful than before. I also want the data to be centered, so I change the scale of my facet to adjust the range of the x-values based on the range of the data in the sub-category.

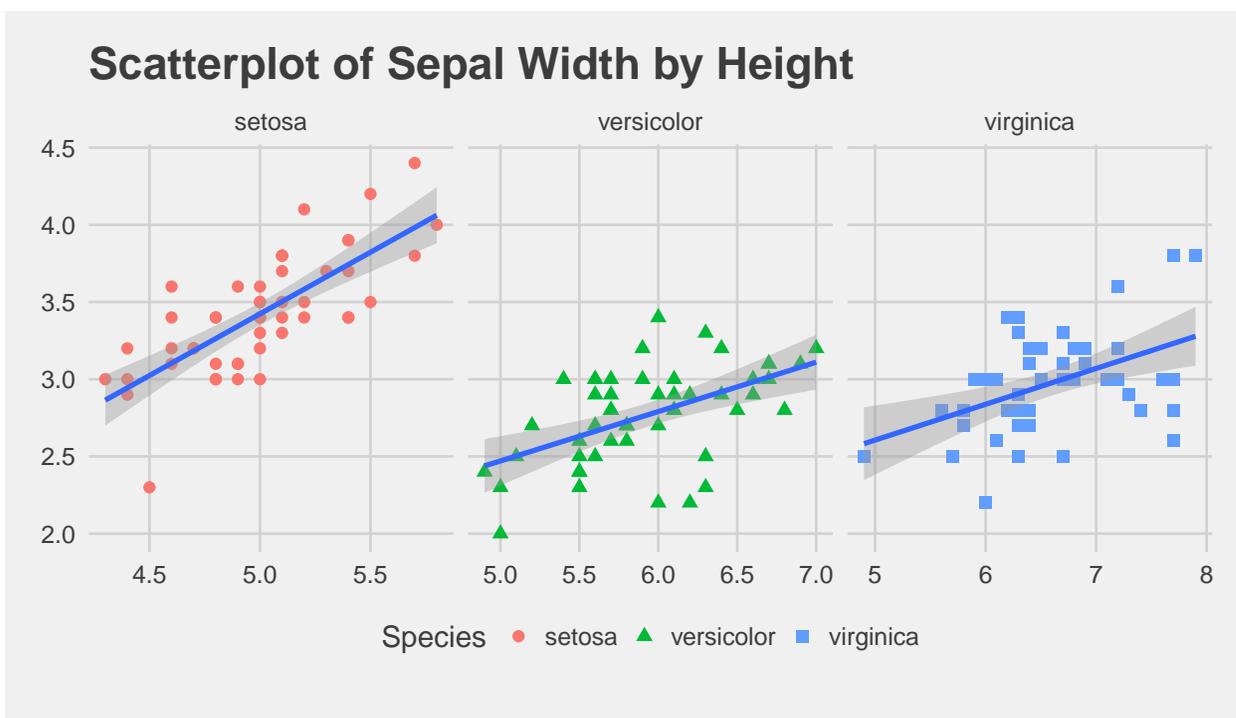
```
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) +  
  geom_point(aes(color=Species, shape=Species), size=2) +  
  scale_shape_manual(values=c(16,17,18)) +  
  geom_smooth(aes(color=Species),method=lm) +  
  xlab("Sepal Length (cm)") + ylab("Sepal Width (cm)") +  
  ggtitle("Scatterplot of Sepal Width by Height") +  
  facet_grid(.~Species, scales = "free_x")
```



5 ggplot Themes

Let's say that you are happy with your graph and want to make it ready for a poster, but the default ggplot graphics won't stand out at your conference. So you want to make your poster stand out. A cool feature of ggplot are themes: there are some that are preset in ggplot, but there are also some preset in the ggthemes package. Once you install the package, you can use any of the default themes. I'm a huge fan of Nate Silver, so I decide to use `theme_fivethirtyeight()` because I have wild dreams of my data being posted on his blog one day and becoming viral on the internet. A girl can dream, right?

```
library(ggthemes)
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point(aes(color=Species, shape=Species), size=2) +
  geom_smooth(method=lm) +
  xlab("Sepal Length (cm)") + ylab("Sepal Width (cm)") +
  ggtitle("Scatterplot of Sepal Width by Height") +
  facet_grid(~Species, scales = "free_x") +
  theme_fivethirtyeight()
```

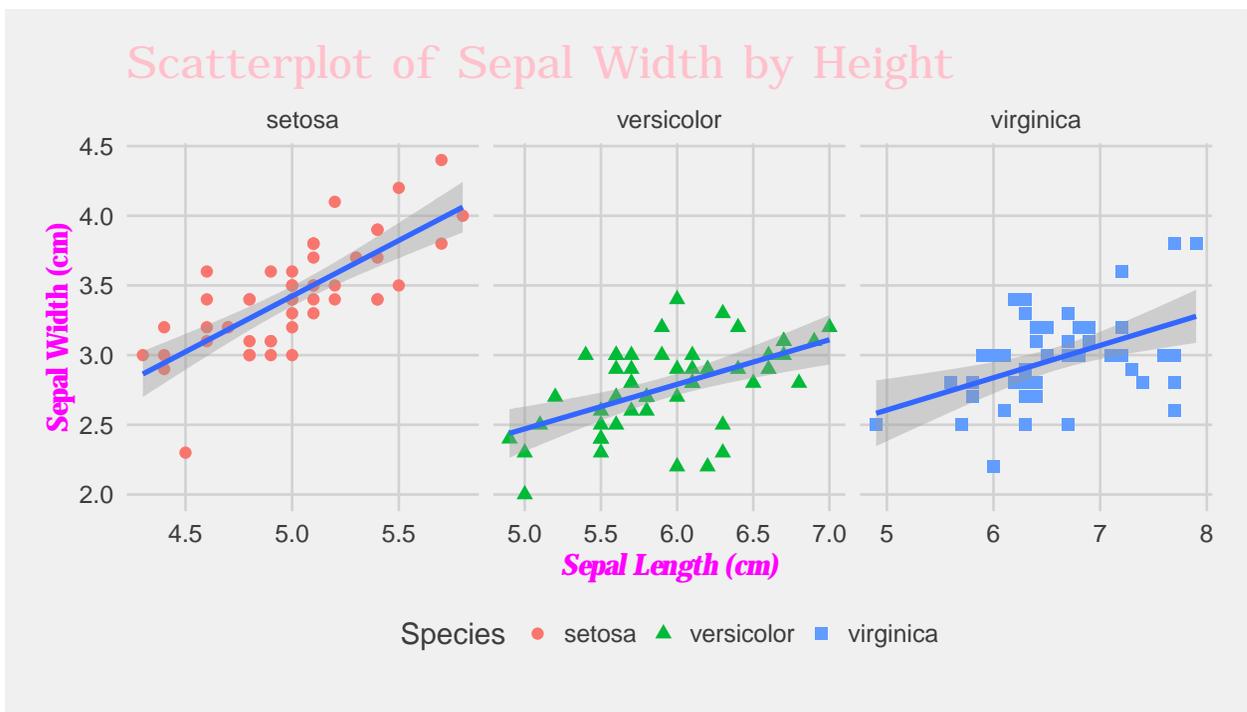


6 Customizing your Themes

If you want to add further customizations to your plot (e.g., make your font size bigger, change font, etc.), you can also do this manually by using the `theme()` function. You can go crazy and change all the colors and sizes if you so desire, but I'd generally recommend sticking with the basic colors, since these combinations might induce a migraine for your viewer. Also, for the love of Fisher, please never use Comic Sans (Just because you can it, doesn't mean you should!).

```
library(extrafont)
font_import() # import all of your fonts, only need to do this once
fonts() # get a list of the fonts
fonttable() # useful table listing of the family name, font name, etc.
```

```
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point(aes(color=Species, shape=Species), size=2) +
  geom_smooth(method=lm) +
  xlab("Sepal Length (cm)") + ylab("Sepal Width (cm)") +
  ggtitle("Scatterplot of Sepal Width by Height") +
  facet_grid(~Species, scales = "free_x") +
  theme_fivethirtyeight() +
  theme(axis.title = element_text(color="magenta", family = "Palatino", face="italic"),
        plot.title = element_text(color = "pink", family = "Comic Sans MS"))
```



7 gridExtra: putting multiple plots in a grid

Sometimes you may want to arrange different types of graphs into one object. So let's say we want to plot a histogram, a boxplot, and a violin plot of the data, because we are detail oriented scientists and are interested in visualizing our data in multiple formats. Or maybe because our advisor may want to see the data in a different format. Whatever the motivation, here's how you can do it.

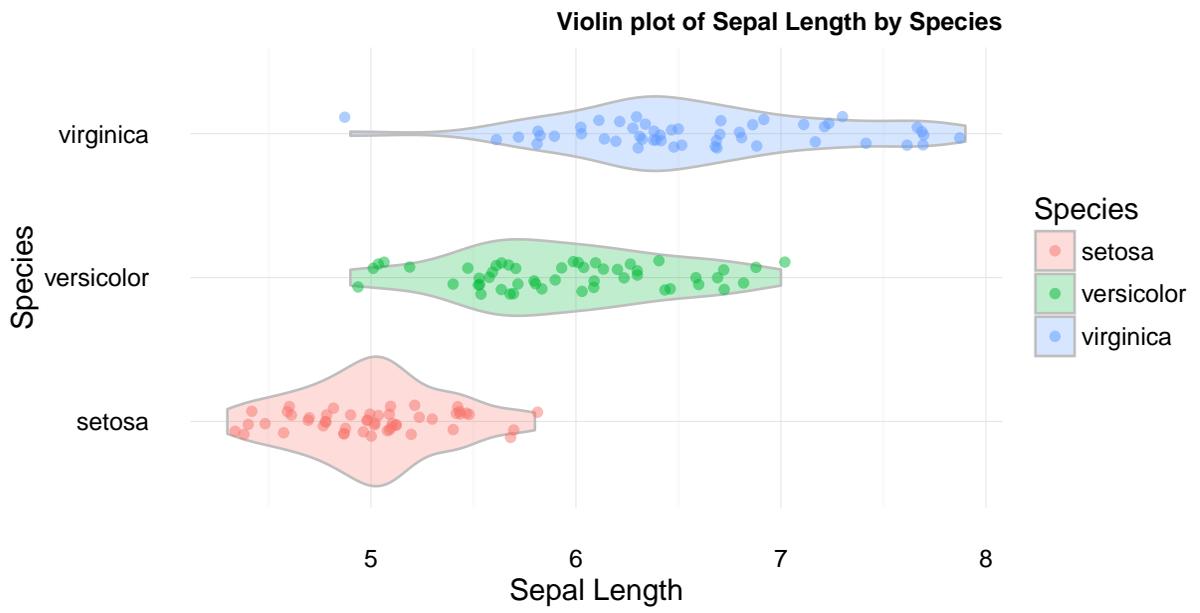
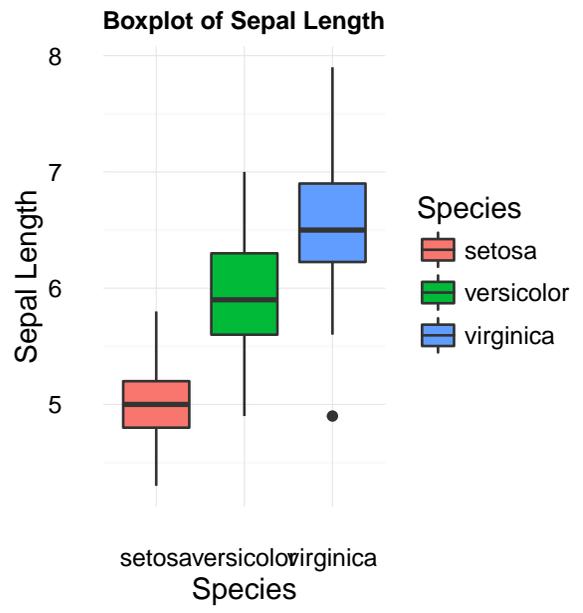
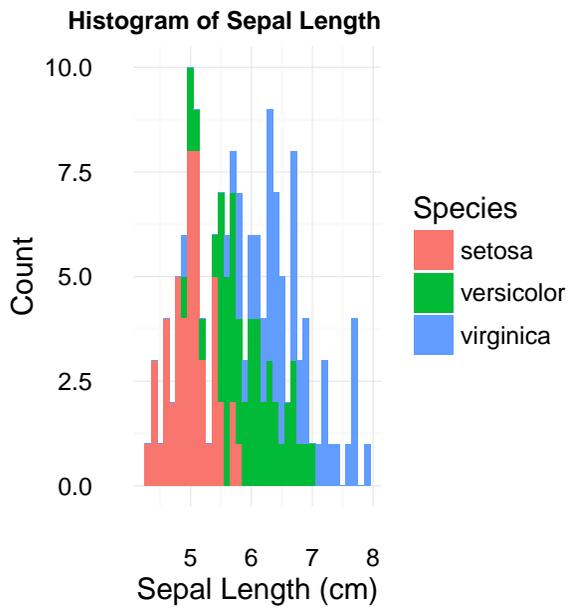
```
# Histogram of Sepal Lengths, separated by Species
p1<-ggplot(data = iris, aes(x = Sepal.Length, fill=Species)) +
  geom_histogram(binwidth = .1) +
  xlab("Sepal Length (cm)") + ylab("Count") +
  ggtitle("Histogram of Sepal Length") +
  theme_minimal() +
  theme(plot.title = element_text(size=10, face = "bold", hjust = 1))

# Boxplots of Sepal Length, separated by Species
p2<-ggplot(data = iris, aes(x = Species, y = Sepal.Length, fill=Species)) +
  geom_boxplot() +
  xlab("Species") + ylab("Sepal Length") +
  ggtitle("Boxplot of Sepal Length") +
  theme_minimal() +
  theme(plot.title = element_text(size=10, face = "bold", hjust = 1))

# Violin plot of Sepal Length, separated by species
p3<-ggplot(data = iris, aes(x = Species, y = Sepal.Length, fill=Species)) +
  geom_violin(alpha=.25, color="gray") +
  geom_jitter(alpha=.5, aes(color=Species), position=position_jitter(width=0.3)) +
  coord_flip() +
  xlab("Species") + ylab("Sepal Length") +
  ggtitle("Violin plot of Sepal Length by Species") +
  theme_minimal() +
  theme(plot.title = element_text(size=10, face = "bold", hjust = 1))

# Creating a matrix that defines the layout (not all graphs need to take up the same space)
lay <- rbind(c(1,2),
             c(3,3))

# Plotting the plots on a grid
grid.arrange(p1, p2, p3, ncol=2, layout_matrix=lay)
```



8 Additional References

And finally, if your thirst for ggplot customizations are not quenched, check out these links below.

- **ggplot:** <http://ggplot2.org/>
- **Customizing ggplot**
 - Facet: [http://www.cookbook-r.com/Graphs/Facets_\(ggplot2\)/](http://www.cookbook-r.com/Graphs/Facets_(ggplot2)/)
 - Customizing Data Shapes: http://www.cookbook-r.com/Graphs/Shapes_and_line_types/
 - Formatting Text: <http://zevross.com/blog/2014/08/04/beautiful-plotting-in-r-a-ggplot2-cheatsheet-3/#quicksetup-the-dataset>
 - Customizing your Fonts: <http://zevross.com/blog/2014/07/30/tired-of-using-helvetica-in-your-r-graphics-heres-how-to-use-the-fonts-you-like-2/>
- **ggplot Themes**
 - Vignette: <http://docs.ggplot2.org/dev/vignettes/themes.html>
 - ggthemes: <https://cran.r-project.org/web/packages/ggthemes/vignettes/ggthemes.html>
 - changing elements of a theme: <http://docs.ggplot2.org/0.9.2.1/theme.html>
- **Arranging Graphical Objects (grobs) on the page**
 - Arrange Grob: <https://cran.r-project.org/web/packages/gridExtra/vignettes/arrangeGrob.html>