Data Visualization: Basic Visuals

Required packages:

tidyverse: Organize and visualize data.
readxl: Read in excel documents.

First, install the packages that we will be using with the install.packages() function. (Remove the # in front of each line to install the packages).

```
# install.packages("tidyverse")
# install.packages("readxl")
```

Now load the packages that you installed.

```
library(tidyverse)

library(readxl)
```

Loading in a dataset

Load in the iris dataset from base R, you can load it in using the data() function.

```
data(iris)
```

You can now see in the upper right pane that the iris dataset is promised.

Looking at the iris dataset

Using the View() function, you can look at the full dataset to decide what to visualize.

```
View(iris)
```

The iris dataset focuses on three iris species and their petal length and width and sepal length and width. It's important to notice that the variables in this dataset are capitalized and separated by periods. This is really important to note because R is case sensitive.

Basic visualization

All visualization in ggplot follows a basic formula where you:

1. Reference the package
   ggplot()
2. Call in your dataset
   ggplot(data)
3. Tell ggplot the plot aesthetics (your x and y values, color, shape, transparency, etc.)
   ggplot(data, aes(x=…, y=…))

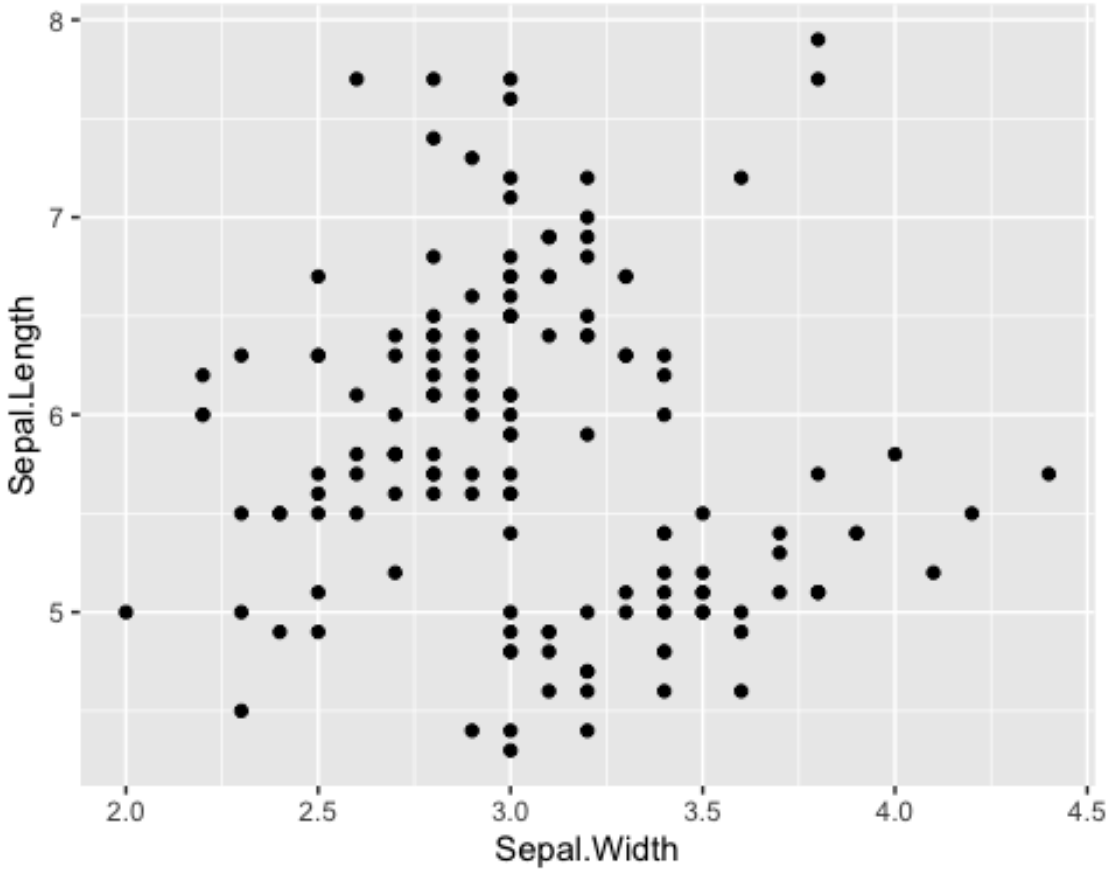
4. Add your visualization type
   ggplot(data, aes(x=..., y=...)) + geom_...

This is the RStudio cheatsheet for ggplot:
https://raw.githubusercontent.com/rstudio/cheatsheets/main/data-visualization.pdf

Scatterplots

Create a simple scatter plot using the iris dataset by plotting sepal width against sepal length.

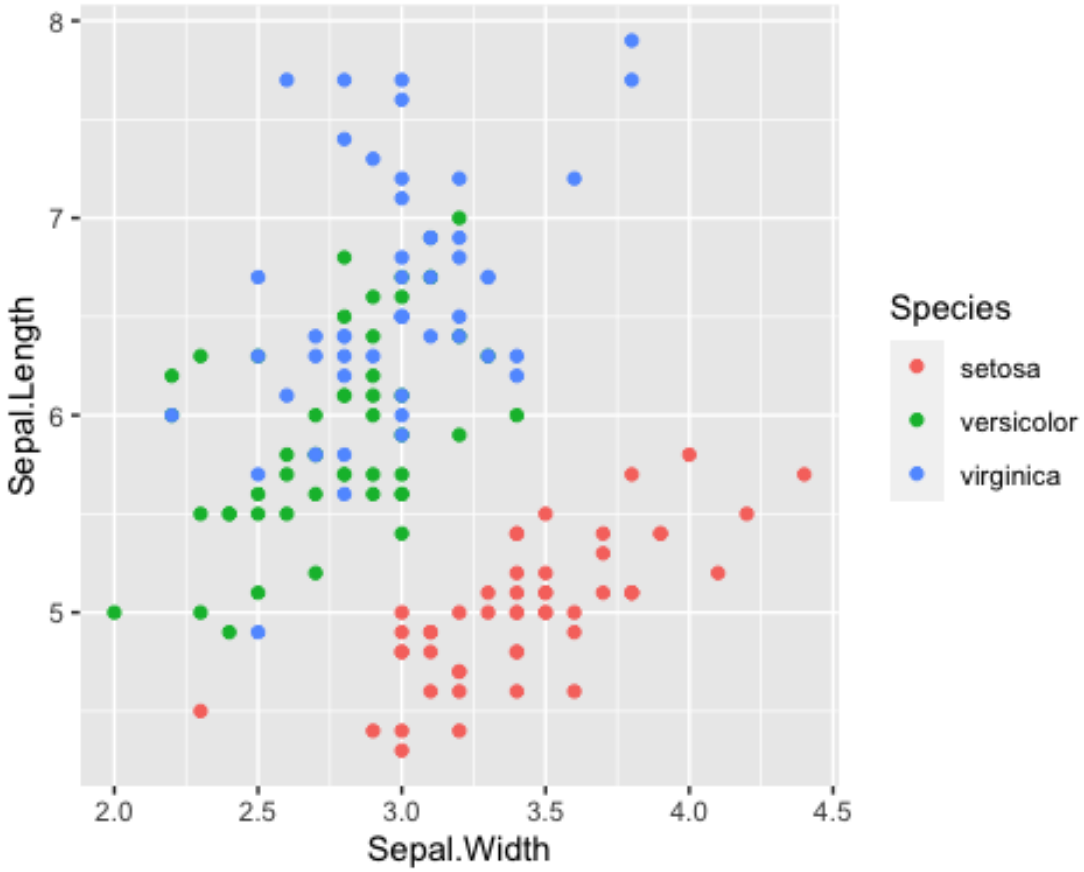
```
ggplot(iris, aes(x=Sepal.Width, y = Sepal.Length)) +
  geom_point()
```



This code gives us a basic scatter plot.

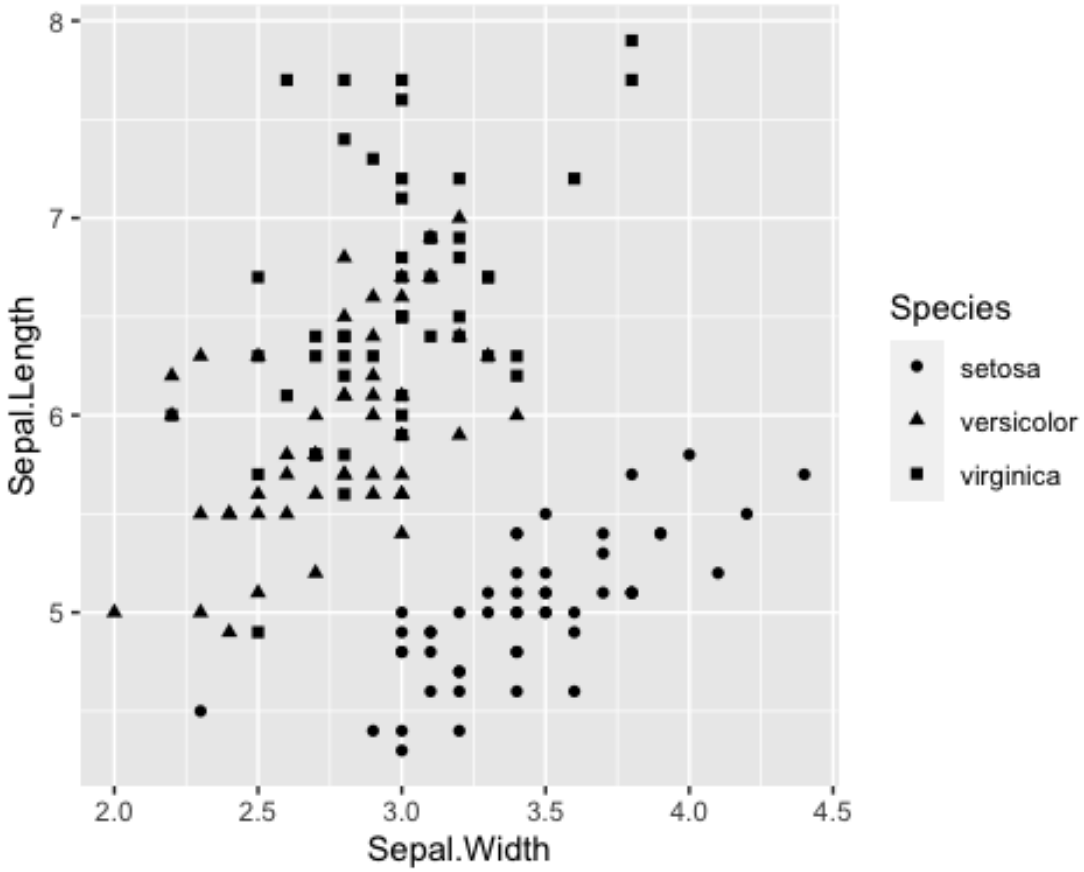
You can add extra dimensions to the aesthetic statement to show data categories and glean more information like changing the color for each species.

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



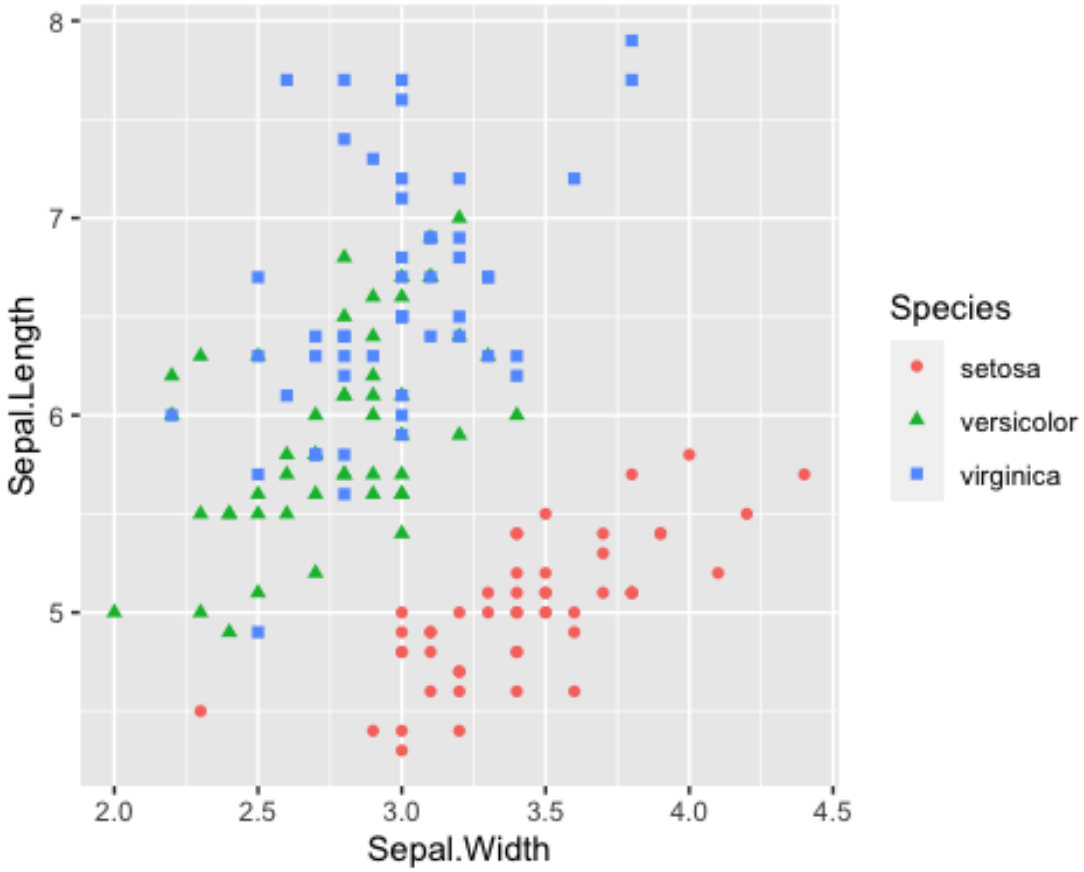
You could also change the shapes that corresponds with each species.

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



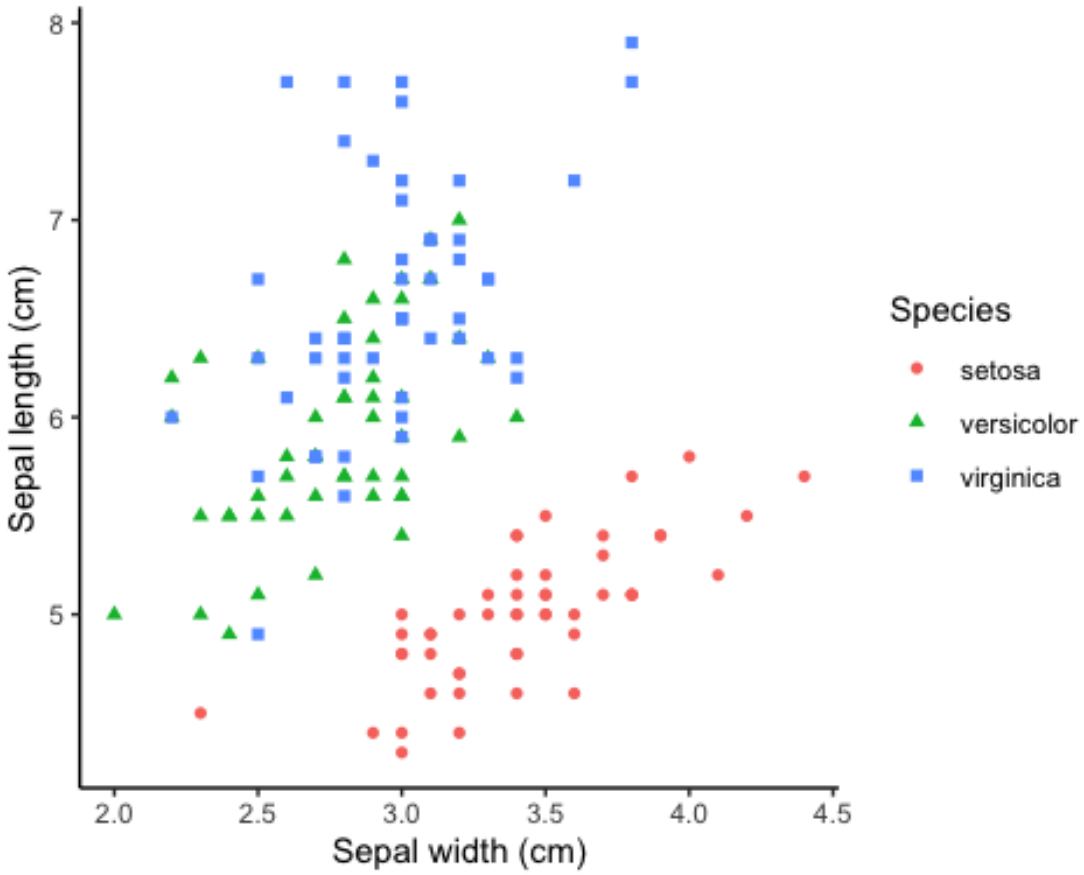
You can even do both.

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



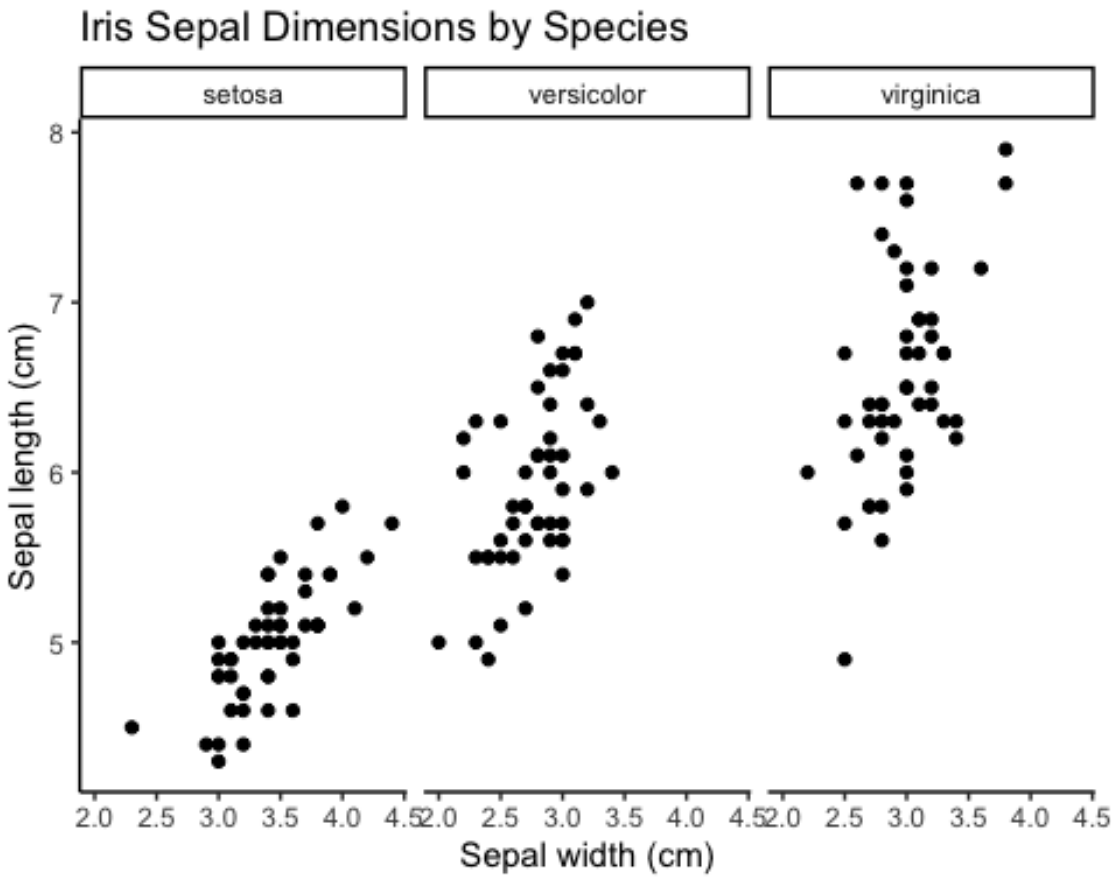
To make our figure a little sleeker, you can add a different theme or change the axis labels.

```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length, color = Species, shape = Spec
ies)) +
  geom_point() +
  theme_classic() +
  labs(x="Sepal width (cm)", y="Sepal length (cm)")
```



One other useful tool is to break apart the data into multiple figures using facet_grid, which may make differentiation by shape or color unneccessary. In this case, split the data up by species.

```
ggplot(iris, aes(x=Sepal.Width, y=Sepal.Length)) +
  geom_point() +
  theme_classic() +
  labs(x="Sepal width (cm)",
       y="Sepal length (cm)",
       title = "Iris Sepal Dimensions by Species") +
  facet_grid(.~Species)
```



From this, you may be able to glean more information, see trends between species, and determine what to analyze or visualize next.

Practice exercise

Practice visualizing data by plotting petal width by petal length in a single plot with species designated by color.
What conclusions can you draw form this figure?

Other common visualization techniques

Two of the most common figure types used in academic settings are boxplots and bar graphs, geom_boxplot() and geom_bar(), respectively. Boxplots and bar graphs are good for datasets with qualitative variables. While the iris dataset does have species as a qualitative variable, it is also good practice to read in and work with new datasets.

Read in the cucumber yield trial dataset.

```
cuke <- read_excel("~/Documents/cucumbers.xlsx")
```

Start by looking at the data using the summary function. This can give you an idea of the types of data in your dataset and the spread of values within each variable.

```
summary(cuke)
```

```
##       loc           rep         variety             yield
##  Min.   :1.0   Min.   :1.00   Length:32          Min.   :11.50
##  1st Qu.:1.0   1st Qu.:1.75   Class :character   1st Qu.:28.62
##  Median :1.5   Median :2.50   Mode  :character   Median :36.64
##  Mean   :1.5   Mean   :2.50                      Mean   :35.74
##  3rd Qu.:2.0   3rd Qu.:3.25                      3rd Qu.:43.50
##  Max.   :2.0   Max.   :4.00                      Max.   :61.48
```

Based on the summarized data, location, replication, and variety are qualitative variables and should be changed to factors.

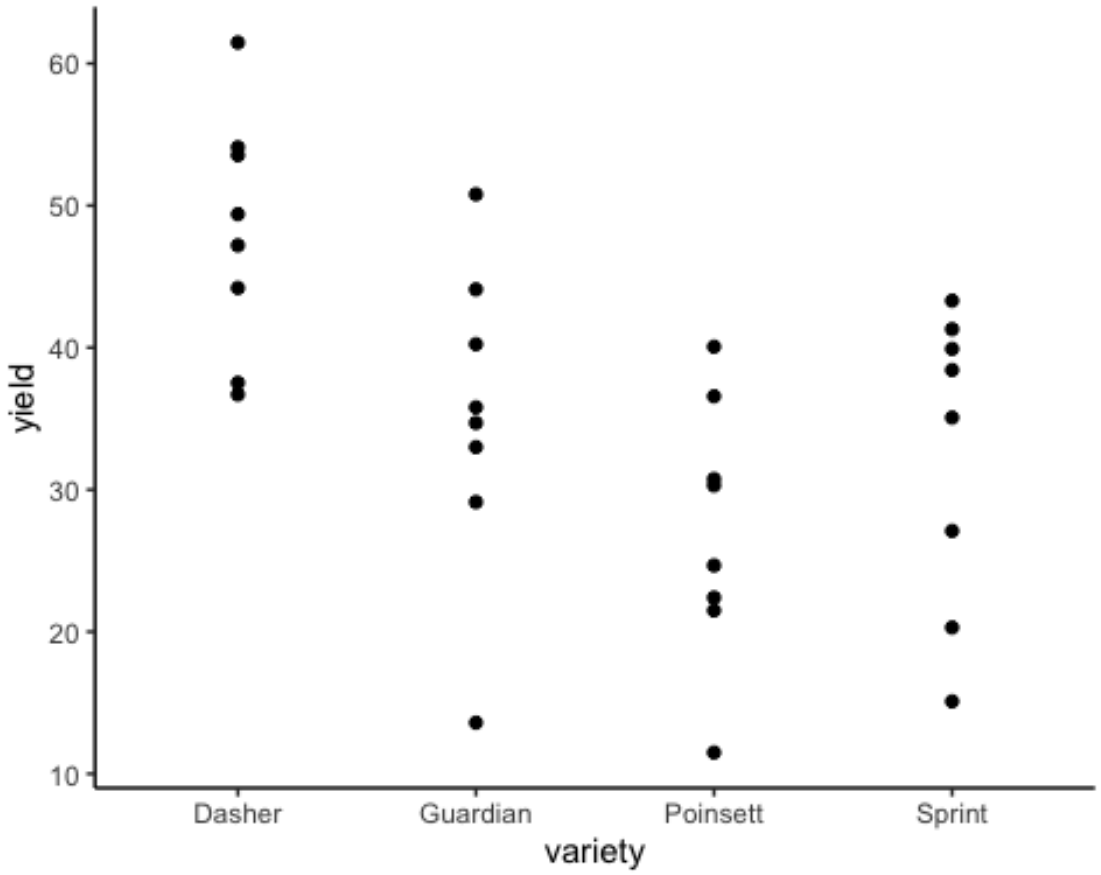
```
cuke$loc <- as.factor(cuke$loc)
cuke$rep <- as.factor(cuke$rep)
cuke$variety <- as.factor(cuke$variety)
summary(cuke)
```

```
##  loc    rep       variety      yield
##  1:16   1:8   Dasher  :8   Min.   :11.50
##  2:16   2:8   Guardian:8   1st Qu.:28.62
##         3:8   Poinsett:8   Median :36.64
##         4:8   Sprint  :8   Mean   :35.74
##                            3rd Qu.:43.50
##                            Max.   :61.48
```

You can see that the data is now grouped into distinct categories rather than continuous numbers or character variables.

Begin by plotting the data as a scatterplot.

```
ggplot(cuke, aes(variety, yield)) +
  geom_point() +
  theme_classic()
```

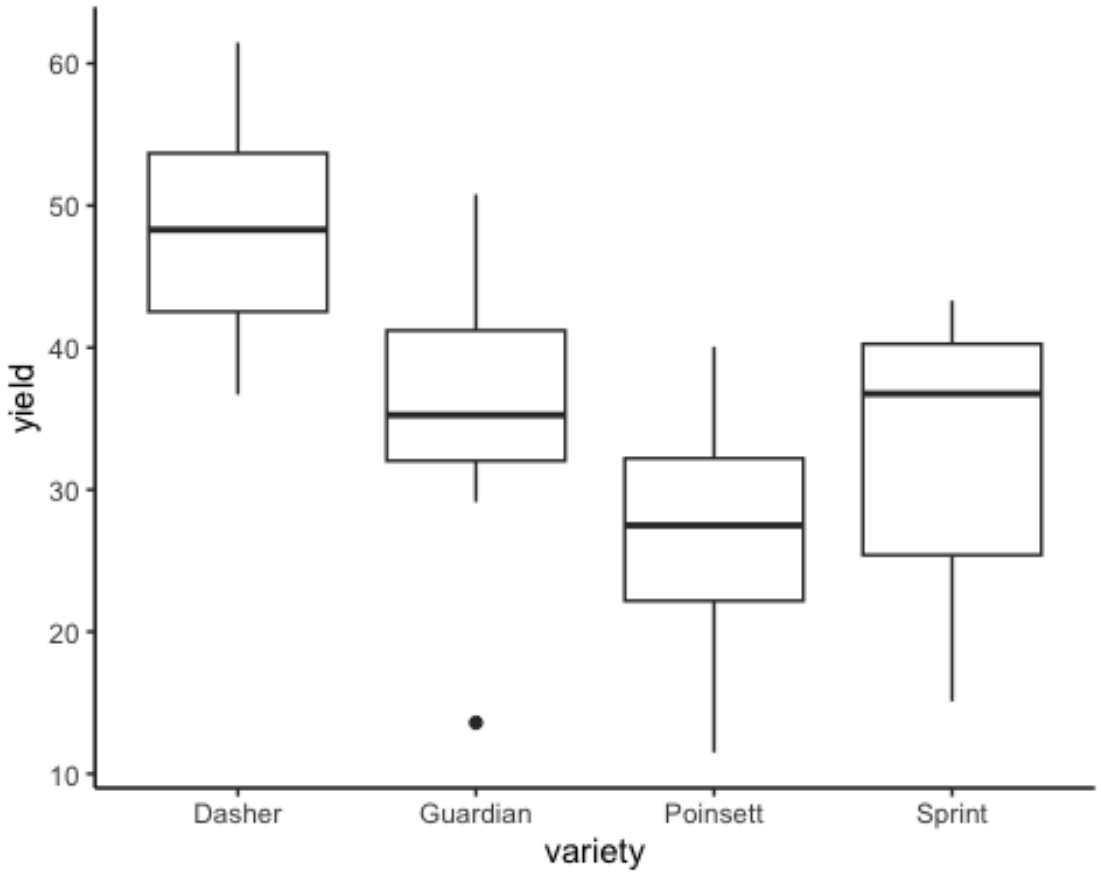


Seeing as this data is categorical, a scatterplot may not be the most effective way to look at the data.

Boxplots

Try creating a simple boxplot of yield by cucumber variety.

```
ggplot(cuke, aes(variety, yield)) +
  geom_boxplot() +
  theme_classic()
```



Bar graphs

Another common visualization is a bar graph. Bar graphs are interesting, before visualizing the data you need to summarize it otherwise ggplot will generate a plot that adds all the values together for the bar graph value. For example, the yield for the Dasher cucumber variety would be in the hundreds rather than average value of about 50, based on the boxplot.

To summarize the data, use the group_by() and summarize() functions. In this case, you will want to group the data by variety and create a new variable to generate the mean yield for each variety.

```
cuke.summary <- cuke %>%
  group_by(variety) %>%
  summarize(mean.yield = mean(yield))
cuke.summary
```

```
## # A tibble: 4 × 2
##   variety  mean.yield
##   <fct>         <dbl>
## 1 Dasher         48.0
## 2 Guardian       35.2
```

```
## 3 Poinsett       27.2
## 4 Sprint         32.6
```

After running the new summary tibble, you can see that there is now one value for each variety that corresponds to the mean yield value. You can add any other calculations to the summary table that you may want to add to your figure such as standard deviation or standard error.

Now plot the cucumber data in a bar graph. Make sure to include the stat = "identity" argument in the geom_bar() function to ensure that the value show in your bar graph matches the value in your dataset. Also be sure to call from the summary table rather than the original cucumber dataset.

```
ggplot(cuke.summary, aes(variety, mean.yield)) +
  geom_bar(stat="identity") +
  labs(x="Variety",
       y="Yield",
       title = "Cucumber Yield by Variety") +
  theme_classic()
```