Lecture 5

Paul Holaway, Abhi Thanvi

June 28th, 2022

Lecture 4 Review

Before we move on to the new material, we will do a quick review of Lecture 4 content. Last time we went over what experimental design is and how to do it. We also compared the two types of studies, experimental and observational. Recall:

1. **Experimental:** We investigate the effects of certain conditions on individuals or objects in the sample.
2. **Observational:** We observe the response for a specific variable for each individual or object.

We learned that experimental is better because with an experimental study we have all three aspects of a study.

1. Comparison
2. Randomization
3. Replications

Remember that randomization helps us lower both bias and variation in the data, which gives the best results possible. Also, to ensure as little bias and variation as possible, take as large a sample as possible. You will usually never be able to completely eliminate bias and there will always be variation in the data. This means you need to make sure you sample correctly and randomly so your analysis is the best and most accurate it can be. This is also why you want to have a study one can replicate. It is possible that by pure accident, you get a biased and/or high variation sample. That way someone else can do your experiment allowing you to compare results.

Example 1; Study Design Review

Many comforters contain both white feathers and down in order to provide a warm, soft cover. A bed and bath company would like to expand its line of products and sell comforters for queen and king size beds. Before manufacturing begins, a random sample of comforters is obtained from other companies and the proportion of white feathers, down, and other components are measured and recorded. These data will be used to determine the exact mixture of feathers and down for the new line of comforters.

Question 1: Is this an observational or an experimental study?

Answer: Observational: The stores that the comforters come from are **NOT** randomly selected. They are chosen beforehand. Even though the comforters are randomly selected, the stores that they come from are not. For a study to be experimental, every part of the selection process must be random.

Question 2: How could we change this sampling procedure to make the study experimental?

Answer: We simply need to randomly select the stores that the comforters come from. This can be done by obtaining a list and taking an SRS or we could divide the stores into strata by size, location, etc. and then take an SRS from those strata. We would still need to randomly select the comforters from each store like the original study.

Question 3: Now let's randomly select the stores. Let's say there are 9 other stores in the area that sell comforters. The following code below makes a list of them.

```
stores = c("Kohls","Bed, Bath, & Beyond", "Mattress Warehouse", "Meijer", "Walmart", "Sam's Club", "Sle
```

Your boss because of financial and time constrains, says to only look at three of them. So let's select three stores randomly.

```
set.seed(143572)
store_sample = sample(stores, size = 3, replace = FALSE)
store_sample
```

```
## [1] "Meijer"              "Bed, Bath, & Beyond" "Macy’s"
```

Notice here how the code for taking the sample is different than last time. This is because before we were randomly sampling a data frame, here we are just sampling a list. The code is much simpler when sampling from a list because there is less work for `RStudio` to do. The code for future reference is `sample(list_name, size = n, replace = ...)`.

Okay, now we can move onto the next portion of lecture content.

Plots and Sample Space

Plots

Today we are going to cover one of the most useful topics in data science, making plots. Most people (including me) cannot simply read through a massive chunk of text and absorb all of the information the first time. It also gets really boring really quick. Plus, most people will not even attempt to read something that is a massive wall of text. That is where creating plots comes it. It is a way that you can express the same results in a simpler, easier, and paper-saving way.

A Note About Plotting in `RStudio`

The ultimate champion for plotting in `RStudio` is the package `ggplot2`. This package contains tools to make some amazing graphs and charts. However, it is also an extremely complicated package to use for beginners. For this class, we will just be using the basic `RStudio` plots. These are still excellent quality graphs and are much easier to use for beginners. We do not want to overwhelm you. If you want to take a look into `ggplot2`, there is a cheat sheet for it on Blackboard. You may play around with it in your spare time.
Let's look at making these kinds of plots using the `MLB` data set. This data set contains basic season statistics from MLB teams for the 2015-2019 seasons.

Histograms

Histogram: A diagram consisting of rectangles whose area is proportional to the frequency or proportion of a variable and whose width is equal to the interval size.

Histograms are useful because they can tell us how often something occurs. Whether it be the number of times it appears in the data set or the proportion it appears in the data set. The code for a basic histogram is `hist(data$x)`, where `x` is the variable you want to look at.

Example 2; Number of Wins Histogram

Let's say you wanted to know the frequency of the number of wins for the data.

```
library(tidyverse)
```

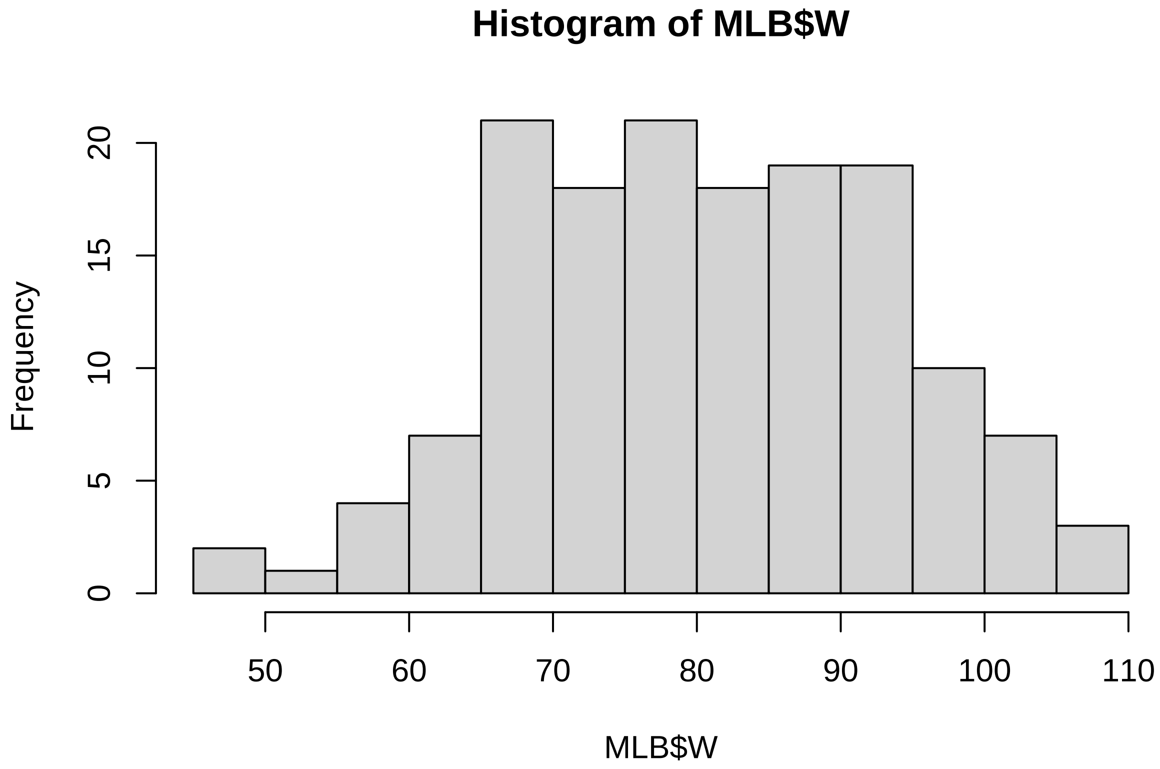
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
MLB <- read.csv("~/Desktop/DPISu22/Data Sets/MLB.csv", stringsAsFactors=TRUE)
```

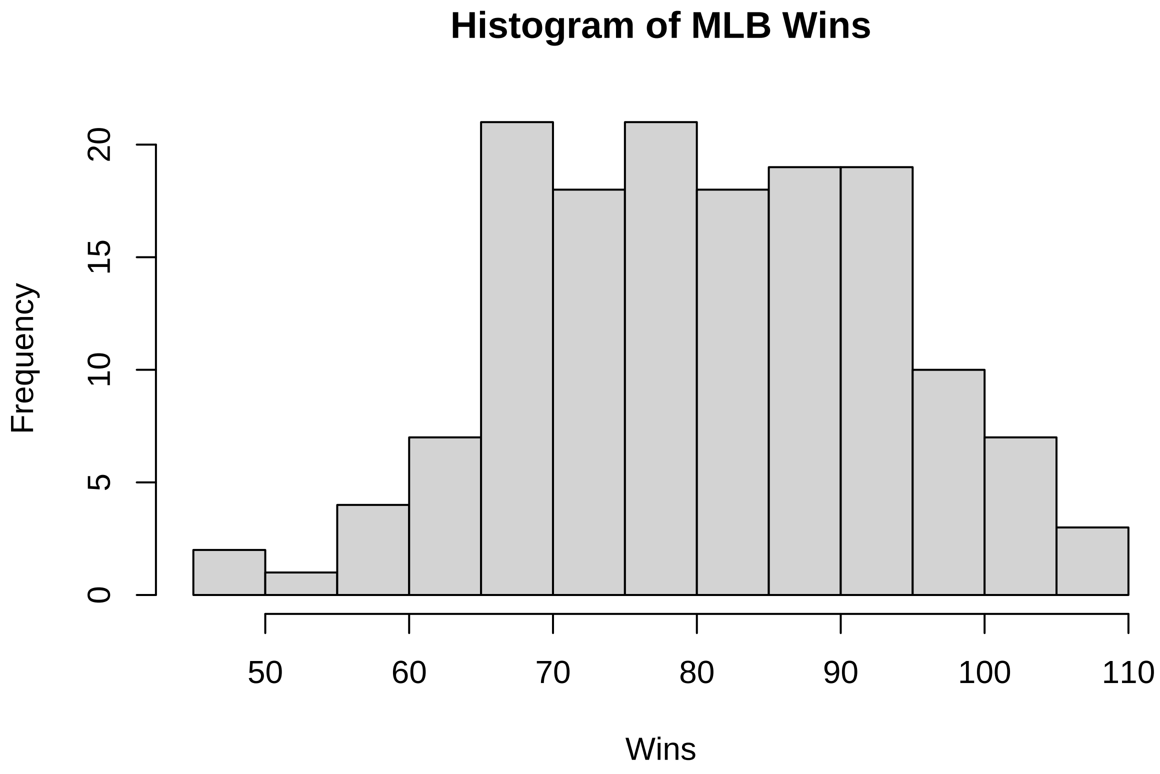
```
hist(MLB$W)
```



How you read this is as follows. Notice how the scale on the x-axis is in increments of 10 and you have two bars per 10. This means that your bars (or bins in statistical terms) encompass 5 wins (60-64,65-69, etc.). Then you look at the height of it using the y-axis frequency scale. You could make a bin for each category, but the problem is then your histogram is really hard to read because the bins are too small (take my word for it).

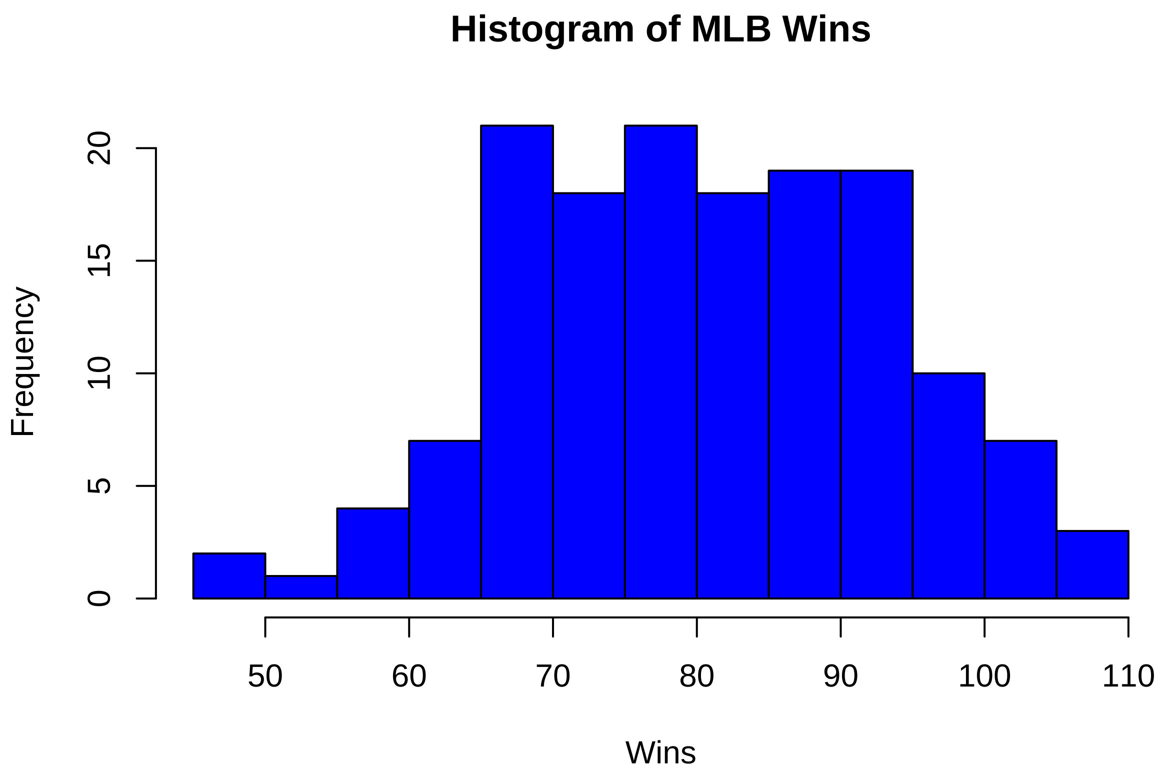
Now the histogram is nice looking, but it is a bit un-intuitive for those who do not know baseball. You know that `MLB$W` is the number of wins because you worked with the data set, but your readers may not. Let's re-label the x-axis and main title to make things more intuitive for your readers. This can be done by adding in `xlab = ...` and `main = ...` in the `hist()` function.

```
hist(MLB$W,xlab = "Wins",main = "Histogram of MLB Wins")
```



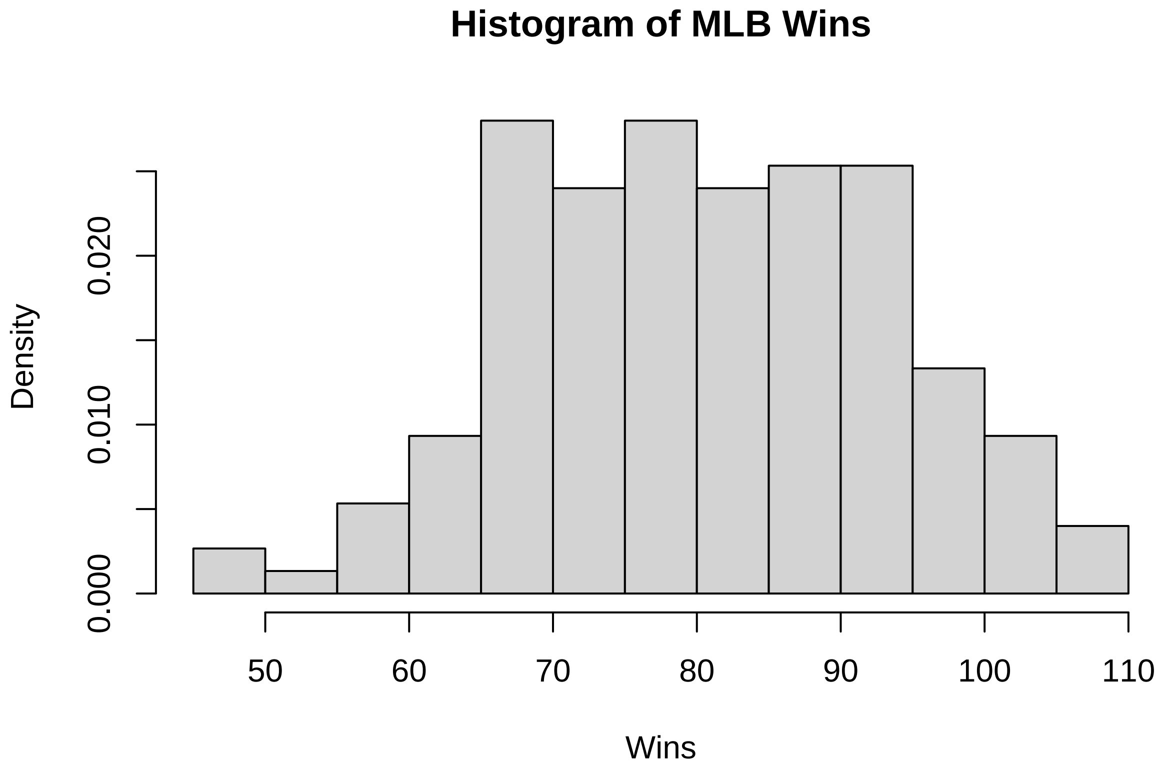
There, much better. We can also change the color of the bars too if we want.

```r
hist(MLB$W,xlab = "Wins",main = "Histogram of MLB Wins", col = "blue")
```



Now what if we want to look at the proportion of the number of wins happening? This can also be done using a histogram, but we need to make one small change to our code. We will add `probability = TRUE` into our `hist()` function.

```r
hist(MLB$W, xlab = "Wins", main = "Histogram of MLB Wins", probability = TRUE)
```



By doing this, we are creating what is called a probability histogram. Notice how the only thing that changed between the probability histogram and the other (frequency) histogram, is that the y-axis is now a percentage of how often the range occurs.

Box Plots

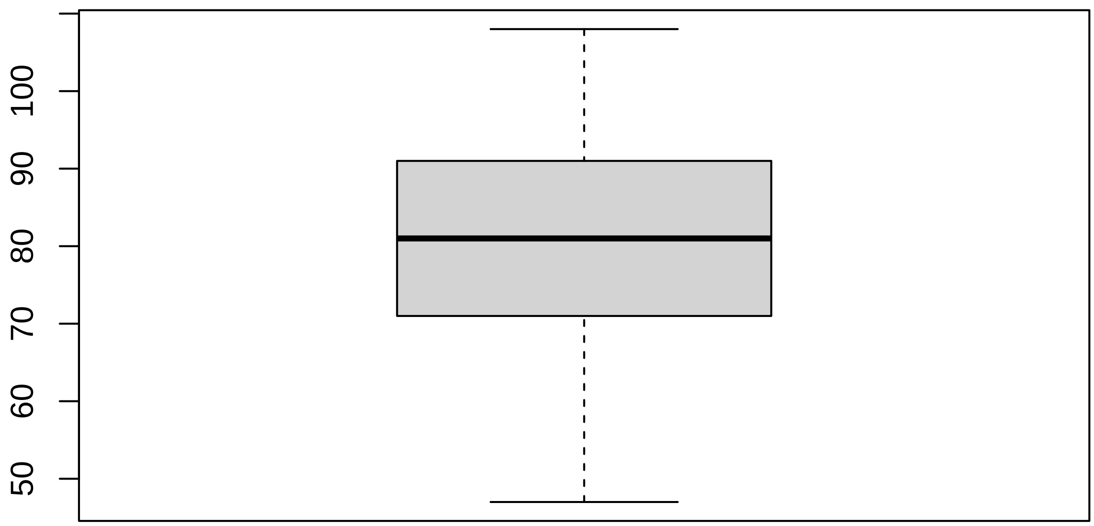
Box Plot: A method for graphically demonstrating the locality, spread, and skewness for groups of numerical data.

A box-plot is nice because it allows you to look at some things that a histogram cannot. While a histogram shows us the spread of the data, a box plot can give us some more details, such as where the majority of the data is, potential extreme values (outliers), and where the median and average of the data is. Let's do this by making a box plot for Wins.

Example 3; Number of Wins Box Plot

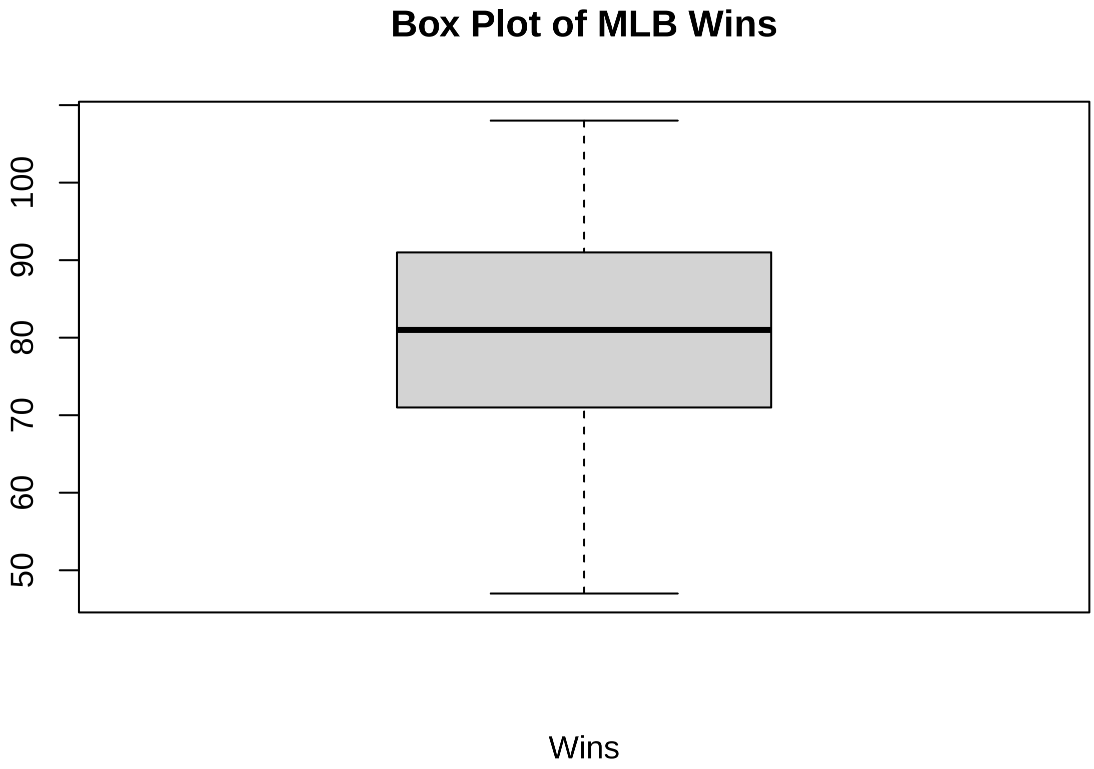
Let's start by making a simple box plot for Wins. The code for this is simply `boxplot(data$x)` where again, `x` is the variable you wish to plot.

```
boxplot(MLB$W)
```



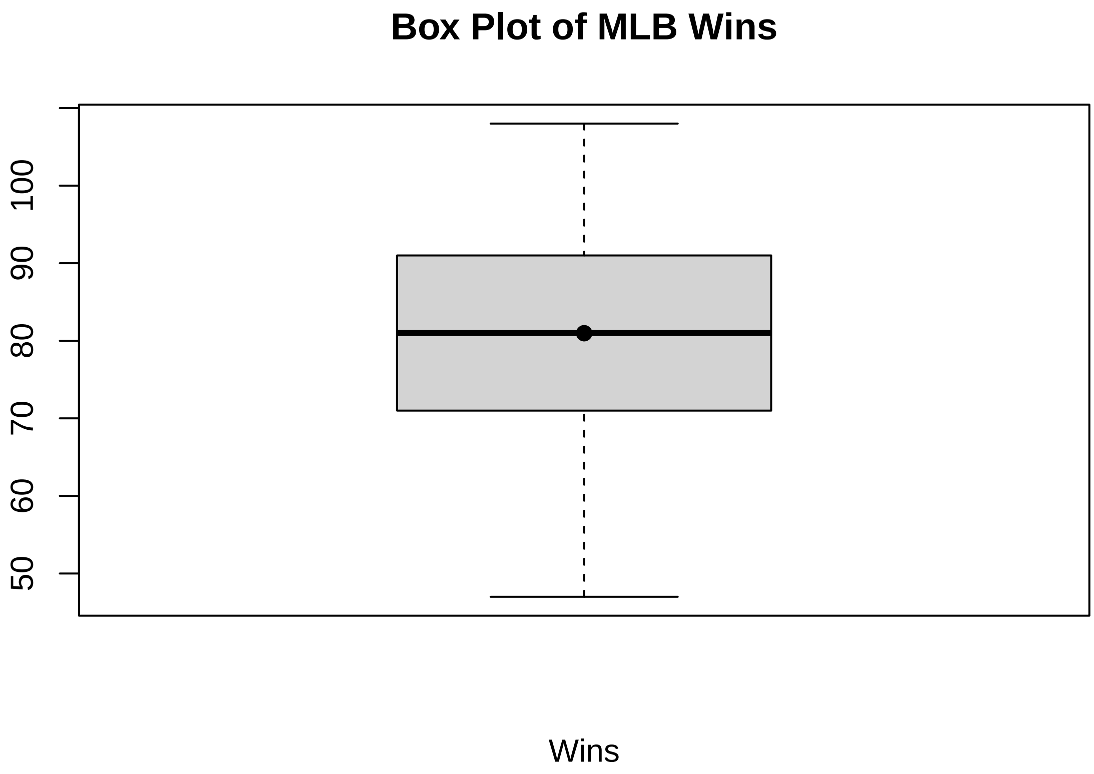
Okay, but you can see there are no labels and the mean is not there. We know how to fix the labels...

```
boxplot(MLB$W, xlab = "Wins", main = "Box Plot of MLB Wins")
```



Now we just need to add the mean. This can be done using the following code, `points(mean(data$x),col="black",pch=19)`. Put this line of code *after* your box plot code.

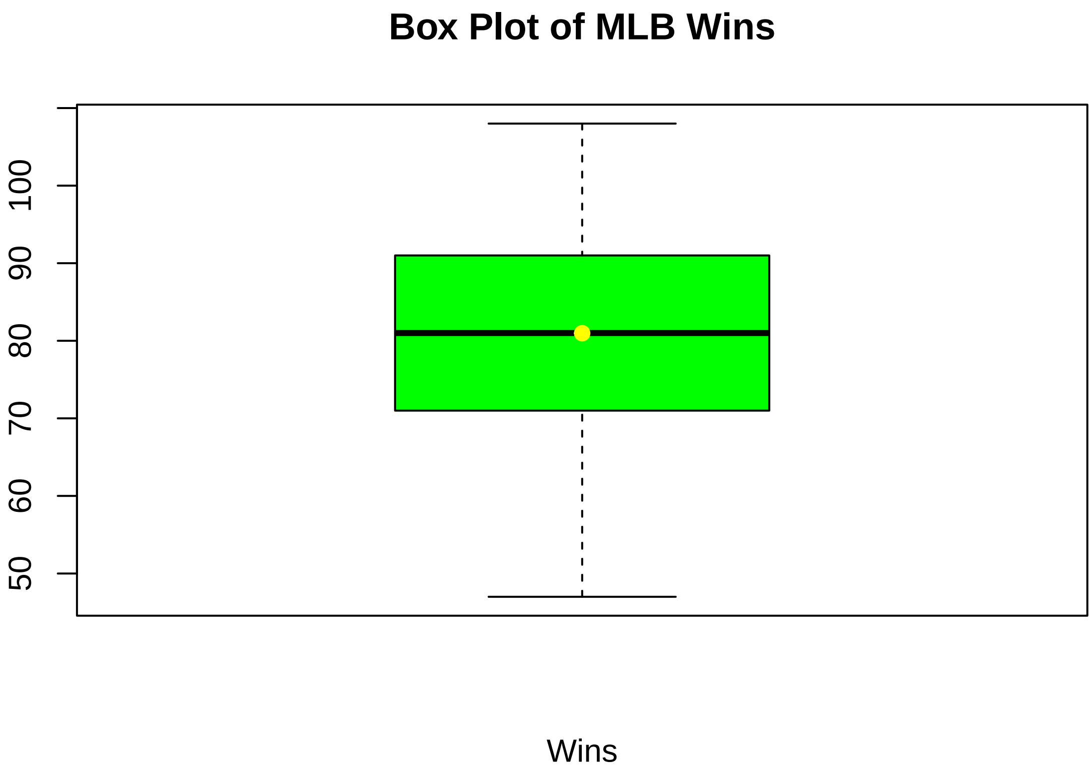
```
boxplot(MLB$W, xlab = "Wins", main = "Box Plot of MLB Wins")
points(mean(MLB$W),col="black",pch=19)
```



How you read this is as follows. The majority of the data is wherever the range of the box is. So here, most teams win between 70-90 games. The solid black line in the box is the median (mid-point) of the data. In this case it looks to be about 80. The solid black dot is the average of the data (also looks to be about 80). The dotted lines extending from the box are called the "whiskers" and tell us where the data should be falling. Any points outside that range, are extreme values. These extreme values could skew your results. We will not cover how to determine if they are skewing your results in this course, but upper level statistics courses will.

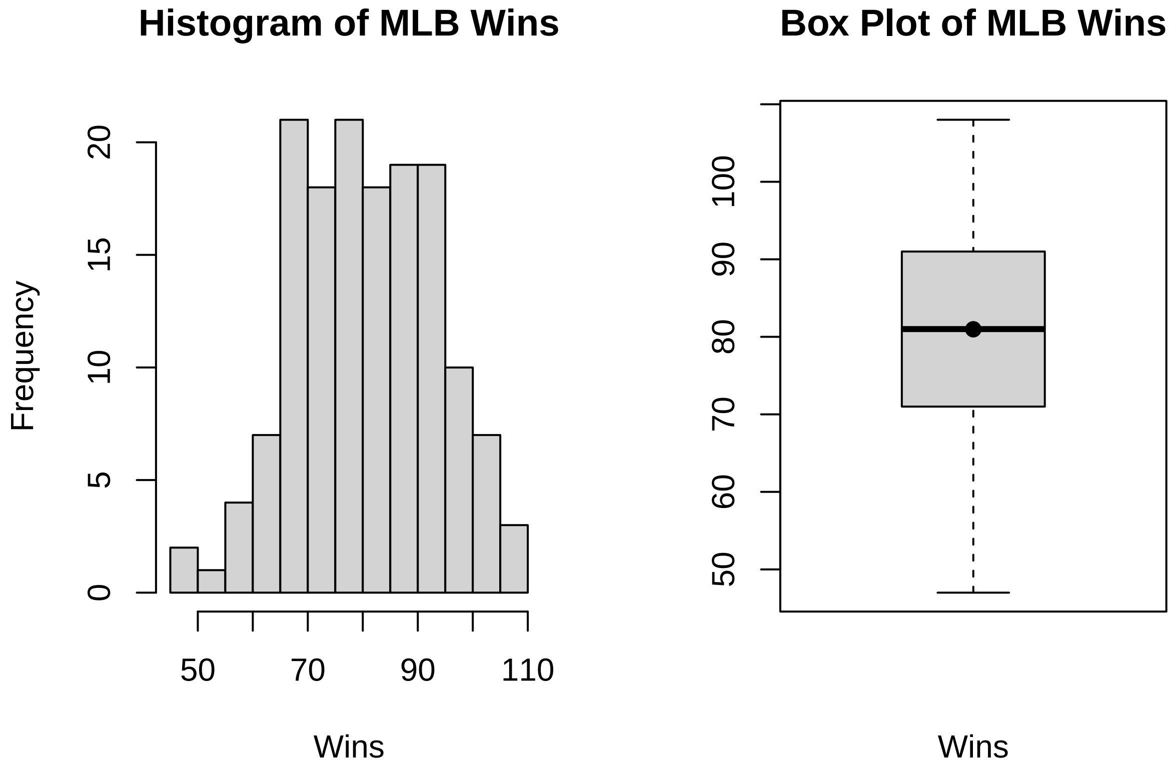
You can also change the colors here too.

```
boxplot(MLB$W, xlab = "Wins", main = "Box Plot of MLB Wins", col="green")
points(mean(MLB$W),col="yellow",pch=19)
```

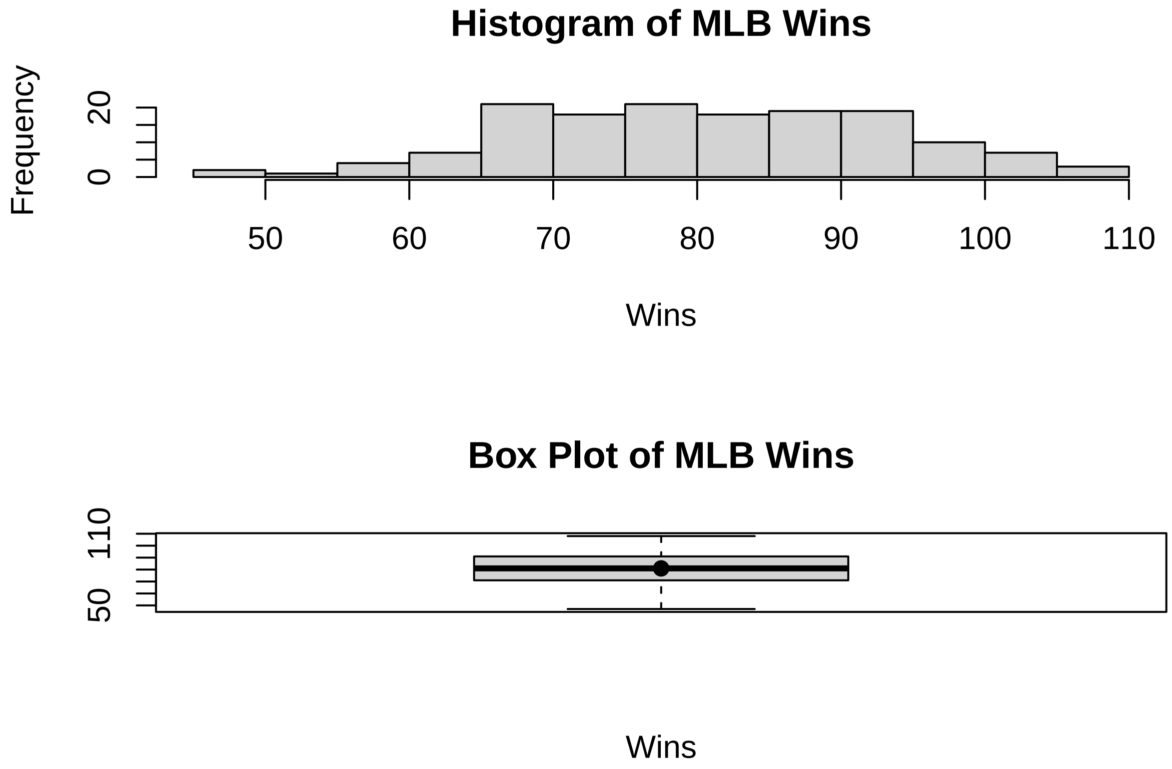


Now let's compare the histogram and box plot next to each other. The problem though is you can only view one at a time, or can you? There is actually a neat trick you can use to view more than one plot at the same time. You can use the `par()` function which changes graphical parameters. You will have to specify how you want the viewing to change. By default, `RStudio` prints out only one graph at a time. If you want to change it, then you will have to do the following. . . `par(mfrow=c(m,n))`. `m` determines the number of display rows while `n` determines the number of display columns. Below is an example using the two plots. You will always have to put the `par()` function at the beginning of each cell you use it in.

```
par(mfrow = c(1,2))
hist(MLB$W,xlab = "Wins",main = "Histogram of MLB Wins")
boxplot(MLB$W, xlab = "Wins", main = "Box Plot of MLB Wins")
points(mean(MLB$W),col="black",pch=19)
```



```
#Alternative Display Method
par(mfrow = c(2,1))
hist(MLB$W,xlab = "Wins",main = "Histogram of MLB Wins")
boxplot(MLB$W, xlab = "Wins", main = "Box Plot of MLB Wins")
points(mean(MLB$W),col="black",pch=19)
```



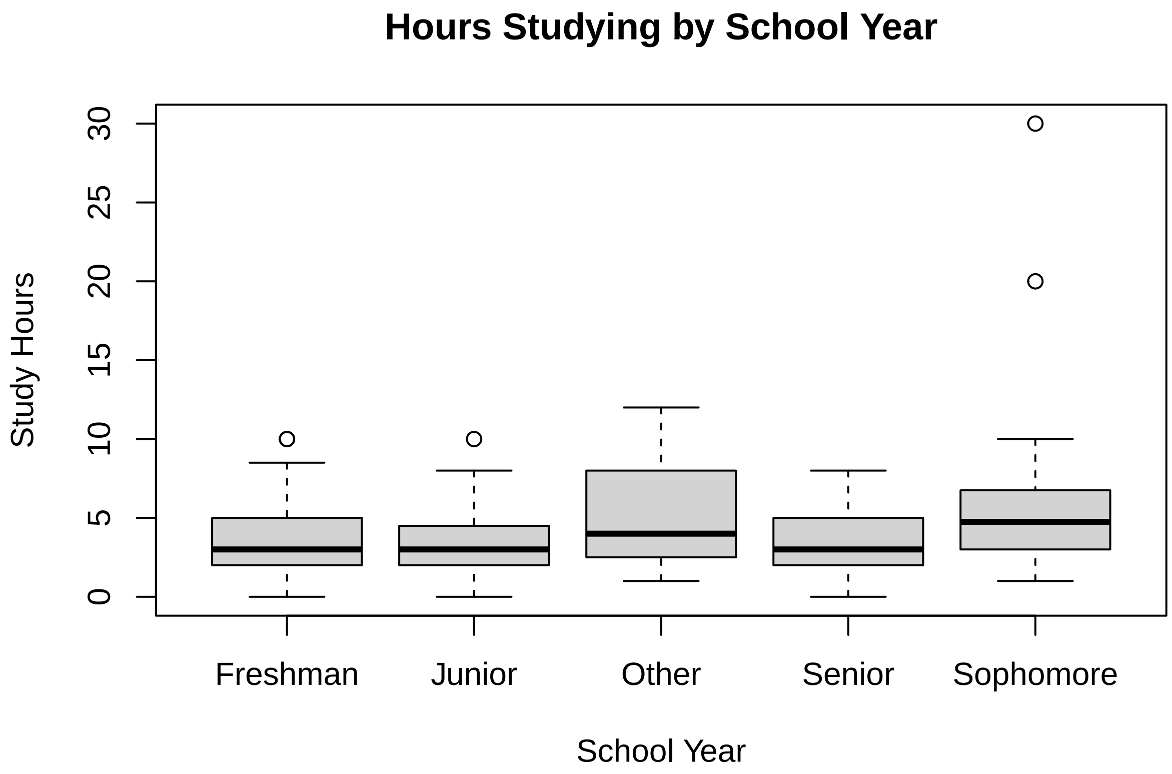
Notice how the first display looks much nicer than the second. Sometimes it is better to display one way, sometimes the other, or sometimes it does not matter and is your choice. It is up to your discretion how you display the plots.

Example 4; Side-by-Side Box Plots

Let's now make switch to the `hello` data set. You are doing a project and are wondering how the different years are effecting student study time. Do students who have been there longer study less, more, the same as younger students? You could do this by filtering each year and make four smaller data sets, then make a box plot for each, and put them together. However, that is a lot of work to do for one plot. There has to be an easier way right? There is, and it is also a simple adjustment to the code you already know. The code for this is `boxplot(data$x ~ data$y)` where `x` is the variable you want to investigate, and `y` is the variable that has the different group labels. Note that this method only works in this circumstance. If you want to make a side by side box plot for Wins and Losses, you will have to use the `par()` function and make two box plots. Let's do this now.

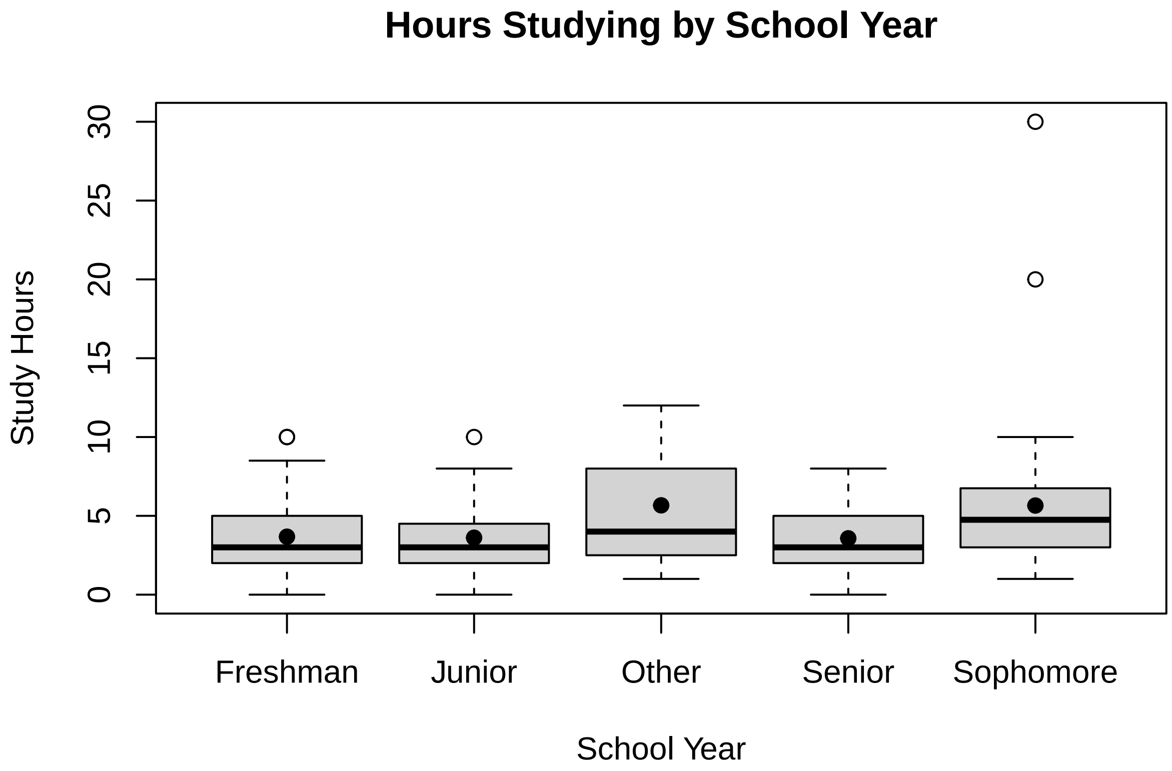
```
hello <- read.csv("~/Desktop/DPISu22/Data Sets/hello.csv", stringsAsFactors=TRUE)
```

```
boxplot(hello$Study.Hours ~ hello$Year, xlab = "School Year", ylab = "Study Hours",
        main = "Hours Studying by School Year")
```



If you wish to add the means for the different groups, we have to get fancy with the code. I will just give you this since it is beyond the scope of this course. Feel free to copy and change it as you see fit in your project.

```
boxplot(hello$Study.Hours ~ hello$Year, xlab = "School Year", ylab = "Study Hours",
        main = "Hours Studying by School Year")
means = tapply(hello$Study.Hours,hello$Year,mean) #Code for means in side-by-side box plot.
points(means, col="black",pch=19)
```



Scatter Plots

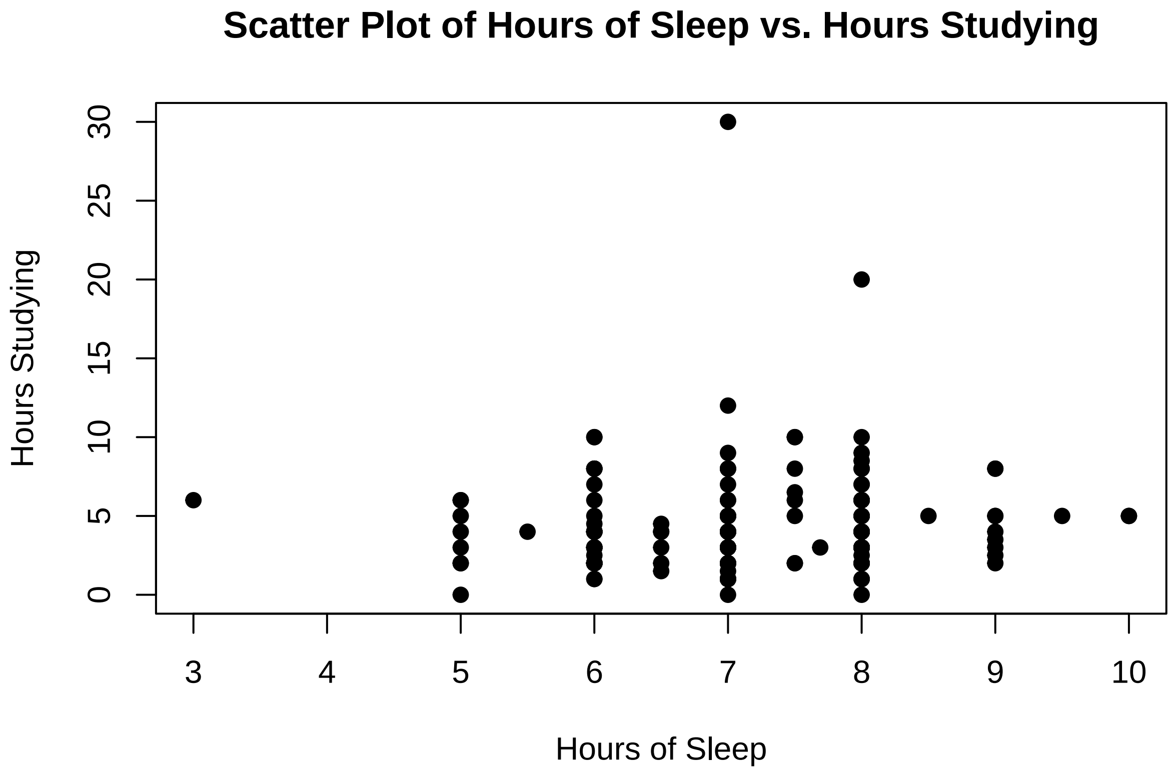
Scatter Plot: A graph in which the values of two variables are plotted along two axes, the patter of the resulting points revealing any correlation present.

Basically, we can use scatter plots to easily see if two variables are correlated. This is useful because it can save us time. If we see two variables are not correlated, then we do not have to waste our time looking for a causation that will not exist. Scatter plots are made using the following syntax, `plot(data$x,data$y)`. Where `x` is the variable you want on the x-axis and `y` is the variable you want on the `y-axis`.

Example 5; Scatter Plot Introduction

For this example, let's look at a scatter plot of sleep versus study hours. I'll add in the labels to make it nicer looking. `RStudio` by default will plot the points as open circles. I personally do not like that look. The fix is to put `pch = 19` into the `plot()` function.

```
plot(hello$Sleep,hello$Study.Hours, xlab = "Hours of Sleep", ylab = "Hours Studying",
     main = "Scatter Plot of Hours of Sleep vs. Hours Studying", pch = 19)
```



From this chart, we do not really seem to see any kind of correlation here. There is no upward or downward trend. The statistical name for this is called a null plot (because there is no correlation showing).

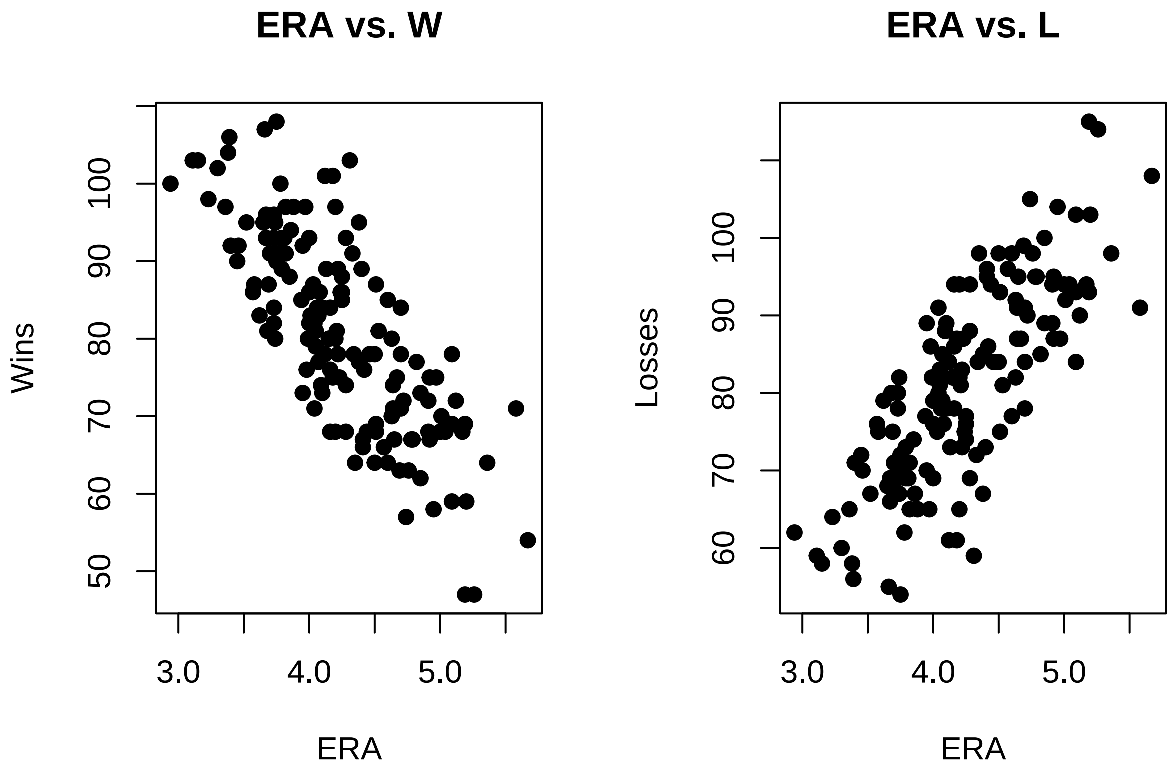
Side-by-Side

What if you want to look at multiple scatter plots at once? This answer is the same as before, we use the `par()` function.

Example 6; Looking at Multiple Scatter Plots

Let's look at two different scatter plots. One comparing Wins to ERA and another comparing Losses to ERA. We want to know how an ERA effects Wins and Losses, so we will put ERA on the x-axis.

```
par(mfrow = c(1,2))
plot(MLB$ERA,MLB$W, xlab = "ERA", ylab = "Wins", main = "ERA vs. W", pch = 19)
plot(MLB$ERA,MLB$L, xlab = "ERA", ylab = "Losses", main = "ERA vs. L", pch =19)
```



As you can see, we have a clear negative correlation between ERA and Wins, and a clear positive correlation between ERA and Losses.

Sample Space

Last time when we were doing random sampling, we were, behind the scenes, using the concept of a sample space. The sample space is an important part of probability (#next2classes), which extends to anything that uses probability, like sampling.

Sample Space: A listing of all possible outcomes *using set notation.* It is the collection of all outcomes written mathematically, with curly braces (`{ }`), and denoted by S.

This is the dictionary definition, but it can be a bit technical. Basically, the sample space is just a list of all the possible outcomes something can have. A few examples should help out.

Example 7; Sample Spaces

1. Flipping a coin

$$S = \{H, T\}$$

This is one of the most basic examples, but it gets the idea across. When you flip a coin, you only have two possible outcomes, so therefor the sample space will only have two items in it. Here, it's heads and tails.

2. Rolling a 6-sided Die

$$S = \{1, 2, 3, 4, 5, 6\}$$

This is another classic example. When rolling a d6, you can only roll a number that is 1-6. Therefor that is the sample space.

3. Flipping a Coin Twice

$$S = \{HH, HT, TH, TT\}$$

What happens if you are flipping the coin twice? This one you have to think a bit more. (It may be helpful to write all possible outcomes down for larger sample spaces.) Here, we can have both flips be heads or tails, the first flip heads and the second tails, or the first flip tails and the second flip heads. You can probably guess that sample spaces can get pretty complicated quickly. That's fine, we have `RStudio` for that part. This is just to help you understand the concept.

4. More Complex Sample Space; Color Wheel with Red, Green, and Blue Spun Twice

$$S = \{RR, RG, RB, GR, GG, GB, BR, BG, BB\}$$

Before we go, we will quickly go over some definitions that will be necessary to know when calculating probability (#next2classes).

- **Event:** Any collection (or set) of outcomes from an experiment (any subset of the sample space).
- **Simple Event:** An event consisting of exactly one outcome.
- An event has **occurred** if the resulting outcome is contained in the event.

Think of an event as a specific sequence/series/group of simple events occurring, a simple event is just one part of that sequence/series/group, and an event that has occurred if it has already happened.

End of Lecture 5 Notes