# *Creating Graphs in R (R Instructional Worksheet)*

*November 14, 2017*

## *Contents*

NOTES: DONT HAVE CHIMP DATA

## *Creating Graphs*

### *Basic Plot*

We will plot the chimpanzee and lion population numbers.

Creates a plot with years on the x-axis and chimpanzee population size on the y-axis.

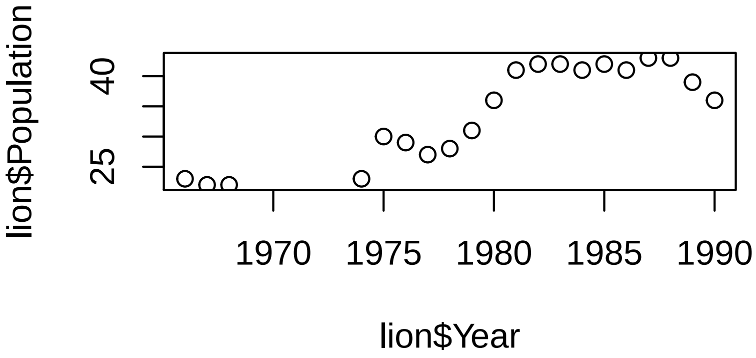
```
plot(chimp$Year, chimp$Population)
```

What does this plot tell us?

The population increases to its highest point at 1966 and then decreases to 1969. The population then increases again until 1972 and decreases in 1973.

We can also create a plot for the lion population numbers.

```
lion <- read.csv("lion.csv")
plot(lion$Year, lion$Population)
```



You will notice that there are no population sizes for the years 1969-1973 – these are the rows that we deleted earlier because the population sizes were unknown. However, you'll notice that the x-axis still goes from 1966-1990 but there are not values for those years that are missing. Now, let's us work on customizing our plot. There

are many more arguments that can be added to the plot command.

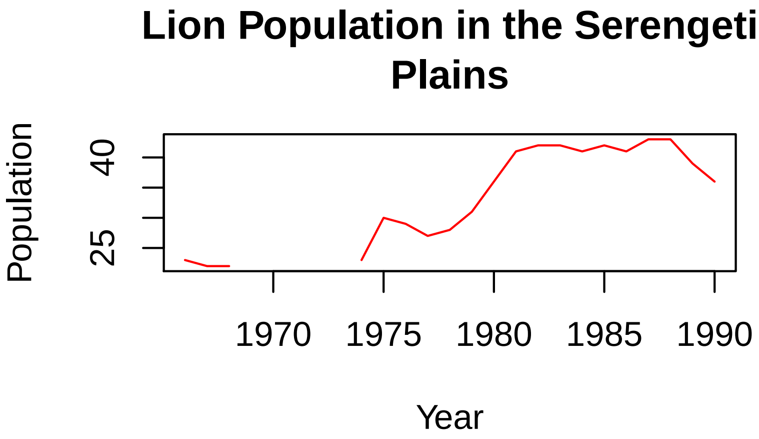
- Type - the type of plot to be drawn (Points(default) or Lines)
- Main - the title of the plot
- Xlab, Ylab - the axis labels
- Xlim, Ylim - the range of values on each axis
- Col - color of symbols

Let's add these arguments to our chimpanzee plot. Let's now make a line plot, instead of a point plot – add a title and axis labels – and change the color to blue.

```
plot(chimp$Year, chimp$Population, type = "l",
    main = "Chimpanzee Population in Gombe National Park",
    xlab = "Year", ylab = "Population", col = "blue")
```

Next, let's fix our lion plot. Let's change it to a line plot, add a title and axis labels, and change the color to red.

```
plot(lion$Year, lion$Population, type = "l", main = "Lion Population in the Serengeti
Plains",
    xlab = "Year", ylab = "Population", col = "red")
```



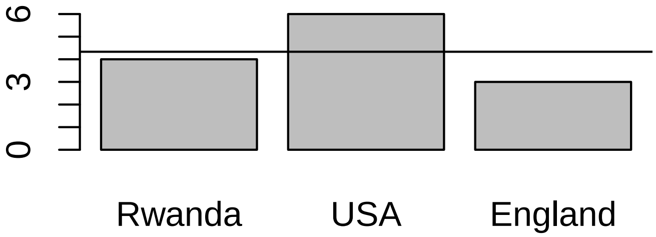
You will notice there is a break in the line for the years where there is missing data.

## *Other Common Plots*

Barplot
A barplot is a graphical way to display and compare the values of different categories within a dataset.

Create a barplot for the following vector and add labels for each bar.

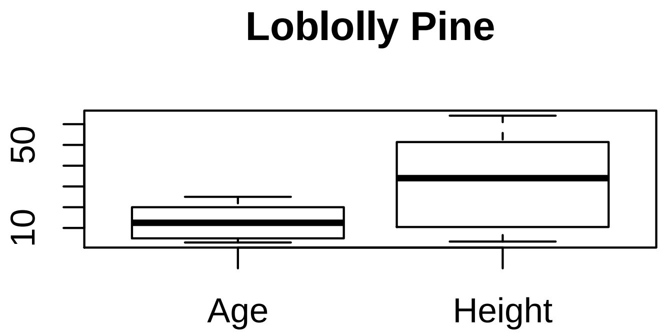
```
data <- c(4, 6, 3)
names(data) <- c("Rwanda", "USA", "England")
barplot(data)
abline(h = mean(data))  #adds a horizontal line to the graph at the mean value
```



Boxplot A boxplot shows the distribution of the data based on the five number summary: minimum, first quartile, median, third quartile, and maximum.

Create a side-by- side boxplot for the Loblolly dataset.

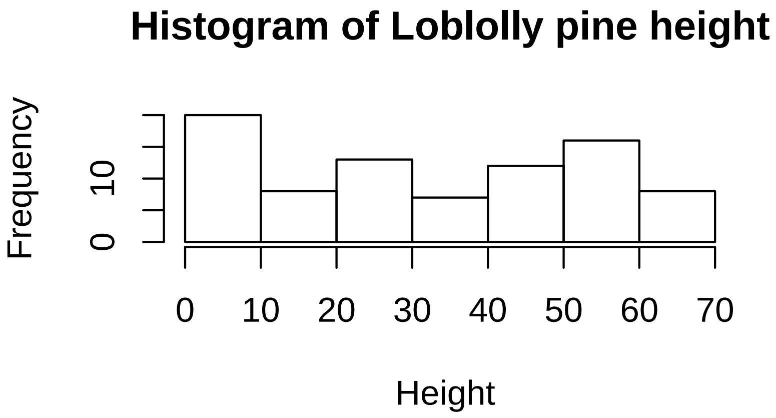
```
boxplot(Loblolly$age, Loblolly$height, main = "Loblolly Pine",
    names = c("Age", "Height"))
```



Histogram A histogram is a graphical way to represent the distribution of a dataset. The bars represent the frequency of each value within the dataset.

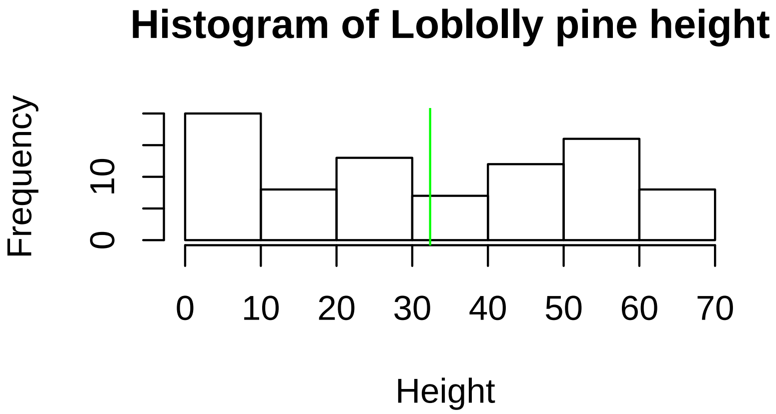
Create a histogram of the Loblolly pine height from the Loblolly dataset.

```
hist(Loblolly$height, xlab = "Height ", main = "Histogram of Loblolly pine height")
```



Add a green vertical line to show the mean tree height.

```
hist(Loblolly$height, xlab = "Height ", main = "Histogram of Loblolly pine height")
abline(v = mean(Loblolly$height), col = "green")
```



Multiple Plots

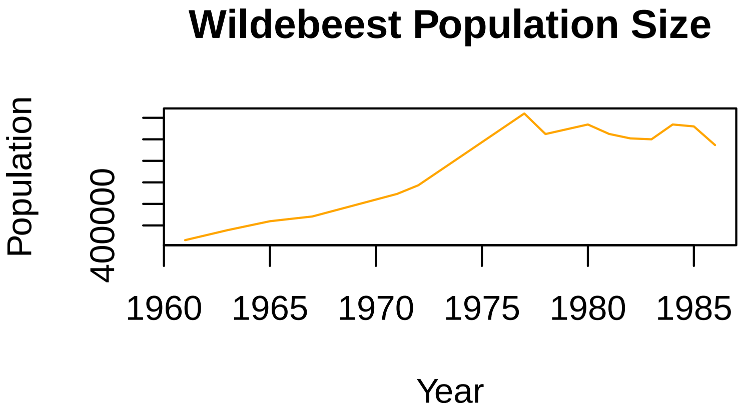
Let's do a quick analysis of two new datasets (Blue Wildebeest and African Buffalo) – and then learn how to create multiple plots on the same page, as well as one plot with multiple sets of data.

Import the Blue Wildebeest (wildebeest.csv) and the African buffalo (buffalo.csv) datasets. These datasets give population numbers of wildebeest and buffalo from the Serengeti Plains.

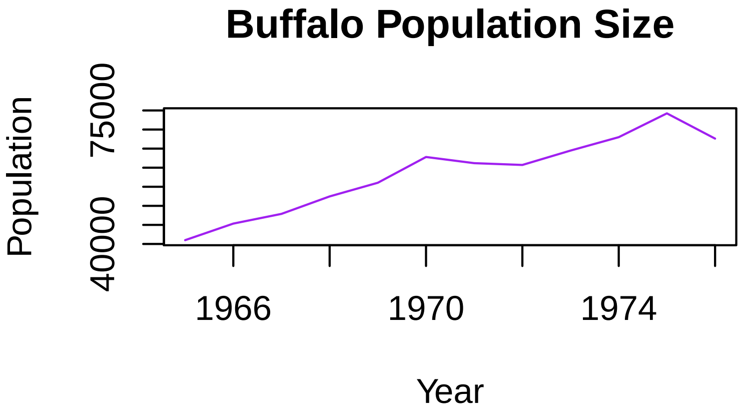
```
wildebeest <- read.csv("wildebeest.csv", header = T)
buffalo <- read.csv("buffalo.csv", header = T)
```

Create a line plot of the population sizes for each of these species – make sure to include a title, axis labels, and make each plot a different color.

```
plot(wildebeest$Year, wildebeest$Population, main = "Wildebeest Population Size",
    xlab = "Year", ylab = "Population", col = "orange",
    type = "l")
```

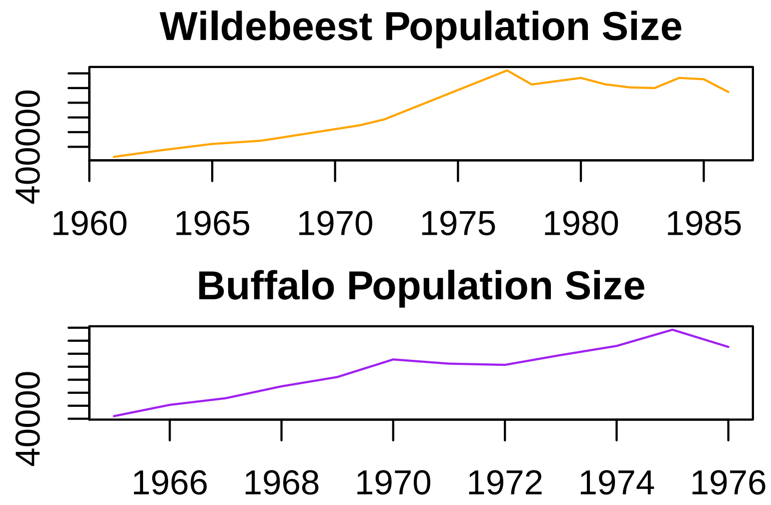


```
plot(buffalo$Year, buffalo$Population, main = "Buffalo Population Size",
    xlab = "Year", ylab = "Population", col = "purple",
    type = "l")
```



Now, let's keep these same plots – but what if we want them both on the same page, for example for in a publication?

```
par(mar = c(2, 2, 2, 2))
par(mfrow = c(2, 1))  #two rows and one column of plots on the page
plot(wildebeest$Year, wildebeest$Population, main = "Wildebeest Population Size",
    xlab = "Year", ylab = "Population", col = "orange",
    type = "l")
plot(buffalo$Year, buffalo$Population, main = "Buffalo Population Size",
    xlab = "Year", ylab = "Population", col = "purple",
    type = "l")
```



If we want to instead have two columns and one row of plots (i.e. two plots side by side) we would use *par(mfrow=c(1,2))*. For two columns and three rows (i.e. six plots total on page) we would use *par(mfrow=c(3,2))* and so forth.

Note: After changing the number of columns/rows on the page – it is necessary to change it back to 1 row and 1 column to have just one plot on the page again *(par(mfrow=c(1,1))*. Multiple Sets of Data on the Same Plot & Legends

Now, let's use the same data again for wildebeest and buffalo, but create just one plot showing both sets of data. We will also include a legend to show which set of data relates to which animal.

These two datasets cover different time periods – so before we create our plot let's subset the wildebeest dataset to cover the same years as the buffalo dataset.

```
wildebeest2 <- subset(wildebeest, wildebeest$Year >
    1964 & wildebeest$Year < 1977)
```

Now let's put the window back to one column and one row, and then create a plot of the wildebeest data in blue – the same way we created plots before. We will also add the *ylim* argument, which sets the limits of the y-axis. We will set the minimum value to the minimum value of the buffalo dataset and the maximum value to the maximum value of the wildebeest dataset – since these are the two extreme values.

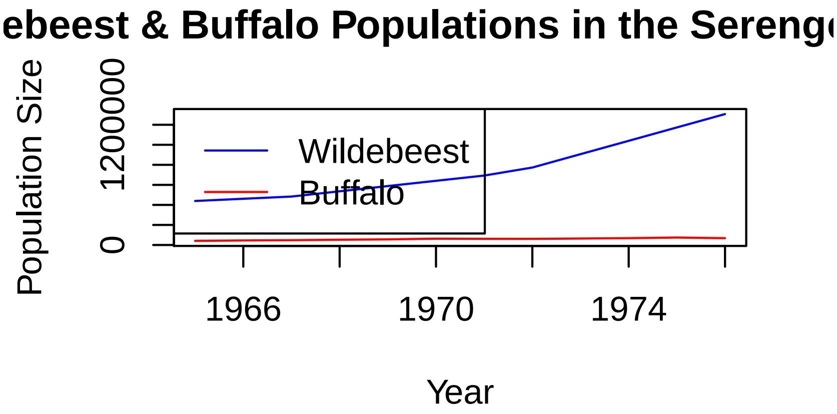
```
par(mfrow = c(1, 1))
plot(wildebeest2$Year, wildebeest2$Population,
    xlab = "Year", ylab = "Population Size", col = "blue",
    type = "l", main = "Wildebeest &amp; Buffalo Populations in the Serengeti Plains",
    ylim = c(41000, 1306603))
```

Now, let's add the buffalo data to the same plot using the *lines* command. If we wanted to add the data as points instead of the lines we would use the *points* command. We will plot the buffalo data in red.

```
lines(buffalo$Year, buffalo$Population, col = "red")
```

For the last step, let's add a legend to the plot so that we know which set of data represent which animal. We will put the legend in the top left corner of the plot. Then put it all together.

```
# par(mfrow=c(1,1))
plot(wildebeest2$Year, wildebeest2$Population,
    xlab = "Year", ylab = "Population Size", col = "blue",
    type = "l", main = "Wildebeest & Buffalo Populations in the Serengeti Plains",
    ylim = c(41000, 1306603))
lines(buffalo$Year, buffalo$Population, col = "red")
legend("topleft", col = c("blue", "red"), lty = 1,
    legend = c("Wildebeest", "Buffalo"))
```



In the above R code, you'll notice the lty argument. This stands for line type, with line type 1 as the solid line – try switching it to other numbers between 1 and 6 for other line types. If you had a plot using points instead of lines, you would use the *pch* command instead – which stands for point character. Point character 16 is a solid circle. Try using point characters 0-25 for different symbols!

## *Problems*

1. Import the Lion and Black Rhino files into R that we were using in the last assignment ('lionCrater.csv' and 'blackRhinoCrater.csv').

2. Create a plot of the lion population size with the year on the x-axis. Include a title, axis labels, and make the points blue. Can you figure out how to change the open circle points to another shape? Try and change them to closed circles. Is the population increasing or decreasing? What year is the minimum and maximum population sizes?

3. Create a plot of the black rhino population size with the year on the x-axis. This time make it a line plot. Include a title, axis labels, and the make the line dark green. Can you figure out how to change the line type to a dotted line? Is the population increasing or decreasing? What year is the minimum and maximum population sizes?

4. Put the two plots that you just made onto one page with one column and two rows.

5. Create two new data frames Lion2 and Rhino2 that contain the subset of years that overlap between the two datasets (like we did in the last assignment). These new datasets will be used for the next plot.

6. Create a line plot of the lion population sizes. On the same plot add the black rhino population sizes as points. Make the lion population line purple and the black rhino population points red. Includes a title and axis labels. You will need to set the limits of the y-axis from (0,100) otherwise the rhino points will not be visible when you add them to the plot.

7. Add a legend to the top right of the plot you just made – make sure next to lion you have a purple line, and next to black rhino you have a red point.

8. Create a barplot showing the minimum population size for black rhino and lion. Make sure to include a title and labels for each of the bars.

9. Create a side-by- side boxplot of the population size for lions and black rhinos – make sure to include a title, and a label below each box to show which animal it represents.

10. Create a histogram of the population size for black rhinos. Add an orange vertical line to show the mean population size.