Exercise 2: Subsetting, Plotting, and T-Test

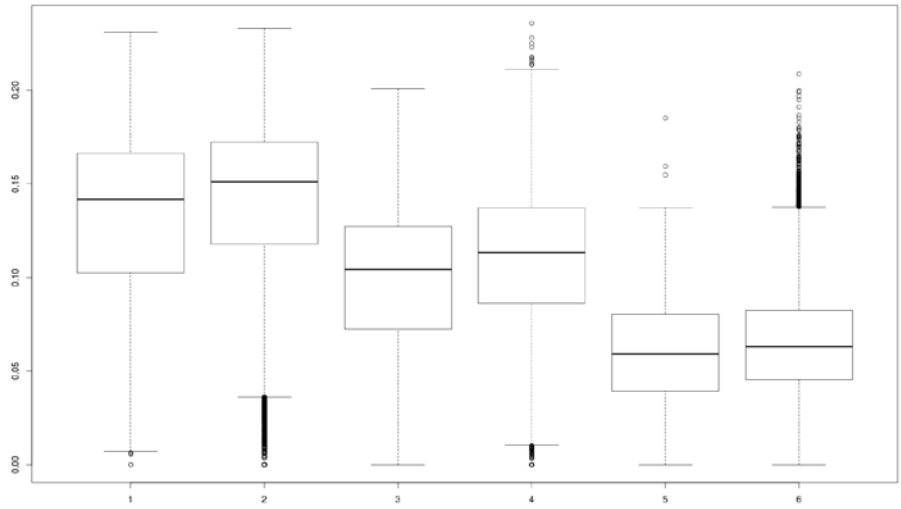
For this exercise we will use a data set that I generated consisting of genetic distance measures for 13,600 loci across 3 species. Our goal will be to see whether or not autosomal and sex-linked loci are diverging at different rates and to graph these two classes of coding sequences in a way that illustrates any differences between them.

Part A Reading and Subsetting

1. Download the file div.csv and save in r.seminar folder.
2. Open an R script and save it as div.script.R. In your working directory.
3. Read the data from the file into a data frame (e.g., call it "mydata") using the read.csv command. By default, all columns with character data will be converted to factors. A factor is like a character variable except that its unique values represent **levels**. Often we will use a factor to subset our data.
4. Use the "str" command to obtain a compact summary of the contents of the data frame. The TC number and the location should be listed as factors.
5. Let's clean this data up by extracting only the columns of interest (divergence measures and location). Assign these columns back to your original variable. You should now have a data frame with 13,600 rows and 4 columns (3 species and the location).
6. Now that we have our data in a good format lets go ahead and use the attach function: attach(mydata). This will make the mydata object the first item in the search path for R. This will allow us to simply use the name of a column rather than the object and column name.

Part B Basic Plotting

1. Start out with just trying some of the different basic plotting functions:
```
plot(T.madens ~ T.confusum)
hist(mydata[,1])
boxplot(mydata[,1],mydata[,2])
```
2. The boxplot looks like it might work well so let's focus on using that function for today. Your goal is to have 6 bars (1 autosome and 1 sex chromosome for each species). To do this we will have to subset the data by column and the value in the location column. For example to pull out the Tribolium confusum sex chromosome data we would type:
```
mydata[Location=="Sex Chromosome",1]
```
3. Try putting this as the first argument in the boxplot function. If this works then begin adding the other subsets in the same way separating each one with a comma. Once you have entered all six you should get a graph like this:



Part C T-test

1. Lets go ahead and do a simple T-test on the data for each species to see whether the values for sex chromosomes and autosomes really are different. To do this we will use the t.test function. The default settings for this function are unpaired and unequal variance - this is appropriate for our data and is actually a Welch's T-test.
   ```
   conf <- t.test(mydata[,1] ~ Location)
   ```
2. If you want to understand more about this function remember to try
   ```
   help(t.test)
   ```
3. to see the result extract the p.value from conf
   ```
   conf$p.value
   ```
4. Perform the same test for column 2 and 3 and store these in variables named free and made.

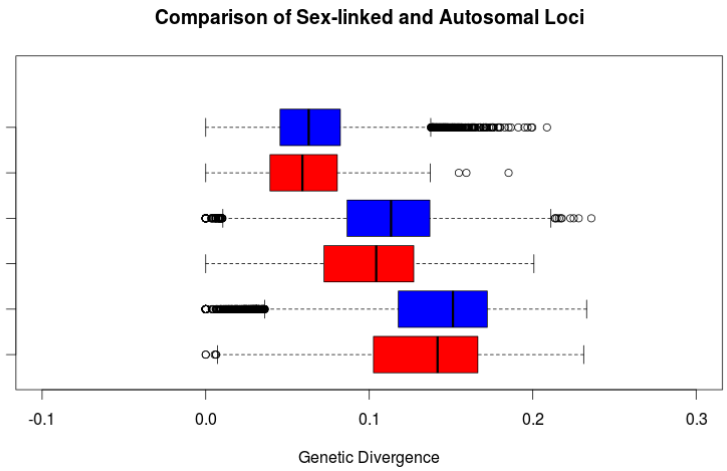
Part D Final graph

Now we can combine our graph and our statistical test together in one plot. First copy and paste your graph code from section B. Now simply begin adding the different arguments that you need to produce the display you want. You can be creative in this part here is the way that I did it:

I added these arguments to my boxplot function:

```
horizontal=T,
xlab="Genetic Divergence",
col=c("red","blue"),
main="Comparison of Sex-linked and Autosomal Loci",
ylim=c(-.1,.3),
xlim=c(.5,7.3),
```

The xlim and ylim allowed me to create space around my plot so that I have room to add species names and p-values.



Once I had this plot I began using the text command to add text that I needed for instance the p-value for confusum was added with this line of code:

```
text(.28,1.5,signif(conf$p.value, digits=3),cex=.8)
```

X position

Y position

Text to plot

Size of text

There is a legend command that you can use but in this graph I decided to make my own legend using the text command above and the point command to add colored boxes above the graph:

```
points(0.01,7,pch=15,col="blue",cex=3)
```

My final graph looked like this: