Test for differences

Aims of this practical

1. Write clear hypotheses

2. Explore descriptive statistics

3. Create appropriate plots to visualise hypotheses

4. Examine assumptions of tests

- Normality of variance between groups (Bartlett test)
- Normally distributed data (Shapiro test)

5. Carry out parametric and non-parametric tests for differences

- One sample t-test
- Two sample t-test
- Mann-Whitney U (aka Wilcoxon) test (non-parametric)
- Paired t-test
- One-way ANOVA (analysis of variance)
- Kruskal-Wallis test (non-parametric)

6. Report your results

Coding you will learn:

Tests conducted

Name	**Data type**	**Test type**
One-sample t-test	Continuous - difference between pop. & mean	Parametric
Two-sample t-test	Continuous - 2 categories only	Parametric
Mann-Whitney U	Continuous - 2 categories only	Non-parametric
Paired t-test	Continuous - 2 categories are paired	Parametric
One-way ANOVA	Continuous - More than 2 categories	Parametric
Kruskal-Wallis	Continuous - More than 2 categories	Non-parametric

Other useful code for…

Creating PDFs
Plotting (boxplot, bar graph, mean plots, histograms)
Data summaries
Transformations
Restructuring data

One sample t-test

To begin with we will explore a data set of daily energy intake of Impala from the Serengeti (units are Kilojoules). Lets first enter the data manually in R, remembering to set up your script as you did in Practical 1 with some descriptive annotation. **Note**: if you're using a university PC you will need to create a folder in the C Drive called **Rpackages** and run the following code **.libPaths("C:/Rpackages")** so you can install packages for this practical.

```
# DD/MM/2019 Your Name Here
# Data: various
# Analysis: Tests for differences: t-test, anova, mann whitney U, kruskall wallis

# clears r console
rm(list=ls())
# set working directory
setwd("H:/Documents/SaRD/Week_2")
# check working directory
getwd()
# list of files in working directory
list.files()
# .libPaths("C:/Rpackages")  # only run if using uni pc (see practical 1 for help)

# Enter data manually
daily.intake <- c(5260, 5470, 5640, 6180, 6390, 6515, 6805, 7515, 7515, 8230, 8770)
```

Now, suppose we wish to investigate whether energy intake in the wild deviates significantly from recommended energy intake in zoos, 7725 Kj/day. Assuming the data come from a normal distribution, the object is to test whether the distribution could have a mean of 7725. We can use a one-sample **t.test()** to ask this.

Before exploring the data lets clearly state our null hypothesis; **there is no significant difference between wild Impala energy intake and the recommended energy intake in captivity.**

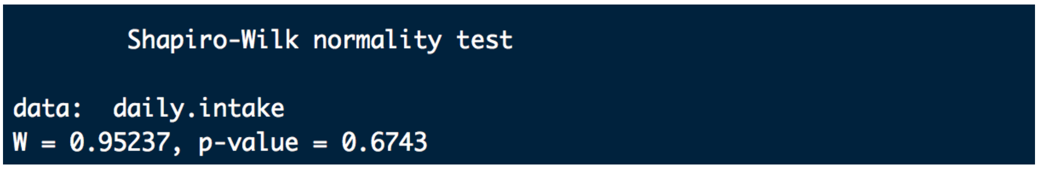
Now lets explore the data with some descriptive statistics and plots. Again, include some annotation explaining what's going on in our script.

```
# Data: Impala energy intake
# Analysis: one-sample t-test
# Null Hypothesis: there is no significant difference between wild Impala energy intake and the #
# recommended energy intake in captivity.

boxplot(daily.intake) # roughly what is the mean?
hist(daily.intake) # plot a histogram, does the data look vaguely normally distributed?

mean(daily.intake) # write down the mean?
sd(daily.intake) # write down the standard deviation?
```

Before we start testing the hypothesis we need to know the data meet the test assumption of normal distribution?

```
# Test for normality
shapiro.test(daily.intake) # Is this data significantly different from normal?
```

###Output from R



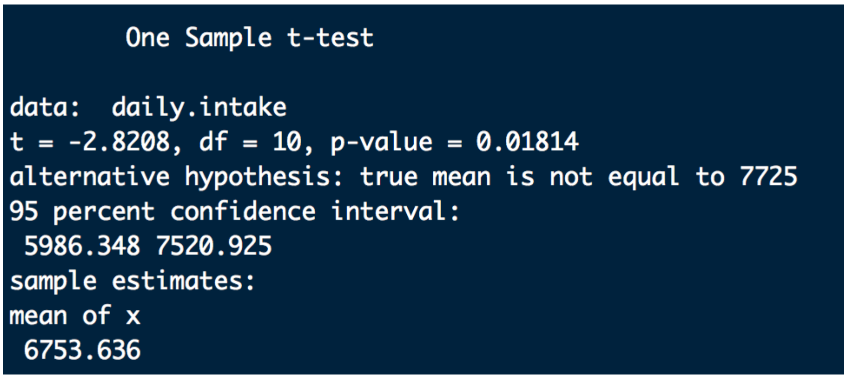
```
        Shapiro-Wilk normality test

data:  daily.intake
W = 0.95237, p-value = 0.6743
```

The p-value for our Shapiro test comes back much greater than 0.05 so we can be confident that our data is NOT significantly different from normal. This is important so I'll repeat it – if you p-value is greater than 0.05 the data are not significantly different from a normal distribution.

So our data is normal, we can proceed with a one-sample t-test.

```
# One sample t-test
t.test(daily.intake, mu=7725)
```

###Output from R



```
        One Sample t-test

data:  daily.intake
t = -2.8208, df = 10, p-value = 0.01814
alternative hypothesis: true mean is not equal to 7725
95 percent confidence interval:
 5986.348 7520.925
sample estimates:
mean of x
 6753.636
```

The test output gives you the value of the t-statistic, the degrees of freedom and finally the significance value (p-value), which tells you there is a significant difference between the wild Impala data and the recommended zoo energy intake.

Now report your results in a clear sentence (see example statement below) and plot (boxplot) the data ready for your publication in Nature!

*RESULTS STATEMENT: **The energy intake of wild Impala (mean±sd, 6754±1142) was significant difference to the zoo recommended energy intake (7725 Kj/day) (t=-2.821, df=10, p=0.018).***

```
pdf("Impala.intake.pdf", width = 8, height = 6) # create pdf saved in working directory
boxplot(daily.intake, ylab="Wild energy intake (Kj/day)") # boxplot with y-axis label
abline(h=7725, lty=3, lwd=2, col="grey") # horizontal line in plot at y=7725
legend('topleft', c("Zoo intake"), lty=3, lwd=2, col="grey", bty='n', cex=1) # legend
dev.off()  # close plotting element
```

Two-sample t-test

We are now going to use the *mtcars* data set that is built into R.

We are interested in considering if the fuel economy of cars (*mpg*; continuous variable) is different between manual and automatic transmission (*am*; categorical variable, with two groups (0=automatic, 1=manual).

State clearly your hypothesis before continuing (add this to your script).

```
# Data: mtcars dataset
# Analysis: Two-sample t-test
# Null Hypothesis: ……..

data(mtcars)
mtcars
names(mtcars)
str(mtcars) # what is this function doing?
hist(mtcars$mpg) # histogram of fuel economy

# Histogram in ggplot
library(ggplot2) # open library for ggplot2 package
ggplot(mtcars, aes(x=mpg)) +
  geom_histogram(binwidth=2, colour="black", fill="white") +
  geom_vline(aes(xintercept=mean(mpg)), color="red", linetype="dashed", size=1)
```

From the histogram does the data look normally distributed?

Now plot the data (boxplot) and see if the two groups look different? What are the means and standard deviation of the two groups?

```
# Boxplot for automatic and manual cars
boxplot(mtcars$mpg~mtcars$am)

# We can use the tapply function to calculate the mean and the sd of groups:
tapply(mtcars$mpg, mtcars$am, FUN=mean)
tapply(mtcars$mpg, mtcars$am, FUN=sd)
```

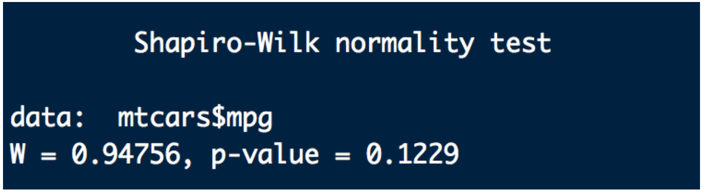
Before we can carry out our two-sample t-test, we must be satisfied that the data meet the assumptions of the t-test;
1) The data are continuous
2) The data are approximately normally distributed
3) The variances of the two sets of data are homogeneous (the same)
We already know that our *mpg* variable is a continuous dataset. Looking at the histogram

and boxplot we have created for this data would you say that the dataset meets the other two assumptions (discuss with a neighbour)?

Lets check these assumptions formally.

```
# Test for normality
shapiro.test(mtcars$mpg)

# Test for homogeneity of variances
bartlett.test(mtcars$mpg, mtcars$am)
```

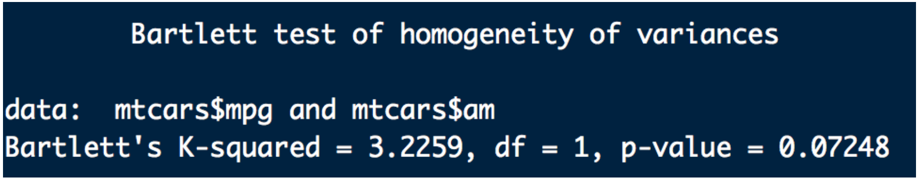
###Output from R



```
        Shapiro-Wilk normality test

data:  mtcars$mpg
W = 0.94756, p-value = 0.1229
```

P-value is greater than 0.05 so the mpg variable is not significantly different from normal

###Output from R



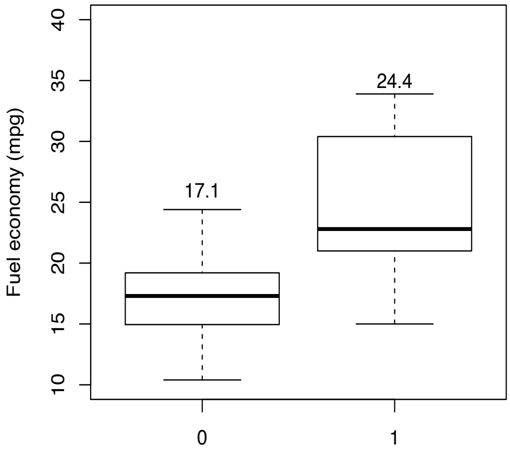
```
        Bartlett test of homogeneity of variances

data:  mtcars$mpg and mtcars$am
Bartlett's K-squared = 3.2259, df = 1, p-value = 0.07248
```

P>0.05 so the variances of the two groups are not significantly different?

Great! We can proceed with our t-test.

```
# Two sample t-test
t.test(mtcars$mpg~mtcars$am)
```

Now report your results clearly in a sentence and create the plot on the right to show your results, remember your plot will need appropriate axes labels, you might also want to add the means of each group to your plot (hint: **text()** function). Search ?boxplot and ?plot.



Paired t-test

For this test we will type in some blood pressure data taken from 15 athletes. Blood pressure measurements were taken from athletes before and after training so the data are paired per athlete.

```
# Data: Blood pressure (n=15 athletes)
# Analysis: Paired t-test
# Null Hypothesis: there is no improvement in blood pressure after exercise.

# Create three objects in R (‘subjects’, 'before' and 'after') with the following data
subjects<-c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15) # OR subjects<-seq(from=1, to=15, by=1)
before<-c(12.9, 13.5, 12.8, 15.6, 17.2, 19.2, 12.6, 15.3, 14.4, 11.3, 14.2, 13.7, 16.8, 12.9, 14.5)
after<-c(12.0, 12.2, 11.2, 13.0, 15.0, 15.8, 12.2, 13.4, 12.9, 11.0, 13.2, 13.4, 13.4, 11.2, 13.5)
```

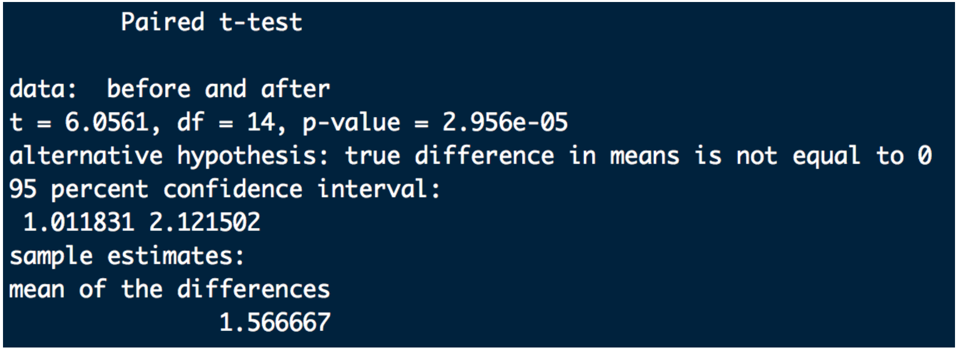
Go ahead and check the mean and standard deviation of the two groups.
It looks like the ‘after’ is lower than the ‘before’ measurements, lets plot the two data groups. Because our data structure is different (two separate columns) the easiest way to visualise the data is by plotting two separate boxplots next to each other using the par function **par(mfrow=c())**

```
# plot two boxplots next to each other
par(mfrow=c(1,2)) # plots are arranged in 1 row and 2 columns
boxplot(before, ylim=c(10,20), main="Before")
boxplot(after, ylim=c(10,20), main="After")

# plot two histograms stacked
par(mfrow=c(2,1))
hist(before)
hist(after)
```

Paired t-tests have the same assumptions as standard t-test and from the boxplot and histogram - the data look roughly ok to proceed.

```
# Paired t-test
t.test(before,after, paired=TRUE)
```

### Output from R



```
        Paired t-test

data:  before and after
t = 6.0561, df = 14, p-value = 2.956e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 1.011831 2.121502
sample estimates:
mean of the differences
               1.566667
```

Now we need to write a sentence that properly reports the results of this test using the output.

RESULTS STATEMENT: ***There is a significant difference in blood pressure before (mean±SD, 14.5±2.1) and after (12.9±1.3) exercise (t= 6.0561, df=14, p<0.001).***
NOTE: standard format for reporting very small p-value, i.e. those below 0.001

We now need to do some data restructuring so we can plot our results on one graph. See the Restructuring Data page on the R Cookbook website for more information. Essentially, we want to combine the data we created in one dataframe and then restructure that from the 'wide' form into the 'long' form. Wide form is often used with programs such as SPSS but R is clever and can create, change and handle all forms!

Make sure you understand what the restructuring is doing to your data, look at both dataframes (blood and blood.long) to check.

```
# Combine the data using the data.frame function
blood<-data.frame(subjects, before, after)
blood # view data in the console
str(blood) # check data structure

# Open the reshape2 package
library(reshape2)
# Use the melt function to change the blood dataframe from wide to long form.
blood.long<-melt(blood, id.vars=c("subjects"), measure.vars=c("before", "after"),
          variable.name="Condition", value.name="BP") # also change the column names
str(blood.long) # see how the data structure has changed
blood.long # view data in the console

par(mfrow=c(1,1)) # change plotting window back to standard single graph format
boxplot(BP~Condition, data=blood.long)
```

Mann-Whitney U test (aka Wilcoxon) – non-parametric

Lets continue looking at two groups but now we will look at some data that does not meet the assumptions of the t-test.

You may be familiar with the *iris* dataset (morphological variables of three plant species) built into R (it was used in the R Basics Bonus Lab). First load the data and then using the **subset** function we are going to separate the two species we are interested in, setosa and versicolor. We are going to test the difference in petal width between setosa and veriscolor. Make sure you write a clear hypothesis for your test.

```
# Data: iris dataset
# Analysis: Mann-Whitney U non-parametric test (aka Wilcoxon)
# Null Hypothesis: …………….

data(iris)
iris
names(iris) # examine variable names
str(iris) # examine the data structure

# subset only those two species we are interested in
notvirginica<-subset(iris,Species!="virginica") # removes virginica species
```

Lets explore the data as before with a histogram, boxplot, means and standard deviation. This time load the ggplot2 library and try to create a boxplot using this package (for help Goggle search ‘R ggplot2 boxplot’). If you are struggling try copying and pasting the example from your web search using the *mtcars* dataset on the following web page http://docs.ggplot2.org/0.9.3.1/geom_boxplot.html and then substituting the *mtcars* parts of the code for the *notvirginica* parts.

Do you think there will be a difference between species?

Do you think the data will conform to the t-test assumptions of normality and homogeneity of variances?

Now test the assumptions using the Shapiro test and the Bartlett test. What are the p-values and results of each? Remember that 1.634e-09 is just the scientific way of writing 0.000000001634. Hence a very small p-value!

Lets try to transform the data and see if that helps with our normality issues.

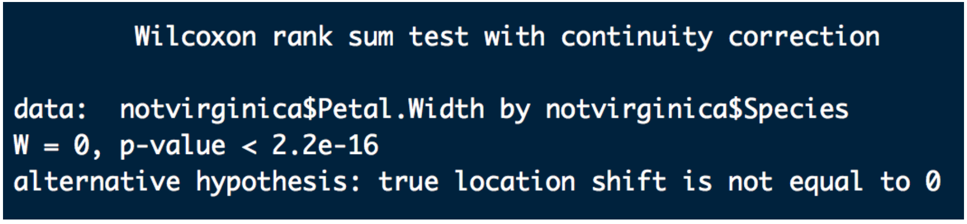
```
# Log10 transform petal width - creating a new object called PW.log
PW.log<-log10(notvirginica$Petal.Width)
# Square root transform petal width - creating a new object called PW.sqrt
PW.sqrt<-sqrt(notvirginica$Petal.Width)
# Use cbind (column bind) to add the new objects to our dataframe
new.notvirginica<-cbind(notvirginica, PW.log, PW.sqrt)

# Test transformed data for normality
hist(new.notvirginica$PW.log)
shapiro.test(new.notvirginica$PW.log) # still very significantly different from normal

hist(new.notvirginica$PW.sqrt)
shapiro.test(new.notvirginica$PW.sqrt) # still very significantly different from normal
```

Ok, so our data are not normally distributed and the variance between groups are not equal. We now need to turn to a non-parametric Mann-Whitney U test, which does not make assumptions about normality and homogeneity of variances. It is a rank test, which means it converts the raw data into ranks before testing it. These types of tests are generally less powerful than their parametric equivalents but are excellent when your data fail the assumptions of t-tests and ANOVAs (note: a one-way ANOVA carried out on two groups is equivalent to a two-sample t-test).

Lets try the Mann-Whitney on our original (non-transformed) data.

```
# Go back to our original data
# Carry out non-parametric Mann-Whitney test (Wilcoxon)
wilcox.test(notvirginica$Petal.Width~notvirginica$Species)
```

###Output from R

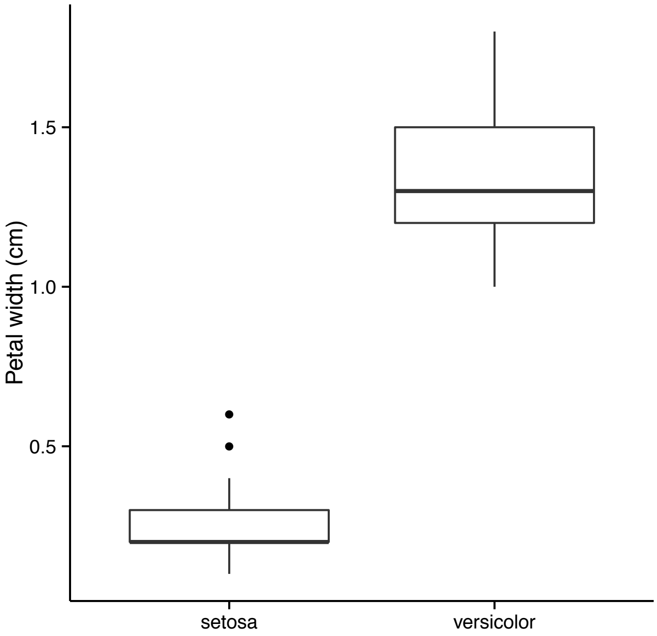


```
	Wilcoxon rank sum test with continuity correction

data:  notvirginica$Petal.Width by notvirginica$Species
W = 0, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
```

The p-value is much smaller than 0.05 so you should reject your null hypothesis, the two groups are significantly different.

Now write this up in a clear statement giving means, standard deviation, test statistic and p-value.

For extra fun try to reproduce the boxplot below using ggplot2! The following Google searches will help, ‘R ggplot2 axis labels cookbook’ and ‘R ggplot2 clear background’. If you are running short of time a standard boxplot is fine.



One-way ANOVA (analysis of variance)

Now we are going to look at testing more than two groups. Again we will use the *iris* dataset. This time we will test for differences in sepal width between all three species. Go ahead and explore the data with means, sd, boxplot and histogram.

```
# Data: iris dataset
# One-way ANOVA
# Null Hypothesis: There is no significant difference in sepal width between three species of iris

data(iris)
iris
names(iris)
str(iris)

# We can use the tapply function to calculate the mean and the standard deviation of groups:
tapply(iris$Sepal.Width, iris$Species, FUN=mean)
tapply(iris$Sepal.Width, iris$Species, FUN=sd)

hist(iris$Sepal.Width)

boxplot(iris$Sepal.Width~iris$Species)
```

Now we need to examine the test assumptions for the ANOVA, which are the same we looked at earlier for t-tests

1. The data are continuous
2. At least approximately normally distributed
3. The variances of the groups are homogenous

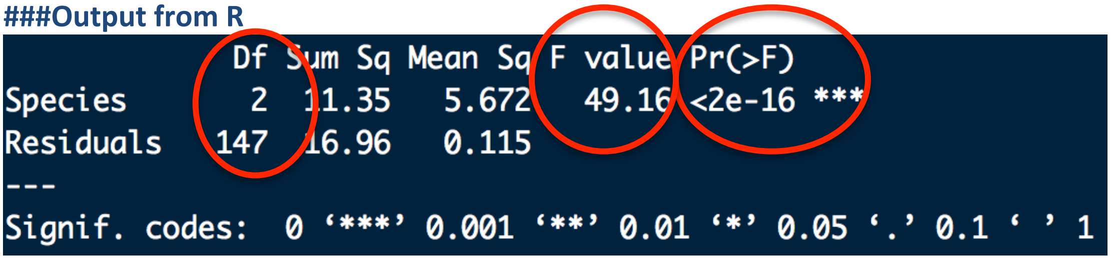
```
# Test for normality
shapiro.test(iris$Sepal.Width) # NOT significantly different from normal

# Test for homogeneity of variances
bartlett.test(iris$Sepal.Width, iris$Species) # NO significant difference in variance
```

Ok, so our data meet the ANOVA assumptions, we can proceed with the test.

```
# ANOVA to test for differences in the three groups
anova1<-aov(Sepal.Width~Species, data=iris)
summary(anova1) # use the summary function to extract your ANOVA results
```

We will use the **summary** function for many of the tests in future practicals, try running ?summary for more information on this function.

###Output from R



```
           Df Sum Sq Mean Sq F value Pr(>F)    
Species     2  11.35   5.672   49.16 <2e-16 ***
Residuals 147  16.96   0.115                   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The summary function gives us the critical F-value, p-value and the degrees of freedom we need to report in our results statement.

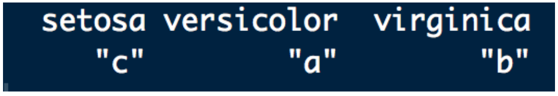
Note:

- R gives the p value as <2e-16 or 0.0000000000000002, remember from above anything less than 0.001 is written as p<0.001.
- When reporting the results of statistical tests, we should always report either the n number or the degrees of freedom (df)
- We could also report the mean ± SD sepal widths for each species in the statement – or put a graph or table with this data in our report.

Our ANOVA results tell us there is a difference but it does not tell us which groups are different from each other, for this we will need to do a **post hoc test**. There are many possible post hoc tests but for simplicity we will use the commonly applied Tukey post hoc test implemented using the **multcomp** package.

```
# Tukey post hoc test
library(multcomp) #

# carry out post hoc test on our anova1 object – Species are our iris categories
cld(glht(anova1, linfct=mcp(Species="Tukey")))
```

##Output from R



```
    setosa versicolor  virginica 
       "c"        "a"        "b" 
```

If groups have the SAME letter they are NOT significantly different. If they have different letters they are significantly different. So we would write this formally as: ***Sepal width was significantly different between all species (F=49.16; df= 2,147; p<0.001).***

To report the results in a plot we could use a boxplot, or we could use a mean plot with error bars, or a bar graph with error bars. Using the following function obtained from the R Cookbook website we can very quickly produce either of these error bar plots.
NOTE: do not write out the code below, copy and paste it from R Cookbook page http://www.cookbook-r.com/Manipulating_data/Summarizing_data/ its under the ‘Using ddply’ heading.

```
## Summarizes data.
##   Gives count, mean, standard deviation, standard error of the mean, and confidence
## interval (default 95%).
##   data: a data frame.
##   measurevar: the name of a column that contains the variable to be summariezed
##   groupvars: a vector containing names of columns that contain grouping variables
##   na.rm: a boolean that indicates whether to ignore NA's
##   conf.interval: the percent range of the confidence interval (default is 95%)
summarySE <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
                      conf.interval=.95, .drop=TRUE) {
 library(plyr)

 # New version of length which can handle NA's: if na.rm==T, don't count them
 length2 <- function (x, na.rm=FALSE) {
  if (na.rm) sum(!is.na(x))
  else       length(x)
 }

 # This does the summary. For each group's data frame, return a vector with
 # N, mean, and sd
 datac <- ddply(data, groupvars, .drop=.drop,
        .fun = function(xx, col) {
         c(N    = length2(xx[[col]], na.rm=na.rm),
          mean = mean   (xx[[col]], na.rm=na.rm),
          sd   = sd     (xx[[col]], na.rm=na.rm)
         )
        },
        measurevar
 )

 # Rename the "mean" column
 datac <- rename(datac, c("mean" = measurevar))

 datac$se <- datac$sd / sqrt(datac$N)  # Calculate standard error of the mean

 # Confidence interval multiplier for standard error
 # Calculate t-statistic for confidence interval:
 # e.g., if conf.interval is .95, use .975 (above/below), and use df=N-1
 ciMult <- qt(conf.interval/2 + .5, datac$N-1)
 datac$ci <- datac$se * ciMult

 return(datac)
}
```

This is the beauty of freely available software with a large user group, people write code and functions for you and you can use them as much as you like, completely freely, for free!

Now we can run our new function to summarise our sepal width data.

```
# Summarise data
SW.sum <- summarySE(iris, measurevar="Sepal.Width", groupvars=c("Species"))
SW.sum
```

If you have run the SW.sum code you will be presented with a neat little table of means, standard deviations, standard errors and confidence intervals. Now the really clever part is this SW.sum object is directly compatible with ggplot2 graphics!

```
# Mean CI plot
ggplot(SW.sum, aes(x=Species, y=Sepal.Width)) +
  geom_errorbar(aes(ymin=Sepal.Width-ci, ymax=Sepal.Width+ci), width=.1) +
  geom_point() +
  theme_bw()

# Bar graph with CI
ggplot(SW.sum, aes(x=Species, y=Sepal.Width)) +
  geom_bar(position=position_dodge(), stat="identity") +
  geom_errorbar(aes(ymin=Sepal.Width-ci, ymax=Sepal.Width+ci), width=.1) +
  theme_bw()
```

Now all you need to do is tidy up our axis labels, add our post hoc results and create a PDF of our plot.

```
# Mean CI plot
pdf("Sepal_Width_Tukey.pdf", width = 5, height = 5)
ggplot(SW.sum, aes(x=Species, y=Sepal.Width)) +
  geom_errorbar(aes(ymin=Sepal.Width-ci, ymax=Sepal.Width+ci), width=.1) +
  geom_point() +
  scale_y_continuous(name="Sepal Width (mm)") +
  annotate("text", x = 1, y = 3.6, label = "c") +
  annotate("text", x = 2, y = 2.9, label = "a") +
  annotate("text", x = 3, y = 3.1, label = "b") +
  theme_bw()
dev.off()
```

Kruskal-Wallis test (non-parametric)

What if I have more than two groups but the data fail the ANOVA test assumptions!! Don't panic, there is a non-parametric equivalent, the Kruskal Wallis test. Lets look at petal width from the iris data again. First explore the data and plot.

```
# Data: iris dataset – 3 groups
# Test: kruskall-wallis
# Null Hypothesis:
data(iris)
iris
names(iris)
str(iris)

# We can use the tapply function to calculate the mean and the sd groups:
tapply(iris$Petal.Width, iris$Species, FUN=mean)
tapply(iris$Petal.Width, iris$Species, FUN=sd)

hist(iris$Petal.Width)

boxplot(iris$Petal.Width~iris$Species)
```

Does the data look normally distributed? Carry out a Shapiro test for normality and a Bartlett test for homogeneity.

Write down your results for each test.

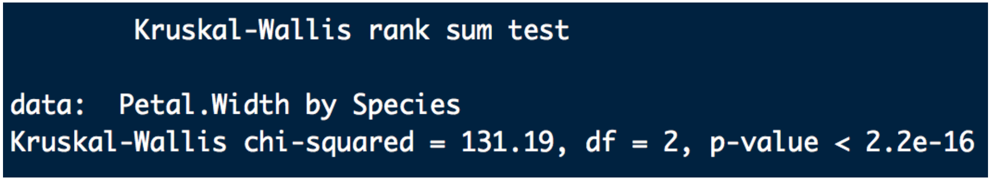
Now we know we have non-normal data, we should try to transform (log10, sqrt) the data and retest for normality.

```
# Log10 transform petal width - creating a new object called PW.log
PW.log<-log10(iris$Petal.Width)
# Square root transform petal width - creating a new object called PW.sqrt
PW.sqrt<-sqrt(iris$Petal.Width)
# Use cbind (column bind) to add the new objects to our dataframe
new.iris<-cbind(iris, PW.log, PW.sqrt)

# Test for normality
hist(new.iris$PW.log)
shapiro.test(new.iris$PW.log) # still very significantly different from normal

hist(new.iris$PW.sqrt)
shapiro.test(new.iris$PW.sqrt) # still very significantly different from normal
```

Still non-normal after transformation so we can now proceed with a non-parametric test for differences.

```
# Kruskal-Wallis test
kruskal.test(Petal.Width~Species, data = iris)
```

###Output from R



```
        Kruskal-Wallis rank sum test

data:  Petal.Width by Species
Kruskal-Wallis chi-squared = 131.19, df = 2, p-value < 2.2e-16
```

Like an ANOVA, this does not tell us which groups (in our case, Species) are significantly different from each other. To do this, we need to do pairwise Mann-Whitney U (Wilcoxon) tests. We will need to subset each pair of species up before applying multiple Mann-Whitney tests.

```
# Subset Species into pairs - then conduct separate Mann-Whitney tests
notvirginica<- subset(iris,Species!="virginica")
notsetosa<-subset(iris,Species!="setosa")
notversicolor<-subset(iris,Species!="versicolor")

# Conduct separate Mann-Whitney tests on pairs of Species
wilcox.test(notvirginica$Petal.Width~notvirginica$Species)
wilcox.test(notsetosa$Petal.Width~notsetosa$Species)
wilcox.test(notversicolor$Petal.Width~notversicolor$Species)
```

RESULTS STATEMENT: ***Petal width was significantly different between all species (Chi=131.19; df= 2; p<0.001).***

We could have also done a pairwise Wilcoxon test in one step using the following code. This gives the same result but we don’t have the fun of subsetting!

```
# Pairwise wilcoxon test
pairwise.wilcox.test(iris$Petal.Width,iris$Species)
```

Now plot your results in either a boxplot, bar plot or mean plot with error bars.

THE END – please make sure you have a clean **working** and **annotated** script before next weeks practical class. Make sure you understand what you have done in the workbook and why you have done each test.