R: Getting comfortable with your data - Descriptive statistics, Normality, and Plotting

Once we have our dataset cleaned and tidied, it is time to start getting familiar with it. Running descriptive statistics and plotting the data will give you a great sense as to what your data looks like and what type of analysis you can conduct on it. Let's start by ensuring that we are all working with the same dataset - the woodchips.xlsx file.

Please also download the R script used in the Getting comfortable with your data.

Descriptive Statistics

Summary

R has a versatile function called Summary() that provides a summary of your analysis results. A function that we will use on a regular basis as we move to working more with our data and its analysis. For now we have a dataset called woodchips - let's try out the summary function on a dataset without any analysis:

summary(woodchips)

The results provide the mean, median, min, max, and quartiles for numeric values and data types for character or string variables. A great start to working with your data are these values.

Normality test - Shapiro-Wilk

When running ANOVAs, which we will tackle later on, one of the assumptions is that our residuals have a normal distribution. We may also be interested to test the distribution of our outcome variable. A common test for this is the Shapiro-Wilk test. To run this test on our woodwt variable, we use the following:

shapiro.test(woodchips$woodwt)

Notice that by running this small piece of code we ONLY get our Shapiro-Wilk statistic and associated p-value. Unlike other statistical packages, we only have the 1 test result and not 3-4 other tests.

Frequency and Cross-tabulations

All data that we collect may not be continuous, we may have categorical data, such as Quality score and SampleID in our Woodchips dataset. Calculating a mean or testing for a normal distribution doesn't make any sense for these 2 variables. We may be interested in their frequencies - for instance how many observations have a quality score of 2 or 5. To calculate a simple frequency we use the table() function in R.

table(woodchips$quality.factor)

Let's try a crosstabulation between the SampleID and Quality score:

table(woodchips$sampleID, woodchips$quality.factor)

Remember we can always save our results as an object, simply be naming it - for example let's save this frequency table in an object called mytable.

mytable <- table(woodchips$sampleID, woodchips$quality.factor)

To view its contents we simply type the name of the object and run it.

mytable

Two more options that accompany the table() include margin.table() and prop.table(). If you want row or column tables for your crosstabulations, you can run the margin.table() function. To obtain row and column percents you would run the prop.table() function. Try it out and see what you get as results.

Plotting your data

Along with running descriptive statistics we may also want to plot our data to get a visual representation of our data. There are a number of functions available to you for plotting. Let's start by using some of the base plotting functions and move onto playing around with one of the more popular plotting packages called **ggplot2**.

Base R plotting functions are:

- plot() for scatterplots and x-y plots
- boxplot() for a box plot
- barplot() for a barchart

Review the attached R script file for the examples and explanation of the code used.

Adding labels to the plot()

The syntax for the plot() is:

plot(woodchips$woodwt, main="Weight of woodchips", xlab="Indiviudal Samples", ylab="Weight in grams")

plot() calls on the function and will produce a plot that is available in the Plots window in RStudio

woddchips$woodwt - the the outcome variable that we want to plot

main="..." creates the Main Plot title. The title must be contained in between the " "

xlab="..." creates the X-axis label

ylab="..." creates the Y-axis label

Aside: Working with dataset names

You may have noticed that whenever recalling a variable name, you must include the name of the dataset with the variable and separated by a $. There are different ways of using the dataset name. Here are 2 lines of syntax that perform the same task but are recalling the name of the dataset in a different way:

with(woodchips, plot(quality, woodwt, pch=as.integer(quality)))
plot(woodchips$quality, woodchips$woodwt, pch=as.integer(woodchips$quality))

We can use these two methods with any R syntax - the way you choose to work will be personal preference.

ggplot2 package

The ggplot2 package is an extremely versatile package used to create plots in R. During the Winter 2018 semester we had the pleasure of having Andrew Frewi, Ph.D., talk to us and show us how to use ggplot2. Let's review some of the plots that were created with Andrew, highlighting some of the features of ggplot2.

As we work through the next sections of the R workshop, we will try to incorporate the plots as we go along.

Michelle