

Before we start any statistical analysis, we should really take a step back and get familiar and comfortable with our data. "Playing" around with it to ensure that you know what's in there. This may sound funny, but getting comfortable with your data by running descriptive statistics really does two things: One, you understand what's been collected and how; and second, gives you the opportunity to review the data and find any errors in it. Sometimes you may find an extra 1 added to the front of a number, or maybe a 6 instead of a 9, or any combinations of data entry errors. By playing around with your data and getting comfortable with it before running your analysis, you may find some of these anomalies.

For this workshop, I will provide you with a starting SAS program, which you can download [here](#). You will be asked to type in the PROCs as we work through them, but if you would rather, you always have the option of copying them from this post and pasting them into your SAS editor or code window. Please note, that there may be some nuances when you copy and paste. Any " will need to be changed in your SAS program!!!

My goals for this session are to review the following PROCedures:

- Proc Contents
- Proc Univariate
- Proc Freq
- Proc Means

## PROC CONTENTS

PROC CONTENTS provides you with the backend information on your dataset. One of the challenges in working with SAS, is that you do not have your dataset in front of you all the time. You read it in and it gets sucked into what I call the "Blackbox of SAS". Sometimes we either want to see the data - to ensure it's still there or simply to be comforted by the sight of it (we use PROC PRINT), or we want to see the contents of the dataset - so the formats of the variables and information about the dataset.

To do this we need to run a Proc CONTENTS on our file. This is the equivalent of the Variable View in SPSS.

```
Proc contents data=woodchips;  
Run;
```

What information were you able to see? Information about the actual SAS datafile along with formatting information about the variables contained in the datafile. View the output [here as a PDF](#).

If you make changes to the variables along the way, or if you add labels, rerun the Proc CONTENTS to ensure the changes were applied.

## PROC UNIVARIATE

Proc UNIVARIATE will be familiar to many of you as the PROC we use to see whether our data is normally distributed or not. This is one use for this PROCedure, but it is also very handy to get a sense for your data. It is one PROC that isn't used to its full capability, in my opinion.

Let's try running it as follows:

```
Proc univariate data=woodchips;  
var weight;  
Run;
```

Here is a link to the output saved as a [PDF file](#).

As you review the output you can see the variety of descriptive statistics that this PROC provides you. You should now have a very good feel for the data we are working with.

## PROC FREQ

Proc FREQ is used to create frequencies and cross-tabulations. In our dataset we only have one categorical variable, quality. To create a frequency table use the following code:

```
Proc freq data=woodchips;  
table quality;  
Run;
```

Here is the link to the output saved as a [PDF file](#).

Should you run a Proc FREQ on a variable such as **weight**? Why or why not?

## PROC MEANS

Proc MEANS is a fabulous and very versatile Proc to get a sense of your continuous variables, **weight**, in our example. Let's start with the overall mean by using this code:

```
Proc means data=woodchips;  
var wood_weight;  
Run;
```

Here is the link to the output saved as a [PDF file](#).

Note the default measures - N, Mean, StdDev, Min, Max

To add other descriptive measures, list them at the end of the Proc MEANS statement. For example, we want the standard error and the Sum:

```
Proc means data=woodchips mean stderr sum;  
var wood_weight;  
Run;
```

Here is the link to the output saved as a [PDF file](#).

One last piece of code for Proc MEANS: We want to see the means for each quality group.

```
Proc means data=woodchips;  
class quality;  
var wood_weight;  
Run;
```

Here is the link to the output saved as a [PDF file](#).

For more ways to use Proc MEANS, visit the following blog entry on SASsyFridays:

- [More Proc MEANS tips and tricks](#)
- [Exporting Proc MEANS results into Excel](#)