Lab #2: Data Input, Basic Functions:

A. Objectives:

1. PROC IMPORT an external dataset (CSV file)
2. Create a scatterplot
3. Compute Summary Statistics
4. Create Histograms

Reading from external files:

One of the strengths of SAS as a data analysis tool is its ability to read data from many sources, subset or combine data sets, and modify the datasets to accomplish various tasks. The most common types of external data sets used in SAS are EXCEL files (XLS extent), comma separated value files (CSV extent) and various space separate text files (PRN or TXT extent). A CSV file is actually a text file and can be read in any text reader (NOTEPAD or WORDPAD in Windows). In fact, the SAS files themselves, as well as the LOG and the LST files produced by a SAS by a batch submit, are also simple text files.[1]

The **PROC IMPORT** statement is the best way to enter external data sets. The CSV file we will be using is called “grades.csv”. Download and save it in your favorite folder and mark the complete path to it. Then use the following code to import it, making sure you put the correct path on the **DATAFILE** argument.

```
PROC IMPORT OUT= WORK.GRADES
            DATAFILE= "put the complete path here…\grades.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;
```

The **IMPORT** statement reads the dataset and stores it as the value designated by “OUT” in this case it will be saved as “Grades” in the library “Work”.

The **DBMS** statement defines the type of input SAS should be reading. The following table gives you all the possible choices. The REPLACE argument forces SAS to overwrite any older datasets with the same name.

Identifier	Input Data Source	Extension
ACCESS	Microsoft Access database	.MDB
DBF	dBASE file	.DBF

[1] After installing SAS you may find that clicking on a file with a LOG or LST extent opens them in SAS. You can request that Windows open these files by default in WORDPAD, which is much faster.

WK1	Lotus 1 spreadsheet	.WK1
WK3	Lotus 3 spreadsheet	.WK3
WK4	Lotus 4 spreadsheet	.WK4
EXCEL	Excel Version 4 or 5 spreadsheet	.XLS
EXCEL4	Excel Version 4 spreadsheet	.XLS
EXCEL5	Excel Version 5 spreadsheet	.XLS
EXCEL97	Excel 97 spreadsheet	.XLS
DLM	delimited file (default delimiter is a blank)	.*
CSV	delimited file (comma-separated values)	.CSV
TAB	delimited file (tab-delimited values)	.TXT

The **GETNAMES=YES|NO** statement for spreadsheets and delimited external files, determines whether to generate SAS variable names from the column names in the input file's first row of data. If you specify GETNAMES=NO or if the column names are not valid SAS names, PROC IMPORT uses the variable names VAR0, VAR1, VAR2, and so on. You may replace the equals sign with a blank.

The **DATAROW** argument tells SAS where to start reading for input data. In our case it is row 2 since row 1 is used for variable names.

Our dataset is now loaded in SAS. If we want to check it out a bit we need to click on the tab view and hit the “explorer” link. A folder based SAS environment appears. Our data set is in the “WORK” folder and it is named “Grades”. Click on it and the table will appear in an excel-like format.

In this specific dataset, the first row shows the maximum points available for each quiz. We need to remove this row so that our analysis is correct. To do that right click on the table and choose “Edit Mode” from the drop down menu. This will allow you to add, delete change etc. Let’s go ahead and delete the first row. For more information on the EXPLORER option follow the link to the official SAS support page:

http://support.sas.com/documentation/cdl/en/lrcon/62955/HTML/default/viewer.htm#a003166643.htm

This also allows you to see missing values and even change them. For now just leave them as they are.

When you are done with manipulating the table click close from the file tab. If you don’t do that the updates on the table will not be saved and SAS will stop working if you utilize that table in the next few lines of code.

Creating a simple Scatterplot:

As we discussed in class an easy way to compare two variables is using the idea of a scatterplot. We can use the command PROC GPLOT to create a scatterplot between the first two quizzes as follows:

```
TITLE ‘Scatterplot Q1 vs Q2’;
PROC GPLOT DATA= GRADES;
           PLOT Quiz1*Quiz2;
RUN;
```

The argument **PLOT** contains the variables that will be ploted with the first one on the x axis and the second one on the w axis. We can beautify the result by changing various arguments in the beginning of the code and we can even group by other variables. Suppose for example that we want to distinguish the males from females. We should then use:

```
SYMBOL1 V=circle C=blue;
TITLE ‘Scatterplot Q1 vs Q2 by Gender’;
PROC GPLOT DATA= GRADES;
           PLOT Quiz1*Quiz2=Gender;
RUN;
```

For more information on changing the view of a scatterplot refer to this nice tutorial from UCLA

https://stats.idre.ucla.edu/sas/modules/graphing-data-in-sas/

Creating a matrix Scatterplot:

There is a quick and easy way to create the pairwise scatterplots for all our variables using the **PROC sgscatter** command as follows:

```
TITLE ‘Scatterplot Matrix for Grades’;
PROC SGSCATTER DATA= GRADES;
               MATRIX Quiz1 Quiz2 Quiz3 Quiz4 Quiz5 Quiz6 Midterm/
               GROUP = Gender;
RUN;
```

The **MATRIX** argument lists the variables you want to compare pairwise. The **GROUP** variable tells SAS how to split the points in different categories, in this case based on gender.

Creating a simple histogram:

PROC GCHART is one of a number of graphical procedures often used for data exploration and examination. This procedure can be used to produce a number of different styles of graphic depending on the statements that are included. The variable to be processed is named in the statement. Some of these statements are:

HBAR – a horizontal bar chart that will also include information on frequency, percent (relative frequency), cumulative frequency and cumulative percent (relative cumulative frequency).

VBAR – a vertical bar chart often called a histogram.

BLOCK – produces a 3D plot with two variables (sugars and shelf) on a surface and blocks who’s height represent a third “response” variable. The default for the response is frequency of occurrence in each combination of the first two variables. The response variable can also be percents, sums or means.

PIE, STAR, DONUT – yields pie chart and similar charts

Let’s use this PROC chart command to create a simple histogram for the grades of the second quiz.

```
TITLE ‘Histogram for Quiz2’;
PROC GCHART DATA= GRADES;
          VBAR Quiz2/midpoints 1 to 8 by 1;
RUN;
```

PROC GCHART OPTIONS: A number of options are available to modify the appearance of charts. We will not discuss size and resolution options here, but some other important options are listed below. The options below are placed on the chart type statement following a slash (i.e. /).

The **MIDPOINTS** argument is the most useful when creating histograms since the width of a histogram clearly affects the picture. By default SAS will determine groupings, or midpoints for groupings. However, you can set your own midpoints with the MIDPOINT option and try the examples below by changing the appropriate line:

```
   vbar Quiz2 / midpoints= 0 to 8 by 2; * by range and interval;
   vbar Quiz2 / midpoints= 4 7;         * by unequal spacing;
```

Suppose now that we want again the data split between male and female in a stacked format. The code we should then use is:

```
TITLE ‘Histogram for Quiz2 by gender’;
PROC GCHART DATA= GRADES;
           VBAR Quiz2 / subgroup=Gender
           CLIPREF
           FRAME TYPE=FREQ
RUN;
```

Notice that since I did not specify midpoints SAS did its best to break the values into “logical” bins.
For a complete discussion of all the features of vbar refer to the official SAS guide page found here:

http://support.sas.com/documentation/cdl/en/graphref/63022/HTML/default/viewer.htm#gchart-bar.htm

Summary Statistics Review:

As we learned last week to do a complete summary statistics on a dataset we use the command **PROC UNIVARIATE**. In our case let’s do a quick summary statistics procedure for quiz 3. Since the Univariate procedure produces many tables lets focus only on the Basic Statistical Measures and the Quantiles. The following code will produce those for us.

```
TITLE ‘Summary statistics for Quiz 2 and Quantiles’;
ODS Select BasicMeasures Quantiles;
PROC UNIVARIATE DATA= GRADES;
                VAR Quiz2;
RUN;
```

The **ODS** argument chooses the output we want, in this case the Basic Statistics Measures and the Quantiles.
We note here that we can use **PROC UNIVARIATE** to create a histogram as follows:

```
TITLE ‘Histogram for Quiz2’;
ODS graphics off;
PROC UNIVARIATE DATA= GRADES noprint;
                Histogram Quiz2;
RUN;
```

The command **noprint** suppresses the summary statistics and produces only the graph. Once again, if one wants to change the bin width one can specify various things like for example the endpoints. Try the following code:

```
TITLE ‘Histogram for Quiz2’;
ODS graphics off;
PROC UNIVARIATE DATA= GRADES noprint;
                Histogram Quiz2/
                Endpoints=0 to 10 by 1
                rtinclude;

RUN;
```

The command **rtinclude** tells SAS to include all the values that fall on the boundary to the bin on the left (I know it is a bit confusing but really it means that you include the right bound of a bin into that bin). Remove that line and see what happens.