

SAS Guide

Data Entry, Summary, and One Sample Analysis

Library

First set the library –

```
libname stats 'a:\';  
run;
```

Entering Data

First you need a **data** step, then an **input** statement, and then a **datalines** command.

Example 1 – One sample data – the dataset is **app** and the variable is **score**.

```
data stats.app;  
input score;  
datalines;  
4  
5  
8  
8  
9  
6  
10  
7  
6  
6  
;  
run;
```

To see if you have entered the data correctly, you can use the **proc print** command.

```
proc print data=stats.app;  
run;
```

If you have a large number of observations you can use the option of printing only the first few with

```
proc print data=stats.app (obs=10);  
run;
```

If you did not create the dataset yourself, use the **proc contents** command to find out how many observations are in the data set and what the variable labels are.

```
proc contents data=stats.app;  
run;
```

Example 2 – Paired Data – when you have paired data, you want to enter the data in two columns and add a line after the data step that takes the difference. The following example looks at matched subjects that differed by whether they received instruction or not. The **input** statement has two variables and after the **datalines** command, data is separated by a space.

```
data stats.dental;
input instruct noninst;
diff=instruct-noninst;
datalines;
1.5 2.0
2.0 2.0
3.5 4.0
3.0 2.5
3.5 4.0
2.5 3.0
2.0 3.5
1.5 3.0
1.5 2.5
2.0 2.5
3.0 2.5
2.0 2.5
;
```

If a data set has been already created and you want to modify it and create new variables, use the **set** command. For example suppose a dataset called **diet** consists of two variables with **w1** and **w2** for the weights before and after the diet. To create the new variable **diff** to be the difference, so the following:

```
data stats.diet2;
set stats.diet;
diff=w1-w2;
run;
```

You are making the new data set **diet2** in the **stats** library from the old data set **diet**. The data set **diet2** will have all the variables from **diet**, plus the new variable **diff**.

Two or more sample data is entered with the first variable as the outcome variable and the second variable identifies the treatment or group. Below, the outcome variable is the hemoglobin level (labeled **level** in the **input** command) and the grouping or treatment variable is labeled **group** in the **input** command.

```
data stats.hemo;
input level group;
datalines;
14.4 1
14.2 1
13.8 1
16.5 1
14.1 1
16.6 1
15.9 1
```

```

15.6 1
14.1 1
15.3 1
15.7 1
16.7 1
13.7 1
15.3 1
14.0 1
17.4 2
16.2 2
17.1 2
17.5 2
15.0 2
16.0 2
16.9 2
15.0 2
16.3 2
16.8 2
;
run;

```

Summary Analyses

```

proc univariate data=stats.datafile normal plots;
var varname;
histogram;
QQplot / normal( mu=est sigma=est color=BLUE l=1
                w=1);
inset normal;
run;

```

The above will produce summary statistics, perform the normality tests (**normal**), do a stem leaf plot (**plots**), produce a histogram (**histogram**), and produce a QQ plot (**QQplot**).

The above will perform all 3 tests for a **univariate** procedure – the **ttest**, the **sign test** and the **Wilcoxon signed rank test** against the null that the center is 0. To test that the mean or median is a number other than 0 (for example 5), add the command **mu0=5** so that you have

```

proc univariate data=stats.datafile normal mu0=5;

```

To make a boxplot – you have two options – use **Analyst**, or you can use the following code. This makes a boxplot for a one sample data set called **stats.cout** and the variable to be summarized is **OUTPUT**. If you use this template, you merely change where you see **STATS.cout** and where you see **OUTPUT**. The first part (**proc sql**) creates a dummy data set of name **WORK._TMP_0** with a fake variable to place the boxplot on a horizontal axis. The **axis1** section tells SAS not to label the x-axis though. The **proc boxplot** makes the boxplot with outliers (the **schematic** option), and black lines (**cboxes=black**), and the **haxis=axis1** tells SAS to look for the definition of **axis1** which makes it not print a label on the horizontal axis. The **proc datasets** command deletes the temporary dataset **WORK._TMP_0**. The sas code below is in the file **boxplot.sas** on the webpage. You can open that in the Program Editor and edit appropriately.

```

proc sql;

```

```
create view WORK._TMP_0 as
  select *,1 as _dummy_
  from STATS.cout
;
axis1 major=none value=none label=none;

proc boxplot data=WORK._TMP_0 ;
  plot (OUTPUT)*_DUMMY_ /boxstyle = SCHEMATIC cboxes=black haxis=axis1;
run;
proc datasets lib=work nolist;
  delete _TMP_0 /memtype=view;
run;
```

To bring a graphic from SAS to Word, make the graphic window the active window in SAS, and click **Edit/Copy**. Then in Word, you want to click **Edit/Paste Special** and then chose the option of **Device Independent Bitmap**. That should make your graph visible and printable in Word.