Grand Valley State University
ScholarWorks@GVSU

Honors Projects

Undergraduate Research and Creative Practice

1-1-2013

SAS Help Guide for Version 9.3

Ruth Farnsworth
Grand Valley State University

Follow this and additional works at: http://scholarworks.gvsu.edu/honorsprojects

Recommended Citation

Farnsworth, Ruth, "SAS Help Guide for Version 9.3" (2013). *Honors Projects.* Paper 187.
http://scholarworks.gvsu.edu/honorsprojects/187

This Open Access is brought to you for free and open access by the Undergraduate Research and Creative Practice at ScholarWorks@GVSU. It has been accepted for inclusion in Honors Projects by an authorized administrator of ScholarWorks@GVSU. For more information, please contact scholarworks@gvsu.edu.

Honors Senior Project

By Ruth Farnsworth

SAS HELP GUIDE FOR VERSION 9.3

Introduction to SAS features

- **Accessing SAS on Grand Valley computers**
  1. Double click the **Application** icon on the desktop
  2. In the open window, double click the **Statistics** folder
  3. Double click the **SAS 9.3 Revised** folder, then click the **SAS 9.3 (English)** icon

Inputting data

- **Inputting a simple, numeric data set**
  1. Click Inside the **Editor** box
  2. Type the word *data* and then the name of your data set.
     - NOTE: Always end a line of code with a semicolon (;)
     - Ex – *data simple;*
  3. On the next line, type *input* and then your variable name(s)
     - Ex- *input x y;*
     - On the next line, type *cards* or *datalines*. End with a semicolon and start a new line. You can now enter your data. When all the data is entered start a new line and end with a semicolon.
       - *Cards;*
         1 2
         2 3
         3 4
         ;
- **Inputting a Non Numeric (character) data set**
  1. Click Inside the **Editor** box
  2. Type the word *data* and then the name of your data set.
  3. On the next line, type *input* and then your variable name(s)
     - NOTE: after the name of a character variable, type a dollar sign ($) to accurately read in the characters

    - Ex- *input x$ y;*

4. On the next line, type *cards* or *datalines*. End with a semicolon and start a new line. You can now enter your data. When all the data is entered start a new line and end with a semicolon.

- **Inputting data in a fixed format**
  1. Click Inside the **Editor** box
  2. Type the word *data* and then the name of your data set.
  3. On the next line, type *input* and then your variable name(s)
     - NOTE: after the name of a variable, type the length of a variable as a range of numbers.
     - Ex- *input x$ 1-3 y;*
       - In this case, the character variable x will be whatever the first three values of the inputted data are.
         a. If the inputted data looked like this
            cards;
            A12345
            B12345
            C12345
            ;
            The values of the variable x would be (A12, B12, C12)
     - NOTE: You can also specify the variable length by typing a number and a period after a variable name.
     - Ex- *input x$ 1. y;*
       - In this case, the variable x is specified to only be 1 character long.
         a. If the inputted data looked like this
            cards;
            A12345
            B12345
            C12345
            ;
            The values of the variable x would be (A, B, C)

- **Importing data from an external text file**
  1. Click Inside the **Editor** box
  2. Type the word *data* and then the name of your data set.
  3. On the next line, type *infile* and then the destination of your file inside ‘ ‘
     - Ex- *infile 'N:\ inportme.txt';*

4. On the next line, type *input* and then your variable name(s)
5. On the next line, type *cards* or *datalines*. End with a semicolon and start a new line. You can now enter your data. When all the data is entered start a new line and end with a semicolon.

- **Importing comma delimited data from an external file**
  1. Do steps 1-3 for importing data from an external text file.
  2. After specifying the file destination, type *delimiter=','* to import data separated by commas.
- **Using the import wizard**
  1. Click **file** and then **import data**
  2. Check **Standard data source** and choose **Microsoft Excel Workbook**, then hit **next**
  3. Click **browse** and locate the excel file you wish to import, then click **ok**
  4. Click **next**
  5. Under the box titled **member** type what name you wish to call your data set, then click **finish**

Summary Statistics

- **Creating a new variable**
  1. Create a new data set by typing *Data* and the data set name
  2. On the next line type *set* and the name of the data set you wish to add a variable to
     - Ex- *set simple;*
  3. On the next line create a new variable. This can be done in many ways
     - *Xsquared =x*x;* will create a squared variable x
     - *Xplusy = x + y;* is the sum of variables x and y
     - *Percent = (x/y) *100;* Is the ratio of x to y as a percent
- **Printing a data set**
  1. Type *proc print* and the name of the data set you wish to print
     - Ex- *proc print data=simple;*
     - NOTE: Two of the most common options are *noobs* and *round*
       - *Noobs* suppresses the obwervation number column in the output.

        - *Round* rounds the values in the output to only two decimal places
            a. *Ex- proc print data=x noobs round;*

2. Type *var and the* names of the variables you want to print
    - Ex- *var x y;*
3. To sum the values of a variable type *sum* and the vairables of interest below the var statement.
    - Ex- *sum x;*

- **How to get basic statistics (mean, standard deviation, quartiles, etc)**
    1. To find descriptive statistics with Proc univariate, type *proc univariate data=* data name on a new line.
        - Ex- *proc univariate data=simple;*
    2. If the variable(s) of interest need to be separated by a categorical variable, use the *by option* to specify that variable.
        - Ex- *by y;*
    3. Type var and the names of the variables you want to analyze
        - Ex- *var x;*

    OR

    4. To find specific descriptive statistics with Proc means, type *proc means data=* data name
        - Ex- *proc means data=simple;*
        - NOTE: To specify which descriptive statistics you need, type the name of the statistic after the *data=* statement.
            - Ex- *proc univariate data=simple RANGE STD MAX MEAN MIN MODE MEDIAN Q3 Q1 QRANGE;*
            - The output from this statement contains the range, standard deviation, maximum, mean, minimum, mode, median, quartile three, quartile one, and the interquartile range.
- **Making Frequency tables**
    1. To construct frequency tables with Proc freq, type *proc freq data=* data name on a new line.
        - Ex- *proc freq data=simple;*

2. Type *weight* and the name of the variable that records the number of observations within a category. In many cases this variable will be *count*
    - Ex- *weight count;*
3. Type *tables* and the names of the categorical variables of interest. You can construct a single table by typing the variables name, or you can construct a two variable table by typing the two variable names separated by a *
    - Ex- *tables haircolor;*
    - Ex- *tables haircolor*eyecolor;*
    - NOTE: to omit some portions of the output table, type a / after the tables statement and which numbers you want to omit
        - Ex - *tables haircolor / nofreq norow nocol nocum nopercent;*
        - *Nofreq* will omit frequencies, *norow* will omit row percentages, *nocol* will omit column percentages, *nocum* will omit cumulative percentages, and *nopercent* will omit all percentages.

<u>Creating graphs</u>

- **Creating Scatterplots**
    1. To create a scatterplot using proc gplot, type *proc gplot* and the name of the data set
        - Ex-*proc gplot data=simple;*
    2. On the next line type *plot* and the names of the two quantitative variables separated by a *
        - Ex- *plot height*weight;*
- **Creating bar charts and pie charts**
    1. To create a bar chart or pie chart using proc gchart, type *proc gchart* and the name of the data set
        - Ex-*proc gchart data=simple;*
    2. If the charts need to be separated by a variable type *by* and the variables name
        - Ex- *by gender;*
    3. To make a bar chart, type *vbar* and the name of the variable or variables of interest
        - Ex- *vbar haircolor;*
    4. To make a pie chart, type *pie* and the variable or variables of interest

  - Ex- *pie haircolor;*

- **Creating Histograms**
  1. To create histograms with Proc univariate, type *proc univariate data=* data name on a new line.
     - Ex- *proc univariate data=simple;*
  2. Type var and the names of the variables you want to analyze
     - Ex- *var x y;*
  3. Type *histogram* and the names of the variable(s) that require histograms
     - Ex- *histogram x;*
- **Creating boxplots**
  1. To create boxplots with Proc boxplot, type *proc boxplot data=* data name on a new line.
     - Ex- *proc boxplot data=simple;*
  2. Type *plot* and then the quantitative variable name and the categorical variable name separated by a *
     - Ex- *plot weight*haircolor;*
     - NOTE: this proceadure is useful for creating side by side boxplots only

     OR
  3. Proc univaritate automatically outputs boxplots (see section on descriptive statistics for how to use proc univariate)

<u>Running tests</u>

- **One sample T test**
  1. To check the assumptions in sas, construct a normal probability plot using proc univariate.
     - After entering the variable of interest in a proc univariate statement, on the next line type *qqplot;* The resulting graph will test the normality of the data
  2. To run the test, type *proc t test* and the name of the data set. After the data name, enter H0 and the value of your null hypothesis,
     - Ex-*proc t test data=simple h0=5;*

     NOTE: To adjust this for a two independent sample t test, type *class* and the name of the categorical variable of interest.

    - Ex- *class gender;*

3. Type *var* and the names of the variable of interest
    - Ex- *var x;*

   NOTE: to run a paired sample t test, type *paired* instead of *var*, and the names of the two variables separated by a *
    - Ex- *paired group1*group2;*

- **Anova tests**
    1. Checking assumptions
        - Use proc univariate to make a normal probability plot (see one sample t test)
    2. To run the anova test, type *proc glm* and the name of the data set of interest
        - Ex- *proc glm data=simple;*
    3. Type class and the name of the categorical variable of interest
        - Ex- *class explanitory;*
    4. Type *model* and the names of the response and explanitor variables separated by an equal sign
        - Ex- *model response=explanatory;*
- **Chi Squared tests**
    1. Checking assumptions
        - Using proc freq to check expected cell counts (see section on creating frequency tables)
    2. To run the Chi Squared test, after entering the names of the variables of interest, type */chisq;*
- **Simple Linear Regression**
    1. TO check assumptions on simple linear regression, make a scatterplot to ensure that your variables have a linear relationship
        - See section on making a scatterplot
    2. To run the simple linear regression, type *proc reg* and the name of the data set
        - Ex- *proc reg data=simple;*
    3. Type *model* and the two variables of interest, separated by an equals sign
        - Ex-*model height=weight;*

Examples

The following are several examples that demonstrate the correct syntax of code in relation to various sas tasks. From the UCLA website in SAS learning modules (see works cited).

1. Data Entry Example

Here is the SAS program that makes a data file called **kids**. It contains three families (**famid**) each with three kids. It contains the family ID, the name of the kid, the order of birth (1 2 3 for 1st, 2nd, 3rd), and the age, weight and sex of each kid.

```
DATA kids;
 LENGTH kidname $ 4 sex $ 1;
 INPUT famid kidname birth age wt sex ;
 CARDS;
1 Beth 1  9  60  f
1 Bob  2  6  40  m
1 Barb 3  3  20  f
2 Andy 1  8  80  m
2 Al   2  6  50  m
2 Ann  3  2  20  f
3 Pete 1  6  60  m
3 Pam  2  4  40  f
3 Phil 3  2  20  m
;
RUN;

PROC PRINT DATA=kids;
RUN;
```

Here is the output of the **proc print**.

```
OBS    KIDNAME    SEX    FAMID    BIRTH    AGE    WT
 1      Beth       f       1        1       9     60
 2      Bob        m       1        2       6     40
 3      Barb       f       1        3       3     20
 4      Andy       m       2        1       8     80
 5      Al         m       2        2       6     50
 6      Ann        f       2        3       2     20
 7      Pete       m       3        1       6     60
 8      Pam        f       3        2       4     40
 9      Phil       m       3        3       2     20
```

2. Proc Means Example

We can request multiple statistics at once. The command below gets the mean, standard deviation and age (**mean std** and **N**) for **age** and **wt** within each family.

```
PROC MEANS DATA=kids mean std min;
CLASS famid;
```

```
VAR age wt;
RUN;
```

The results below shows the output of the **proc means**.

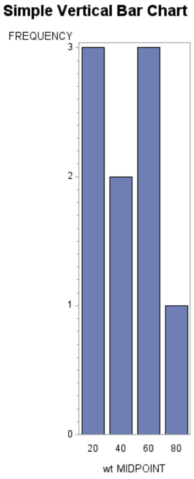
```
FAMID  N Obs  Variable  N         Mean     Std Dev     Minimum
--------------------------------------------------------------
    1      3  AGE       3    6.0000000   3.0000000   3.0000000
              WT        3   40.0000000  20.0000000  20.0000000
    2      3  AGE       3    5.3333333   3.0550505   2.0000000
              WT        3   50.0000000  30.0000000  20.0000000
    3      3  AGE       3    4.0000000   2.0000000   2.0000000
              WT        3   40.0000000  20.0000000  20.0000000
--------------------------------------------------------------
```

3. **Graphing Example**

We create vertical Bar Charts with **proc gchart** and the **vbar** statement. The program below creates a vertical bar chart for **kids**.

```
TITLE 'Simple Vertical Bar Chart ';
PROC GCHART DATA=kids;
 VBAR wt;
RUN;
```

This program produces the following chart.



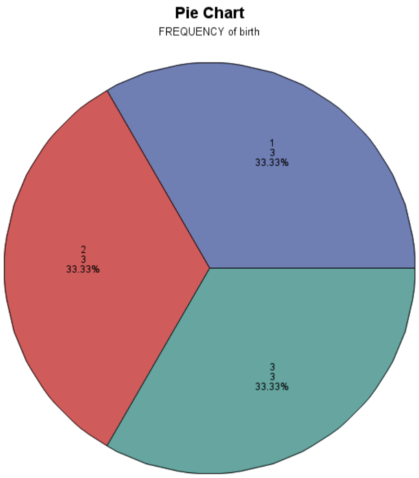
This graph shows the distribution of weights among the kids in the sample. The **vbar** statement produces a vertical bar chart, and while optional the **title** statement allows you to label the chart.

You can produce a pie chart by replacing **vbar** in the above example with **pie**. The **value=**, **percent=**, and **slice=** options control the location of each of those labels.

```
TITLE 'Pie Chart';
PROC GCHART DATA=kids;
PIE birth/ DISCRETE VALUE=INSIDE PERCENT=INSIDE;
RUN;
```

value=inside causes the frequency count to be placed inside the pie slice. **Value=outside** would cause the frequency to be placed outside the slice.
percent=inside causes the percent to be placed inside the pie slice. **Percent=outside** would cause the percent to be placed outside the slice.

This program produces the following pie chart.



This chart represents the percent of children who make up each birth order; first born, second born, third born.

Works Cited

Agresti, Alen, Christine Franklin, and Joeseph Blitzstein. Statistics : The Art and Science of Learning from Data. N.p.: Pearson Education Canada, 2012. N. pag. Print.

Introduction to SAS. UCLA: Statistical Consulting Group. UCLA, n.d. Web. Apr. 2013 <http://www.ats.ucla.edu/stat/sas/notes2/>.

SAS: The Power to Know. Base SAS(R) 9.2 Procedures Guide: Statistical Procedures, Third Edition. SAS, n.d. Web. Apr. 2013. <http://support.sas.com/documentation/>.

SAS: The Power to Know. SAS OnlineDoc 9.1. SAS, 2002-2003. Web. Apr. 2013. <http://support.sas.com/91doc/docMainpage.jsp>.