

Introduction to Tables and Graphs in SAS

- 1) Descriptive statistics are valuable as analytical tools and in developing data
 - in the introduction, we went over basic reporting commands
 - frequencies: PROC FREQ
 - univariate statistics: PROC MEANS and PROC UNIVARIATE
 - correlations: PROC CORR
 - this class will
 - discuss these routines in more detail, focusing on how to produce simple tables and cross-tabulations (conditional tables) of statistics
 - introduce a general tabulation procedure, PROC TABULATE
 - discuss some plotting and graphing procedures
- 2) Printing information from a data set
 - the simplest reporting procedure is PROC PRINT, which prints the values from a data set
 - the syntax for this procedure is

```
PROC PRINT <option-list>;  
VAR <variable-list>;
```

- this procedure is helpful for analyzing modestly sized data sets
 - you need to be careful not to call this procedure with large data sets, such as the PSID, because the resulting output can overwhelm the OUTPUT window
 - the VAR statement will help to control the width of the output
 - data set options and general options are two ways to control the length of the output
- 3) Univariate statistics
 - in empirical analyses, we often include a table of univariate statistics to document the data set; we also want to be sure that the measures that we are using have reasonable values and distributions (e.g., all the values are in range, some variation in the values)
 - we can use the MEANS and UNIVARIATE procedures to generate these statistics
 - the advantage of these procedures is that they produce a great deal of standard output with very few commands
 - PROC MEANS
 - the syntax of the PROC MEANS statement is

```
PROC MEANS <option-list> <statistic-keyword-list>;
```

- if you don't specify any statistic keywords (the default), the MEANS procedure reports the number of non-missing values, means, standard deviations, minimum values, and maximum values for all of the variables in your data set or all of the variables listed in an accompanying VAR statement
- you can also specify alternative statistics such as the median (MEDIAN), the sum (SUM), variance (VAR), standard error (STDERR), and coefficient of variation (CV); a complete list

is available at

<http://support.sas.com/documentation/cdl/en/proc/59565/HTML/default/a000146729.htm>

- the MEANS procedure generally outputs one line per variable; however, it may output multiple lines if your variables are labeled or if you request lots of statistics
- PROC UNIVARIATE
 - the syntax for the PROC UNIVARIATE statement is

```
PROC UNIVARIATE <option-list>;
```

- although it is not required, it is a good idea to include a VAR statement specifying the variables for which statistics will be generated
- the main difference between the UNIVARIATE and MEANS procedures is that the UNIVARIATE procedure estimates many more statistics, usually 1-2 pages per variable
- to produce simple distributions of categorical variables we can use the FREQUENCY procedure
 - the syntax for a univariate distribution is:

```
PROC FREQ;  
TABLE <row-var>;
```

where *row-var* is the categorical variable

- this will produce a table with rows containing the numbers, percentages, cumulative numbers, and cumulative percentages of observations in each category of *row-var*

4) Two-way cross-tabulations

- cross-tabulations are tables of statistics that are computed conditionally
- for example, in an empirical analysis, we might want to list simple descriptive statistics or distributions of variables for separate sub-samples of our larger analysis sample (e.g., list results separately for women and men or separately by ethnicity)
- many statistical procedures in SAS have simple commands that allow for conditional processing
- PROC FREQ
 - in a previous lecture we showed how the FREQUENCY procedure could be used to produce two-way tables; the syntax is

```
PROC FREQ;  
TABLE <row-var> * <col_var> / <options>;
```

- this will produce a two-way table with cells corresponding to each possible combination of *row-var* and *col-var*
- be careful in your table requests
 - ◊ you generally want to avoid using this with variables with lots of different outcomes, especially continuous variables (note: if needed, you can specify formats to control the display of continuous variables)

- ◇ also, to minimize output, you generally want to use the row variable to be variable with the most potential outcomes
- in the output, each cell will contain
 - ◇ the cell frequency (the number of observations in that cell)
 - ◇ the table percentage (cell observations as a percent of all non-missing observations in the dataset)
 - ◇ the row percentage (cell observations as a percent of observations with a given value of *row-var*), and
 - ◇ the column percentage (cell observations as a percent of observations with a given value of *col-var*)
- options can be specified to suppress the calculation of some of these statistics
 - ◇ NOFREQ suppresses the printing of cell frequencies
 - ◇ NOPERCENT suppresses the printing of table percentages
 - ◇ NOROW suppresses the printing of row percentages
 - ◇ NOCOL suppresses the printing of column percentages
- you can also request formal statistics of the association between the row and column variables, such as the Pearson and Spearman correlation coefficients; the general option for these is MEASURES
- unless you specify otherwise, the FREQUENCY procedure ignores missing values; if you would like missing values to be included as an additional category, include the MISSING option
- the CLASS statement in the MEANS and UNIVARIATE procedures
 - the MEANS and UNIVARIATE procedures also will produce conditional statistics
 - in newer versions of SAS, this can be done using a CLASS statement
 - the syntax for this statement is

```
CLASS <variable-list>;
```

- with this, the procedures will calculate statistics for subgroups defined by the CLASS variables
 - ◇ if there is just one CLASS variable, the procedures will calculate statistics for each different value
 - ◇ if there are two CLASS variables, the procedures will calculate statistics for each observed combination of values
- for example, suppose that you had a data set with two categorical variables, *gender* and *education*

```
PROC MEANS;
  CLASS gender education;
```

produces simple descriptive statistics for each combination in the cross-product, *gender x education*

- tests of differences of means, the TTEST procedure
 - when we calculate conditional means and other statistics, we are often interested in formally testing whether these statistics differ across groups

- the TTEST procedure tests for differences in means and variances across two groups
- the syntax is

```
PROC TTEST;
  VAR <variable-list>;
  CLASS <class-var>;
```

- the CLASS variable needs to be restricted to two outcomes
- if the VAR statement is omitted, TTEST calculates tests for all of the numerical variables in your data set
- PROC TABULATE
 - the TABULATE procedure is a general procedure for generating tables, as such it combines features of the MEANS and FREQUENCY procedures—but with a lot more flexibility, including multiple conditioning
 - the general syntax is

```
PROC TABULATE <option-list>;
  CLASS <class-variable-list>;
  VAR <analysis-variable-list>;
  TABLE <page-expression,> <row-expression,> <column-expression>
  / <table-options>;
```

- the CLASS statement is needed to identify potential conditioning variables
- the VAR statement is needed to identify potential outcome variables
- the actual use of the variables depends on the TABLE statement
- expressions consist of variables, operators, statistical keywords, and formatting instructions
- some TABLE operators
 - ◊ , used to distinguish page, row, and column dimensions
 - ◊ * cross-product conditioning within a dimension
 - ◊ () used to group elements
- consider a dataset with two categorical (CLASS) variables c and d and two analytical variables x and y
- example #1: specification

```
PROC TABULATE;
  CLASS c d;
  VAR x y;
  TABLE c, d * (x y) * (MEAN STD);
```

- would produce a table of means and standard deviations of x and y conditional on the possible combinations of c and d , where the values of c would appear in the rows and d would appear in the columns
- example #2: changing the TABLE statement to

```
TABLE c, (ALL d) * (x y) * (MEAN STD);
```

would add an initial column where the means of x and y were only conditioned on c (i.e., weren't conditioned on d)

- example #3: changing the TABLE statement to

```
TABLE (ALL c), (ALL d) * (x y) * (MEAN STD);
```

would add an initial column where the means of x and y weren't conditioned on c

- example #4: changing the TABLE statement to

```
TABLE c * d, (x y) * (MEAN STD);
```

would produce the same statistics as example #1; however, all of the conditioning would take place by rows instead of by rows and columns

- example #5: changing the TABLE statement to

```
TABLE c, d * N;
```

would produce frequencies of the combinations of c and d

- additional information on PROC TABULATE, including examples and more elaborate formatting options, is available at

<http://support.sas.com/documentation/cdl/en/proc/59565/HTML/default/a000146759.htm>

5) Creating data sets that contain statistics

- with standard statistical procedures
 - the MEANS procedure allows you to include an OUTPUT statement
 - the syntax for this statement is

```
OUTPUT OUT=<SAS data set> <other specifications>;
```

- unless you specify otherwise, the OUTPUT statement for the MEANS procedure will generate one observation
 - ◇ per standard statistic (N, MEAN, STD, MIN, MAX); the type of statistic would be identified with the `_STAT_` variable
 - ◇ per possible classification group
- you can request specific statistics; to do this, you would list the type of statistic that you want and name it
- for example,

```
PROC MEANS;  
  CLASS c d;  
  VAR x y;  
  OUTPUT OUT=tsum  
         MEDIAN=x_med y_med;
```

- produces output with one observation per possible classification combination (all, classified by c , classified by d , classified by c and d) and with the MEDIANs for x and y stored in the variables x_med and y_med , respectively
 - note that requesting specific statistics results in one observation per classification group
- OUTPUT statements can also be used with the UNIVARIATE and FREQUENCY procedures
- the SUMMARY procedure can also be used to produce data sets with the same types of statistics as the MEANS procedure
 - the primary difference between the two procedures is PROC SUMMARY does not produce listing output
 - thus, the SAS data set *tsum* from the previous example could have also been produced with the for example,

```
PROC SUMMARY;
  CLASS c d;
  VAR x y;
  OUTPUT OUT=tsum
         MEDIAN=x_med y_med;
```

6) Graphs and SAS/GRAPH procedures

- SAS has an extensive graphics capability through the SAS/GRAPH module
- SAS/GRAPH supports many types of graphs, including bar charts, line charts, pie charts, and maps
- Scatter and line plots, the GPLOT procedure
 - PROC GPLOT produces two-way scatter and line plots; it will also produce bubble plots
 - the syntax for scatter and line plots is

```
PROC GPLOT <options>;
  PLOT <plot-request> / <plot-options>;
```

- the plot request for an outcome variable, y , along the vertical axis and an independent variable, x , along the horizontal axis would be $y * x$
- you can also specify a series of conditional plots
 - ◊ let c be the conditioning variable
 - ◊ the request for plotting y against x conditional on different values of c would be $y * x = c$
- unless you specify otherwise, the PLOT command in GPLOT produces scatter plots
- specifying line plots is a little tricky
 - ◊ you need to define the SYMBOLs for the PLOT and specify that the SYMBOLs will INTERPOLATE between values
 - ◊ the syntax for requesting that the symbols for the first category in your graph be joined is

```
SYMBOL1 I=JOIN;
```

- example of a simple scatter plot

```
PROC GPLOT;  
  PLOT y * x;
```

- example of a simple line plot

```
PROC GPLOT;  
  PLOT y * x;  
  SYMBOL1 I=JOIN;
```

- unless otherwise specified, output from SAS/GRAPH procedures will be directed to a graph window; from there you can save the graphs as JPEG or other types of graphics files, using FILE • EXPORT AS IMAGE
- SAS/GRAPH involves numerous procedures; only some simple examples for a particular type of graph have been shown here; for more information about the SAS/GRAPH procedures go to <http://support.sas.com/documentation/onlinedoc/graph/index.html>

7) Looking for other ways to store your output

- SAS sends listing output, including graph output, to its Output Delivery System (ODS)
- the default for the ODS is the LISTING source
 - results from most procedures are sent to the OUTPUT window
 - results from SAS/GRAPH are sent to a GRAPH window
- it is possible to redirect SAS output to other sources through ODS commands; the ODS destinations are
 - LISTING (the default)
 - HTML – this destination will produce HTML code and graphics files
 - PDF – creates a PDF file with the output
 - RTF – creates a RTF (MS Word) file with the output
- syntax
 - there are two ODS commands: one to open a destination and one to close it
 - to open a destination, the command is

```
ODS <destination> <options>;
```

where the <destination> is LISTING, HTML, PDF, or RTF and the <options> are style options for the output

- one of the options is to specify a FILE, e.g., FILE = '<windows-file>'
- the destination is closed when SAS terminates; if you want to close a destination before then and write the output to it (i.e., have the output available to read), you type an ODS CLOSE command

```
ODS <destination> CLOSE;
```

- example: to redirect the output from the LISTING to a PDF file, *c:\temp\ex1.pdf*, you would type

```
ODS LISTING CLOSE;      /* closes default LISTING dest.    */
ODS PDF FILE='c:\temp\ex1.pdf';
                        /* opens PDF as an ODS destination */
```

SAS statements that produce output here...

```
ODS PDF CLOSE;         /* closes PDF dest.; writes output  */
ODS LISTING;           /* re-opens LISTING as destination  */
```

- for more information on the SAS Output Delivery System, go to <http://support.sas.com/documentation/cdl/en/odsug/59523/HTML/default/a000933274.htm>