STAT 425 and STAT 625
Statistical Software

Lecture 10

Descriptive Statistics and Creating Tabular Reports

Saida Zardi

Summarizing your Data using PROC MEANS

Descriptive statistics give a feel of the data

Summary statistics

PROC MEANS

Summarizing your Data using PROC MEANS

MEANS procedure:

Proc Means options

Control options:

MAXDEC = n *specifies the number of decimal places to be displayed*

MISSING *treats missing values as valid summary groups*

Summarizing your Data using PROC MEANS

Summary statistics options:

MAX	*maximum value*
MIN	*minimum value*
MEAN	*average value*
MEDIAN	*median value*
MODE	*mode value*
N	*number of non missing values*
NMISS	*number of missing values*
RANGE	*range*
STDDEV	*standard deviation*
SUM	*sum*

Summarizing your Data using PROC MEANS

Proc means data = 'data set name' ; (no other statement)

Summary statistics for all variables in the data set:

N, mean, standard deviation, minimum, maximum

Summarizing your Data using PROC MEANS

Example:

```
*16-1;
title "PROC MEANS With All the Defaults";
proc means data=Learn.Blood;
run;
```

PROC MEANS With All the Defaults

The MEANS Procedure

Variable	Label	N	Mean	Std Dev	Minimum	Maximum
Subject		1000	500.5000000	288.8194361	1.0000000	1000.00
WBC		908	7042.97	1003.37	4070.00	10550.00
RBC		916	5.4835262	0.9841158	1.7100000	8.7500000
Chol	Cholesterol	795	201.4352201	49.8867157	17.0000000	331.0000000

Summarizing your Data using PROC MEANS

PROC MEANS Optional statements:

BY	variable - list	performs separate statistical analyses for each level of the variables in the list. Data must be sorted first by these variables (use PROC SORT).
CLASS	variable - list	performs separate statistical analyses for each level of the variables in the list. The output is more compact than with the BY statement. Data don't have to be sorted first.
VAR	variable - list	Specifies each numeric variables to use in the analysis. If it's absent, then SAS uses all numeric variables.

Summarizing your Data using PROC MEANS

Example:

```
proc sort data=Learn.Blood out=Blood;
   by Gender;
run;
title "Adding a BY Statement to PROC MEANS";
proc means data=Blood n nmiss mean median
                        min max maxdec=1;
   by Gender;
   var RBC WBC;
run;
```

Adding a BY Statement to PROC MEANS

The MEANS Procedure

Gender=Female

Variable	N	N Miss	Mean	Median	Minimum	Maximum
RBC	409	31	5.5	5.6	1.7	8.8
WBC	403	37	7112.4	7150.0	4620.0	10260.0

Gender=Male

Variable	N	N Miss	Mean	Median	Minimum	Maximum
RBC	507	53	5.5	5.5	2.3	8.4
WBC	505	55	6987.5	6930.0	4070.0	10550.0

Writing Summary Statistics to a SAS data set

OUTPUT statement:

OUTPUT OUT = data-set output – statistic – list;

Where:

- data-set is the name of the SAS data set which will contain the results, it could be either temporary or permanent.
- Output – statistic – list defines which statistics you want and the associated variable names.

Writing Summary Statistics to a SAS data set

Multiple OUTPUT statements:

OUTPUT OUT = data-set

statistics (variable – list) = name-list;

Where:

Variable – list : defines which of the variables int the VAR list, you want to output

Writing Summary Statistics to a SAS data set

Examples:

```
proc means data=Learn.Blood noprint;
    var RBC WBC;
    output out      = Many_Stats
           mean     = M_RBC M_WBC
           n        = N_RBC N_WBC
           nmiss    = Miss_RBC Miss_WBC
           median   = Med_RBC Med_WBC;
run;
```

	TYPE	_FREQ_	RBC_Mean	WBC_Mean	RBC_N	WBC_N	RBC_NMiss	WBC_NMiss	RBC_Median	WBC_Median
1	0	1000	5.4835262009	7042.9735683	916	908	84	92	5.52	7040

Counting Data with PROC FREQ

PROC FREQ most useful to:

- create tables showing the distribution of categorical data values .
- Reveal irregularities in the data

One – Way frequencies → counts for one variable

Tables variable – name;

Two – Way frequencies → counts for two or more variables

Counting Data with PROC FREQ

PROC FREQ options:

LIST	Prints cross tabulations in list format rather than grid
MISSPRINT	Includes missing values in frequencies but not in percentages
MISSING	Includes missing values in frequencies and in percentages
NOCOL	Suppresses printing of column percentages in cross-tabulations
NOPERCENT	Suppresses printing of percentages
NOROW	Suppresses printing of row percentages in cross-tabulations
OUT= data - set	Writes a data set containing frequencies

Counting Data with PROC FREQ

To produce cross tabulation list the variables separated by an asterisk:

*Tables var1 * var2*

Counting Data with PROC FREQ

Example:

```
title "A Two-way Table of Gender by Blood Type";
proc freq data=Learn.Blood;
    tables Gender * BloodType;
run;
```

Counting Data with PROC FREQ

A Two-way Table of Gender by Blood Type

The FREQ Procedure

Frequency Percent Row Pct Col Pct	Table of Gender by BloodType				
	BloodType(Blood Type)				
Gender(Gender)	A	AB	B	O	Total
Female	178 17.80 40.45 43.20	20 2.00 4.55 45.45	34 3.40 7.73 35.42	208 20.80 47.27 46.43	440 44.00
Male	234 23.40 41.79 56.80	24 2.40 4.29 54.55	62 6.20 11.07 64.58	240 24.00 42.86 53.57	560 56.00
Total	412 41.20	44 4.40	96 9.60	448 44.80	1000 100.00

Producing Tabular Report : PROC TABULTATE

PROC TABUALTE general form is as:

```
PROC TABULATE;
    CLASS  classification – variable  - list;
    TABLE  page – dimension, row – dimension,  column – dimension;
```

Where:

- CLASS statement tells SAS which variable contain categorical data to be used for dividing observations into groups.
- TABLE statement defines one table, but it’s possible to have multiple TABLE statements.

Producing Tabular Report : PROC TABULTATE

- Any variable listed in a TABLE statement must be listed in a CLASS statement or a VAR statement.
- If a variable is listed in a CLASS statement, then by default PROC TABULATE produces simple counts of the number of observations in each category of that variable

Producing Tabular Report : PROC TABULTATE

DIMENSIONS:

by default if:

One dimension specified → column dimension

Two dimensions specified → row and column dimensions

Three dimensions specified → page, row and column dimensions

Tip: in a dimension statement start by the column , debug, then add row, in front of column dimension, debug, then add page dimensions in front of row dimension.

Producing Tabular Report : PROC TABULTATE

```
title "Demonstrating Table Dimensions";
proc tabulate data=Learn.Blood format=6.;
    class Gender BloodType;
    table Gender,
          BloodType;
run;
```

Demonstrating Table Dimensions

	Blood Type			
	A	AB	B	O
	N	N	N	N
Gender				
Female	178	20	34	208
Male	234	24	62	240

Producing Tabular Report : PROC TABULTATE

MISSING data:

- By default missing observation values are excluded
- To keep missing values, then use MISSING option to PROC statement:

PROC TABULATE MISSING;

Producing Tabular Report : PROC TABULTATE

Examples:

```
title "The Effect of Missing Values on CLASS variables";
proc tabulate data=Learn.Missing format=4. missing;
   class A B;
   table A ALL,B ALL;
run;
```

The Effect of Missing Values on CLASS variables

	B				All
		X	Y	Z	
	N	N	N	N	N
A					
X	1	1	2	.	4
Y	.	.	.	1	1
Z	.	.	.	1	1
All	1	1	2	2	6

Statistics in PROC TABULATE Output

PROC TABULATE;

VAR analysis - variable – list;

CLASS classification – variable – list;

TABLE page-dimension, row – dimension, column – dimension;

On top of variable names, each dimension can contain **keywords**:

SUM	sum	MAX	Highest value	PCTN	Percentage of observations for that group
MEAN	average	MIN	Lowest value	PCTSUM	Percentage of total represented by that group
MEDIAN	median	N	Number of non-missing values	STDDEV	Standard deviation
MODE	mode	NMISS	Number of missing values	ALL	Adds a row, column or page showing the total

Statistics in PROC TABULATE Output

- To concatenate variables or keywords, list them separated by a space:

TABLE LOCOMOTION TYPE ALL;

- To cross variables or keywords , separate them with an asterisk (*):

*TABLE MEAN * Price ;*

- To group variables or keywords, enclose them in parenthesis

*TABLE PCTN *(LOCOMOTION TYPE);*

Statistics in PROC TABULATE Output

Example:

```
title "Computing Percentages on a Numerical Value";
proc tabulate data=Learn.Sales;
   class Region;
   var TotalSales;
   table (Region ALL),
         TotalSales*(n*f=6. sum*f=dollar8.
                     pctsum*f=pctfmt7.);
           keylabel ALL   = 'All Regions'
           n         = 'Number of Sales'
           sum       = 'Sum'
           pctsum    = 'Percent';
   label TotalSales = 'Total Sales';
run;
```

Statistics in PROC TABULATE Output

Computing Percentages on a Numerical Value

	Total Sales		
	Number of Sales	Sum	Percent
Region			
East	4	$41,593	44.8%
North	5	$36,825	39.7%
South	4	$12,003	12.9%
West	2	$2,290	2.4%
All Regions	15	$92,710	100.0%

Enhancing the Appearance of PROC TABULATE Output:

Three simple options can enhance the output of PROC TABULATE:

1. FORMAT = option
    - to be used in the PROC statement
    - It changes the format of all the data cells in the table

PROC TABULATE FORMAT = Comma10.0;

Enhancing the Appearance of PROC TABULATE Output:

2. BOX = option
    - To be used in the TABLE statements
    - It allows to write a brief phrase in the upper left corner box of every TABULATE report

*TABLE Region, MEAN*sales / BOX = 'Mean sales by Region';*

Enhancing the Appearance of PROC TABULATE Output:

3. MISSTEXT = option:

    - To be used in the TABLE statement.
    - It specifies a value for SAS to print in empty data cells

*TABLE region, MEAN*sales / MISSTEXT = 'No Sales';*

So, in the table, where a cell is empty because of a missing value, it will be written '*No Sales*'.

Enhancing the Appearance of PROC TABULATE Output:

Example:

```
title "Demonstrating the MISSTEXT TABLES Option";
proc tabulate data=Learn.Missing format=7. missing;
    class A B;
    table A ALL,B ALL / misstext='No Data';
run;
```

Example:

Demonstrating the MISSTEXT TABLES Option

	B				All
		X	Y	Z	
	N	N	N	N	N
A					
X	1	1	2	No Data	4
Y	No Data	No Data	No Data	1	1
Z	No Data	No Data	No Data	1	1
All	1	1	2	2	6

Changing Headers in PROC TABULATE Output

Two types of headers:

1. **CLASS variable values**: To change headers in a CLASS statement use a FORMAT procedure to create a user-defined format, then assign the format to the variable in FORMAT statement.

Changing Headers in PROC TABULATE Output

2. VARIABLE names and keywords: to change headers which are the names of variables or keywords:

*TABLE REGION = ' ' , MEAN = ' '*Sales = 'Mean Sales by Region'*

;

- *REGION and MEAN headers are removed because there's a blank in between the quotes.*
- *The header of Sales is changed to 'Mean Sales by Region'*

Multiple Formats for Data Cells in Proc Tabulate Output

To specify more than one format in a table, it's possible to do that by putting the FORMAT = option in the TABLE statement.

To apply a format to an individual variable , cross it with the variable name:

*Variable- name * FORMAT = format.d*

Then it can be inserted in the TABLE statement, as in this example:

*TABLE Region, MEAN * (SALES*FORMAT = COMMA8.0*

*Profit*FORMAT=DOLLAR1010.2);*