PSTAT 130

SAS BASE PROGRAMMING

- Lecture 8 -

Dr. Julie Swenson

Objectives

- Summarizing Data
  - PROC MEANS
    - Select variables
    - Specify keywords
    - Specify groups
  - PROC FREQ
    - Select tables
    - Cross-tabular tables
    - Categorize values
  - PROC TABULATE
    - Table construction
    - Class vs. analysis variables
    - Statistics

Summarize Data

- When summarizing a data set, we are often interested in the following characteristics:
  - The Center
    - Mean, Median, Mode
  - The Spread
    - Standard Deviation, Range
  - The Shape
    - Frequency Distribution, Outliers

Procedures to Summarize Data

- PROC MEANS
  - Calculate and display simple summary statistics
- PROC FREQ
  - Calculate and display frequency counts
- PROC TABULATE
  - Calculate and display multi-dimensional tables with summary statistics
- PROC REPORT (*next lecture*)
  - Create list and summary reports

The MEANS Procedure

- Calculates common summary statistics
- Summarizes numeric variables
- BY and CLASS statements can be used to create summaries for subgroups
- Can create an output data set of summary statistics

The MEANS Procedure

- General form of a simple MEANS procedure

```
PROC MEANS DATA=SAS-data-set;
RUN;
```

- Example

```
proc means data=data1.admit;
run;
```

PROC MEANS Default Output

The MEANS Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
Age	21	38.0476190	10.3124982	22.0000000	60.0000000
Date	21	14.5238095	9.1630729	1.0000000	31.0000000
Height	21	68.2380952	4.3116674	61.0000000	76.0000000
Weight	21	156.5238095	22.6398301	118.0000000	193.0000000
Fee	21	127.9500000	24.1524222	85.2000000	149.7500000

PROC MEANS Default Output

- By default, PROC MEANS
  - Analyzes every numeric variable in the SAS data set
  - Displays five statistics
    - N
    - MEAN
    - STD
    - MIN
    - MAX
  - Excludes missing values before calculating statistics

Statements and Options

- **VAR** <variable list>
  - selects [numeric] variables to be summarized
- Statistical Keywords
  - option(s) in the PROC MEANS statement
- **BY** <variable list>
  - creates separate summaries for each BY group
- **CLASS** <variable list>
  - creates separate summaries for each CLASS group
- **OUTPUT out=**SAS data set
  - creates an output data set containing summary statistics

Select Variables

- Use the VAR statement (as seen in the PRINT procedure) to select specific variables for analysis

- Example

```
proc means data=data1.admit;
   var age height weight;
run;
```

Specify Statistical Keywords

- List keywords for statistics as *options* to the PROC MEANS statement

```
proc means data=data1.admit n mean stddev;
   var age height weight;
run;
```

Common Statistical Keywords

- MIN
- MAX
- RANGE
- MEAN
- MEDIAN
- STDDEV
- SUM
- N
- NMISS: the number of observations missing a value for each variable
- Confidence intervals, percentiles, and probability functions can also be requested – see the SAS Help for statistic keywords in the MEANS procedure

Analyze Subgroups: BY Statement

- Use the BY statement to request summaries for subgroups

- Example

```
proc means data=work.admit n mean stddev;
   var age height weight;
   by actlevel;
run;
```

- Note: Data MUST be sorted on the BY variable(s) first.

BY Statement Partial Output

The MEANS Procedure

ActLevel=HIGH

Variable	N	Mean	Std Dev
Age	7	34.2857143	7.5213980
Height	7	70.1428571	4.2201332
Weight	7	163.5714286	21.1412483

ActLevel=LOW

Variable	N	Mean	Std Dev
Age	7	39.2857143	14.0441481
Height	7	66.4285714	5.0284903
Weight	7	150.7142857	26.1897835

Analyze Subgroups: CLASS Statement

- Use the CLASS statement to request summaries for subgroups

- Example

```
proc means data=data1.admit n mean stddev;
   title 'PROC MEANS With a CLASS Variable';
   var age height weight;
   class actlevel;
run;
```

- Note: Data does <u>NOT</u> need to be sorted on the class variable.

CLASS Statement Output

PROC MEANS With a CLASS Variable

The MEANS Procedure

ActLevel	N Obs	Variable	N	Mean	Std Dev
HIGH	7	Age	7	34.2857143	7.5213980
		Height	7	70.1428571	4.2201332
		Weight	7	163.5714286	21.1412483
LOW	7	Age	7	39.2857143	14.0441481
		Height	7	66.4285714	5.0284903
		Weight	7	150.7142857	26.1897835
MOD	7	Age	7	40.5714286	8.6575043
		Height	7	68.1428571	3.2877840
		Weight	7	155.2857143	21.8305160

Save the Output

- Use the `Output out=` statement to save the results of your MEANS procedure to a **SAS data set**

- Example

```
proc means data=work.admit;
   var age height weight;
   by actlevel;
   output out=meansout;
run;
```

Output SAS Data Set

Obs	ActLevel	_TYPE_	_FREQ_	_STAT_	Age	Height	Weight
1	HIGH	0	7	N	7.0000	7.0000	7.000
2	HIGH	0	7	MIN	25.0000	66.0000	140.000
3	HIGH	0	7	MAX	44.0000	76.0000	193.000
4	HIGH	0	7	MEAN	34.2857	70.1429	163.571
5	HIGH	0	7	STD	7.5214	4.2201	21.141
6	LOW	0	7	N	7.0000	7.0000	7.000
7	LOW	0	7	MIN	22.0000	61.0000	118.000
8	LOW	0	7	MAX	60.0000	73.0000	191.000
9	LOW	0	7	MEAN	39.2857	66.4286	150.714
10	LOW	0	7	STD	14.0441	5.0285	26.190
11	MOD	0	7	N	7.0000	7.0000	7.000
12	MOD	0	7	MIN	30.0000	63.0000	123.000

Limit Number of Decimals

- Use the `MAXDEC=` option in the PROC MEANS statement to limit the number of decimal places in the summary statistics

```
proc means data=data1.admit n mean stddev maxdec=2;
   var age height weight;
run;
```

Displays 2 decimal places

The MEANS Procedure

Variable	N	Mean	Std Dev
Age	21	38.05	10.31
Height	21	68.24	4.31
Weight	21	156.52	22.64

The FREQ Procedure

Distribution of Job Code Values

The FREQ Procedure

JobCode	Frequency	Percent	Cumulative Frequency	Cumulative Percent
FLTAT1	14	20.29	14	20.29
FLTAT2	18	26.09	32	46.38
FLTAT3	12	17.39	44	63.77
PILOT1	8	11.59	52	75.36
PILOT2	9	13.04	61	88.41
PILOT3	8	11.59	69	100.00

Create a Frequency Report

- General form of a simple PROC FREQ step

```
PROC FREQ DATA=SAS-data-set;
RUN;
```

- Example

```
proc freq data=data1.admit;
run;
```

PROC FREQ Default Output

The FREQ Procedure

ID	Frequency	Percent	Cumulative Frequency	Cumulative Percent
2458	1	4.76	1	4.76
2462	1	4.76	2	9.52
2501	1	4.76	3	14.29
2523	1	4.76	4	19.05
2539	1	4.76	5	23.81
2544	1	4.76	6	28.57
2552	1	4.76	7	33.33
2555	1	4.76	8	38.10
2563	1	4.76	9	42.86
2568	1	4.76	10	47.62
2571	1	4.76	11	52.38

PROC FREQ Default Output

- By default, PROC FREQ
  - Analyzes every variable in the SAS data set
  - Displays each distinct data value
  - Displays the number of observations in which each data value appears (and the corresponding relative and cumulative percentages)
  - Indicates for each variable how many observations have missing values

PROC FREQ: TABLES Statement

- Use the `TABLES` statement to select variables and to specify the type of frequency report.

- General form of a PROC FREQ step with a `TABLES` statement

```
PROC FREQ DATA=SAS-data-set;
   TABLES variable-list / options;
RUN;
```

Create a Frequency Report

```
proc freq data=data1.crew;
  tables JobCode;
  title 'Distribution of Job Code Values';
run;
```

Displays a frequency table for the variable, JobCode

JobCode	Frequency	Percent	Cumulative Frequency	Cumulative Percent
FLTAT1	14	20.29	14	20.29
FLTAT2	18	26.09	32	46.38
FLTAT3	12	17.39	44	63.77
PILOT1	8	11.59	52	75.36
PILOT2	9	13.04	61	88.41
PILOT3	8	11.59	69	100.00

The table lists each value of JobCode, and its frequency

Analyze Categories of Values

- What if we would like to analyze categories of values?
- For example, instead of analyzing JobCode values individually, we would like to analyze two categories of JobCode: Flight Attendant and Pilot

The FREQ Procedure

JobCode	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Flight Attendant	44	63.77	44	63.77
Pilot	25	36.23	69	100.00

Analyze Categories of Values

- Use the `FORMAT` statement to analyze the frequency of observations within *user-defined* categories

- Example

```
proc format;
   value $codefmt
      'FLTAT1'-'FLTAT3'='Flight Attendant'
      'PILOT1'-'PILOT3'='Pilot';
run;
proc freq data = data1.crew;
   format JobCode $codefmt.;
   tables JobCode;
run;
```

Crosstabular Frequency Reports

- General form of the two-way tables, known as crosstabs (row × column)

```
PROC FREQ DATA=SAS-data-set;
   TABLES variable1 * variable2;
RUN;
```

- Two-way tables categorize observations on the combination of two sets of categories (i.e. Male Pilots, Female Pilots, Male Flight Attendants, and Female Flight Attendants)

Crosstabular Example

```
proc format;
   value $codefmt
      'FLTAT1'-'FLTAT3'='Flight Attendant'
      'PILOT1'-'PILOT3'='Pilot';
   value money
      low-<25000 ='Less than 25,000'
      25000-50000='25,000 to 50,000'
      50000<-high='More than 50,000';
run;
proc freq data=data1.crew;
   tables JobCode*Salary;
   format JobCode $codefmt. Salary money.;
   title 'Salary Distribution by Job Codes';
run;
```

Crosstabular Output

Salary Distribution by Job Codes

The FREQ Procedure

Frequency
Percent
Row Pct
Col Pct

Table of JobCode by Salary

JobCode	Salary: Less than 25,000	25,000 to 50,000	More than 50,000	Total
Flight Attendant	5 7.25 11.36 100.00	39 56.52 88.64 100.00	0 0.00 0.00 0.00	44 63.77
Pilot	0 0.00 0.00 0.00	0 0.00 0.00 0.00	25 36.23 100.00 100.00	25 36.23
Total	5 7.25	39 56.52	25 36.23	69 100.00

The TABULATE Procedure

- The report writing features of PROC TABULATE include
  - Control of table construction
  - Differentiating between classification variables and analysis variables
  - Specifying statistics
  - Formatting of values
  - Labelling of variables and statistics.

PROC TABULATE Syntax

- General form of a PROC TABULATE step

```
PROC TABULATE DATA=SAS-data-set <options>;
   CLASS class-variables;
   VAR analysis-variables;
   TABLE page-expression,
         row-expression,
         column-expression </ option(s)>;
RUN;
```

Specify Classification Variables

```
PROC TABULATE DATA=SAS-data-set <options>;
   CLASS class-variables;
   VAR analysis-variables;
   TABLE page-expression,
         row-expression,
         column-expression </ option(s)>;
RUN;
```

- Class variables are used to create subgroups on one or more dimensions

Specify Classification Variables

```
PROC TABULATE DATA=SAS-data-set <options>;
   CLASS class-variables;
   VAR analysis-variables;
   TABLE page-expression,
         row-expression,
         column-expression </ option(s)>;
RUN;
```

- Summary statistics (e.g., means) are calculated for the VAR variables

Specify Classification Variables

```
PROC TABULATE DATA=SAS-data-set <options>;
   CLASS class-variables;
   VAR analysis-variables;
   TABLE page-expression,
         row-expression,
         column-expression </ option(s)>;
RUN;
```

- The TABLE statement specifies the format of the table

Use of Class Variables Only

```
title 'Flight Attendant Counts by Location';
proc tabulate data=data1.fltat;
   class Location;
   table Location;
run;
```

Flight Attendant Counts by Location

Location		
CARY	FRANKFURT	LONDON
N	N	N
17.00	12.00	15.00

Obtain a Total

```
proc tabulate data=data1.fltat;
   class Location;
   table Location All;
run;
```

Blank Operator between Location and All concatenates information

Flight Attendant Counts by Location

Location			All
CARY	FRANKFURT	LONDON	
N	N	N	N
17.00	12.00	15.00	44.00

Two-Dimensional Tables

```
proc tabulate
data=data1.fltat;
   class Location JobCode;
   table JobCode, Location;
run;
```

Row Dimension

Comma operator moves to a new dimension

Column Dimension

Two-Dimensional Tables

Column Dimension

Row Dimension

	Location		
	CARY	FRANKFURT	LONDON
	N	N	N
JobCode			
FLTAT1	5	4	5
FLTAT2	7	5	6
FLTAT3	5	3	4

Subset the Data

```
title 'Counts for Cary and Frankfurt';
proc tabulate data=data1.fltat;
   where Location in ('CARY', 'FRANKFURT');
   class Location JobCode;
   table JobCode, Location;
run;
```

Subset the Data

Flight Attendant Counts by Location
by JobCode

	Location	
	CARY	FRANKFURT
	N	N
JobCode		
FLTAT1	5.00	4.00
FLTAT2	7.00	5.00
FLTAT3	5.00	3.00

Two-Dimensional Tables

```
proc tabulate data=data1.fltat;
      where Location in ('CARY', 'FRANKFURT');
      class Location JobCode;
      table JobCode all, Location all;
run;
```

Row Dimension

Column Dimension

Two-Dimensional Tables

All in Column Dimension

All in Row Dimension

Total Salary for Cary and Frankfurt

	Location		
	CARY	FRANKFURT	All
	N	N	N
JobCode			
FLTAT1	5.00	4.00	9.00
FLTAT2	7.00	5.00	12.00
FLTAT3	5.00	3.00	8.00
All	17.00	12.00	29.00

Use of Analysis Variables

```
title 'Total Salary for Cary and Frankfurt';
proc tabulate data=data1.fltat;
   where Location in ('CARY', 'FRANKFURT');
   class Location JobCode;
   var Salary;
   table JobCode, Location*Salary;
run;
```

Use of Analysis Variables

Total Salary for Cary and Frankfurt

	Location	
	CARY	FRANKFURT
	Salary	Salary
	Sum	Sum
JobCode		
FLTAT1	131000.00	100000.00
FLTAT2	245000.00	181000.00
FLTAT3	217000.00	134000.00

Salary within each Location

Format the Statistic

```
proc tabulate data=data1.fltat format=dollar12.;
   where Location in ('CARY', 'FRANKFURT');
   class Location JobCode;
   var Salary;
   table JobCode, Location*Salary;
run;
```

Format the Statistic

Total Salary for Cary and Frankfurt

	Location	
	CARY	FRANKFURT
	Salary	Salary
	Sum	Sum
JobCode		
FLTAT1	$131,000	$100,000
FLTAT2	$245,000	$181,000
FLTAT3	$217,000	$134,000

FORMAT= option changes default format for ALL cells

Specify a Statistic

```
title 'Average Salary for Cary and Frankfurt';
proc tabulate data=data1.fltat format=dollar12.;
   where Location in ('CARY', 'FRANKFURT');
   class Location JobCode;
   var Salary;
   table JobCode, Location*Salary*mean;
run;
```

Specify a Statistic

Average Salary for Cary and Frankfurt

	Location	
	CARY	FRANKFURT
	Salary	Salary
	Mean	Mean
JobCode		
FLTAT1	$26,200	$25,000
FLTAT2	$35,000	$36,200
FLTAT3	$43,400	$44,667

MEAN statistic

ALL with Analysis Variable

- General form for generating overall information when using an analysis variable

ALL*analysis-variable*statistic keyword

- Example

```
proc tabulate data=data1.fltat format=dollar12.;
   where Location in ('CARY', 'FRANKFURT');
   class Location JobCode;
   var Salary;
   table JobCode all,
         Location*Salary*mean all*Salary*mean;
run;
```

```
proc tabulate data=data1.fltat format=dollar12.;
   where Location in ('CARY', 'FRANKFURT');
   class Location JobCode;
   var Salary;
   table JobCode all,
         Location*Salary*mean all*Salary*mean;
run;
```

Column Dimension

All in Column Dimension

Salary within each Location

Row Dimension

FORMAT= option

	Location		
	CARY	FRANKFURT	All
	Salary	Salary	Salary
	Mean	Mean	Mean
JobCode			
FLTAT1	$26,200	$25,000	$25,667
FLTAT2	$35,000	$36,200	$35,500
FLTAT3	$43,400	$44,667	$43,875
All	$34,882	$34,583	$34,759

All in Row Dimension

Class Exercise

- Use the **`heart`** data set in the data1 folder
  - Inspect the descriptor portion and data portion of the data set
  - Run PROC MEANS using default options
  - Now limit the statistics to two decimal places
  - Run PROC MEANS for the `Arterial` and `Cardiac` variables
  - Run PROC MEANS using `Sex` as a class variable
  - Run PROC FREQ using default options
  - Run PROC FREQ for the `Shock` and `Survive` variables
  - Run PROC FREQ to obtain a crosstabular table for `Shock` (row) by `Survive` (column)