Print

Summary of Lesson 11: Creating Summary Reports

This summary contains topic summaries, syntax, and sample programs.

Topic Summaries

To go to the movie where you learned a task or concept, select a link.

Using PROC FREQ to Create Summary Reports

You can use PROC FREQ to produce frequency tables that report the distribution of any or all variable values in a SAS data set. You use the TABLES statement to specify the frequency tables to produce. You can include a WHERE statement in the PROC FREQ step to subset the observations. SAS has TABLES statement options that you can use to suppress the default statistics.

```
PROC FREQ DATA=SAS-data-set <option(s)>;
     TABLES variable(s) </option(s)>;
     <additional SAS statements>
RUN;
```

Frequency distributions work best with variables whose values are categorical and best summarized by counts instead of averages. Variables that have continuous numeric values, such as dollar amounts and dates, or many discrete values, can result in a lengthy and meaningless frequency table. To create a useful frequency report for these variables, you can apply a SAS or user-defined format to group the values into categories.

You can list multiple variables in a TABLES statement, separated by spaces. This creates a one-way frequency table for each variable. You can request a separate analysis for each group by including a BY statement. You can request a two-way frequency table by separating the variables with an asterisk instead of a space. The resulting crosstabulation table displays statistics for each distinct combination of values of the selected variables. You can use TABLES statement options to suppress statistics, change the table format, and format the displayed values.

Using PROC FREQ for Data Validation

The FREQ procedure can also be used to validate a data set. A one-way frequency table, which displays all discrete values for a variable and reports on missing values, easily identifies the existence of invalid or missing values. You can use the ORDER=FREQ and NLEVELS options to identify duplicate values. After you've identified invalid values, you can use PROC PRINT to display the corresponding observations.

Using the MEANS and UNIVARIATE Procedures

You can use PROC MEANS to produce summary reports with descriptive statistics. By default, it reports the number of nonmissing values, the mean, the standard deviation, the minimum, and the maximum value of every numeric variable in a data set. You can use the VAR statement to specify the numeric variables to analyze, and add a CLASS statement to request statistics for groups of observations. The variables listed in the CLASS statement are called classification variables, or class variables, and each combination of class variable values is called a class level.

```
PROC MEANS DATA=SAS-data-set <statistic(s)>;
     VAR analysis-variable(s);
     CLASS classification-variable(s);
RUN;
```

When you use the CLASS statement, the output includes N Obs, which reports the number of observations with each unique combination of class variables. You can request specific statistics by listing them as options in the PROC MEANS statement. Other options are available to control the output.

You can also use PROC MEANS to validate a data set. The MIN, MAX, and NMISS statistics can be used to validate numeric data when you know the range of valid values. PROC UNIVARIATE can be more useful because it displays the extreme observations, or outliers. By default, it displays the five highest and five lowest values of the analysis variable, and the number of the observation with each extreme value. You can use the NEXTROBS= option to display a different number of extreme observations.

PROC UNIVARIATE DATA=*SAS-data-set***;**
  VAR *variable(s)***;**
RUN;

Using the Output Delivery System

You can use the SAS Output Delivery System to create different output formats by directing output to various ODS destinations. For each type of formatted output that you want to create, you use an ODS statement to open that destination, submit one or more procedures that generate output, and then close the destination. A file is created for each open destination.

ODS *destination* **FILE="***filename***"** <options>;
  <*SAS code to generate the report***>**
ODS *destination* **CLOSE;**

Sample Programs

Creating a One-Way Frequency Report

```
proc freq data=orion.sales;
   tables Gender;
   where Country='AU';
run;
```

Using Formats in PROC FREQ

```
proc format;
   value Tiers low-25000='Tier1'
               25000<-50000='Tier2'
               50000<-100000='Tier3'
               100000<-high='Tier4';
run;

proc freq data=orion.sales;
   tables Salary;
   format Salary Tiers.;
run;
```

Listing Multiple Variables on a TABLES Statement

```
proc freq data=orion.sales;
   tables Gender Country;
run;

proc sort data=orion.sales out=sorted;
   by Country;
run;

proc freq data=sorted;
   tables Gender;
   by Country;
run;
```

Creating a Crosstabulation Table

```
proc freq data=orion.sales;
   tables Gender*Country;
run;
```

Examining Your Data

```
proc print data=orion.nonsales2 (obs=20);
run;
```

Using PROC FREQ Options to Validate Your Data

```
proc freq data=orion.nonsales2 order=freq;
   tables Employee_ID/nocum nopercent;
run;

proc freq data=orion.nonsales2 nlevels;
   tables Gender Country Employee_ID/nocum nopercent;
run;

proc freq data=orion.nonsales2 nlevels;
   tables Gender Country Employee_ID/nocum nopercent noprint;
run;
```

Using PROC PRINT to Validate Your Data

```
proc print data=orion.nonsales2;
   where Gender not in ('F','M') or
         Country not in ('AU','US') or
         Job_Title is null or
         Salary not between 24000 and 500000 or
         Employee_ID is missing or
         Employee_ID=120108;
run;
```

Creating a Summary Report with PROC MEANS

```
proc means data=orion.sales;
   var Salary;
run;
```

Creating a PROC MEANS Report with Grouped Data

```
proc means data=orion.sales;
   var Salary;
   class Gender Country;
run;
```

Requesting Specific Statistics in PROC MEANS

```
proc means data=orion.sales n mean;
   var Salary;
run;

proc means data=orion.sales min max sum;
   var Salary;
   class Gender Country;
run;
```

Validating Data Using PROC MEANS

```
proc means data=orion.nonsales2 n nmiss min max;
   var Salary;
run;
```

Validating Data Using PROC UNIVARIATE

```
proc univariate data=orion.nonsales2;
   var Salary;
run;

proc univariate data=orion.nonsales2 nextrobs=3;
   var Salary;
run;

proc univariate data=orion.nonsales2 nextrobs=3;
   var Salary;
   id Employee_ID;
run;
```

Using the SAS Output Delivery System

```
/*Use a filepath to a location where you have Write access.*/
ods pdf file="c:/output/salaries.pdf";

proc means data=orion.sales min max sum;
   var Salary;
   class Gender Country;
run;

ods pdf close;

ods csv file="c:/output/salarysummary.csv";

proc means data=orion.sales min max sum;
   var Salary;
   class Gender Country;
run;

ods csv close;
```

SAS Programming 1: Essentials

Close