STA 311 Statistical Computing and Data Management

Instructor: Cheng Peng, Ph.D.
Department of Mathematics
West Chester University
West Chester, PA 19383

Office: 25 University Avenue, RM 111
Phone: 610-435-2369
Email: cheng.peng@maine.edu

5. Descriptive Statistics with SAS

Computing Univariate Statistics

- Review of Descriptive Statistics
- PROC MEANS
- PROC FREQ
- PROC UNIVARIATE

Computing Univariate Statistics

What we have learned from Introduction to Statistics

- Basic Concepts of Statistics
- Summarizing Data
- Probability
- Confidence Intervals
- Hypothesis Testing - Means & Proportions

Basic Statistics

Continuous Variables

Measures of Central Tendency

- Median
- Mean

Graphs

- Histograms
- Scatter Plots

Measures of Dispersion

- Interquartile Range (IQR)
- Variance
- Standard Deviation (SD)

Two Continuous Variables

- Correlation
- Regression

Categorical Variables

Descriptive Statistics

- Counts
- Proportions

Graphs

- Bar charts
- Pie charts

Basic Statistics SAS PROCs

SAS procedures MEANS, UNIVARITE and FREQ produce much more than what are covered in Introductory Statistics

1. Continuous variable

mean, variance, standard deviation, median, mode, upper quartile, lower quartile, percentiles, skewness, kurtosis, number of missing observations.

2. Classification variable

Frequency in each category, relative frequency.

Several Options of Descriptive Statistics

N	Number of observations with non-missing values
Sum Wgts	Sum of weights = N unless a weighted analysis is requested
Mean	Arithmetic average
Sum	Total of all values
Std Dev	Sample standard deviation
Variance	Sample variance
CV	Coefficient of variation = (standard deviation divided by mean) times 100
Std Mean	Standard error of the mean = standard deviation/(square root of N)

Sum of Squares

USS Uncorrected sum of squares =
sum of the squared values of the observations

CSS Corrected sum of squares =
sum of the squares of the differences between the observations and their mean.

Num ≠ 0 the number of observations not equal to zero.

Num > 0 the number of observations greater than zero.

$$USS = \sum_{i=1}^{N} x_i^2$$

$$CSS = \sum_{i=1}^{N} \left(x_i - \bar{x}\right)^2$$

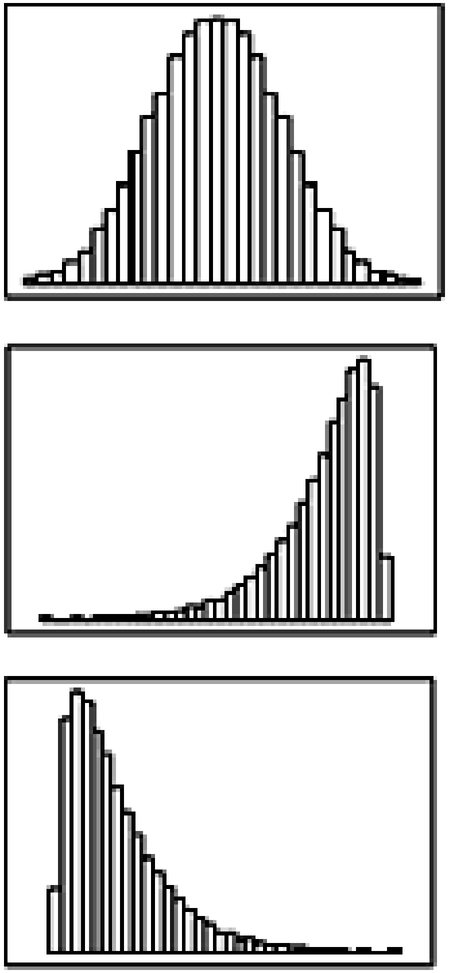
Skewness

The **skewness** is a measure of the tendency of the deviations from the mean to be larger in one direction than in the other. The sample **skewness** is calculated as:

$$\mu_3 = \frac{n}{(n-1)(n-2)} \sum_{i=1}^{n} \left(x_i - \bar{x} \right)^3$$

Negative values of μ_3 indicate **skewness** to the right.
Positive values of μ_3 indicate **skewness** to the left.
The normal and t distributions have zero **skewness**.

Skewness



Symmetric
Bell shaped

Skewed to
the Left

Skewed to
the Right

Test Statistics

T: Mean=0 t-test statistic for testing whether the population average is zero.

This is often used for paired t-tests in which one examines whether the difference between two measurements on the same experimental unit is different from zero.

Pr > |T| P-value for a two-tailed t-test of whether the population average is zero.

Low P-values provide evidence that the population mean is not zero.

PROC MEANS Syntax

The MEANS procedure provides descriptive statistics such as the mean, minimum, and maximum provide useful information about numeric data.

Procedure Syntax

```
PROC MEANS <DATA=SAS-data-set>
           <statistic-keyword(s)> <option(s)>;
RUN;
```

Where

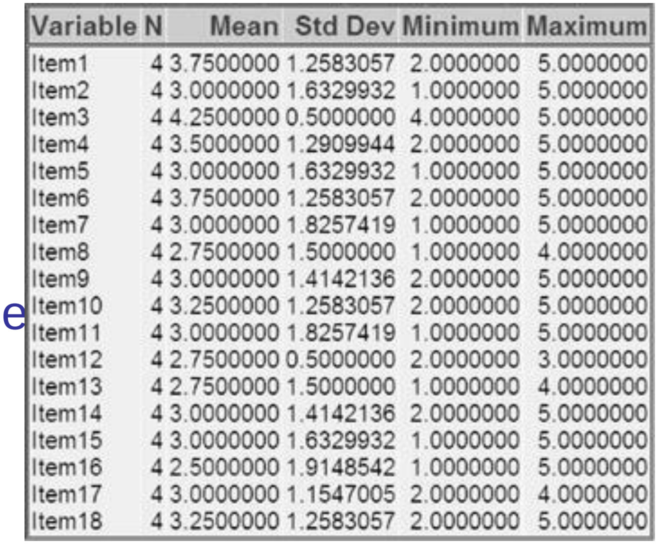
SAS-data-set is the name of the data set to be used

statistic-keyword(s) specify the statistics to compute

option(s) control the content, analysis, and appearance

Example

```
proc means data=perm.survey;
run;
```



Variable	N	Mean	Std Dev	Minimum	Maximum
Item1	4	3.7500000	1.2583057	2.0000000	5.0000000
Item2	4	3.0000000	1.6329932	1.0000000	5.0000000
Item3	4	4.2500000	0.5000000	4.0000000	5.0000000
Item4	4	3.5000000	1.2909944	2.0000000	5.0000000
Item5	4	3.0000000	1.6329932	1.0000000	5.0000000
Item6	4	3.7500000	1.2583057	2.0000000	5.0000000
Item7	4	3.0000000	1.8257419	1.0000000	5.0000000
Item8	4	2.7500000	1.5000000	1.0000000	4.0000000
Item9	4	3.0000000	1.4142136	2.0000000	5.0000000
Item10	4	3.2500000	1.2583057	2.0000000	5.0000000
Item11	4	3.0000000	1.8257419	1.0000000	5.0000000
Item12	4	2.7500000	0.5000000	2.0000000	3.0000000
Item13	4	2.7500000	1.5000000	1.0000000	4.0000000
Item14	4	3.0000000	1.4142136	2.0000000	5.0000000
Item15	4	3.0000000	1.6329932	1.0000000	5.0000000
Item16	4	2.5000000	1.9148542	1.0000000	5.0000000
Item17	4	3.0000000	1.1547005	2.0000000	4.0000000
Item18	4	3.2500000	1.2583057	2.0000000	5.0000000

PROC MEANS Example

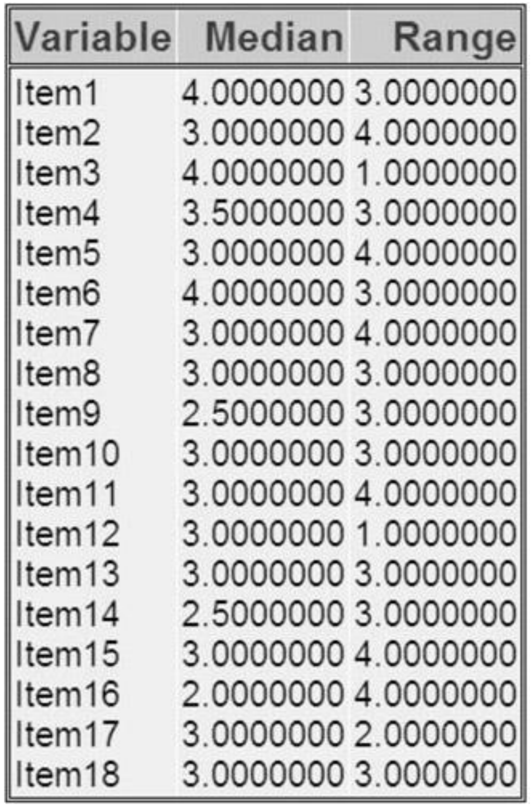
Selecting Statistics
Consider that you want to see the median and range of Perm. Survey numeric values, add the MEDIAN and RANGE keywords as options.

Example

```
proc means data=perm.survey median range;
run;
```

The following keywords can be used with PROC MEANS to compute statistics:

Keyword	Description
MAX	Maximum value
MEAN	Average
MODE	Value that occurs most frequently
MIN	Minimum value
VAR	Variance



Variable	Median	Range
Item1	4.0000000	3.0000000
Item2	3.0000000	4.0000000
Item3	4.0000000	1.0000000
Item4	3.5000000	3.0000000
Item5	3.0000000	4.0000000
Item6	4.0000000	3.0000000
Item7	3.0000000	4.0000000
Item8	3.0000000	3.0000000
Item9	2.5000000	3.0000000
Item10	3.0000000	3.0000000
Item11	3.0000000	4.0000000
Item12	3.0000000	1.0000000
Item13	3.0000000	3.0000000
Item14	2.5000000	3.0000000
Item15	3.0000000	4.0000000
Item16	2.0000000	4.0000000
Item17	3.0000000	2.0000000
Item18	3.0000000	3.0000000

PROC MEANS: CLASS Statement

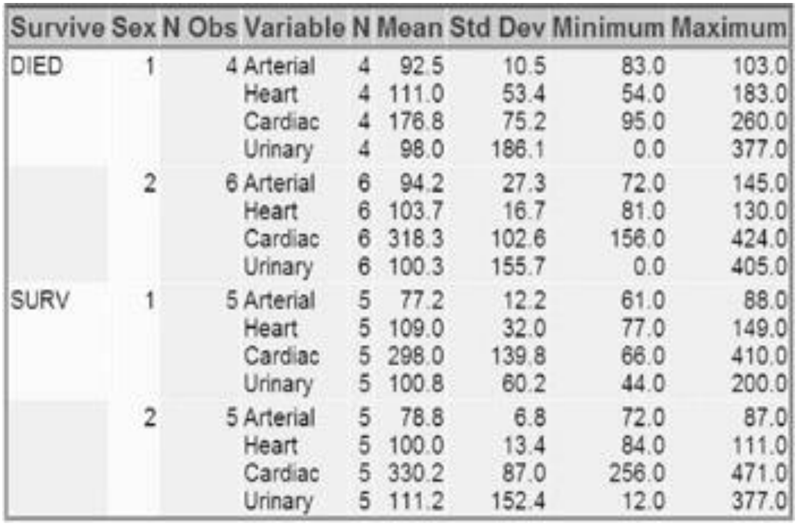
Group Processing Using the CLASS Statement

To produce separate analyses of grouped observations, add a CLASS statement to the MEANS procedure. General form, CLASS statement:

```
CLASS variable(s);
```

where *variable(s)* specifies category variables for group processing.

CLASS variables can be either character or numeric, but they should contain a limited number of discrete values that represent meaningful groupings.



Survive	Sex	N Obs	Variable	N	Mean	Std Dev	Minimum	Maximum
DIED	1	4	Arterial	4	92.5	10.5	83.0	103.0
			Heart	4	111.0	53.4	54.0	183.0
			Cardiac	4	176.8	75.2	95.0	260.0
			Urinary	4	98.0	186.1	0.0	377.0
	2	6	Arterial	6	94.2	27.3	72.0	145.0
			Heart	6	103.7	16.7	81.0	130.0
			Cardiac	6	318.3	102.6	156.0	424.0
			Urinary	6	100.3	155.7	0.0	405.0
SURV	1	5	Arterial	5	77.2	12.2	61.0	88.0
			Heart	5	109.0	32.0	77.0	149.0
			Cardiac	5	298.0	139.8	66.0	410.0
			Urinary	5	100.8	60.2	44.0	200.0
	2	5	Arterial	5	78.8	6.8	72.0	87.0
			Heart	5	100.0	13.4	84.0	111.0
			Cardiac	5	330.2	87.0	256.0	471.0
			Urinary	5	111.2	152.4	12.0	377.0

Example

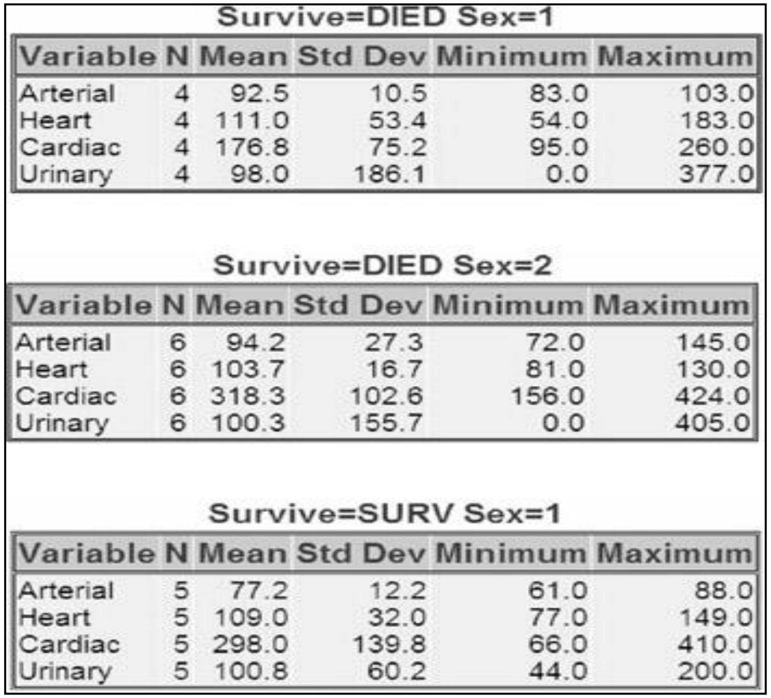
```
proc means data = clinic.heart maxdec=1;
   var arterial heart cardiac urinary;
   class survive sex;
run;
```

PROC MEANS: BY Statement

When using the BY statement, you must SORT the data by the variable to be used in the BY statement!

```
proc sort data=clinic.heart
   out=work.heartsort;
   by survive sex;
run;

proc means data= work.heartsort
             maxdec=1;
  var arterial heart cardiac
      urinary;
  by survive sex;
run;
```



Survive=DIED Sex=1

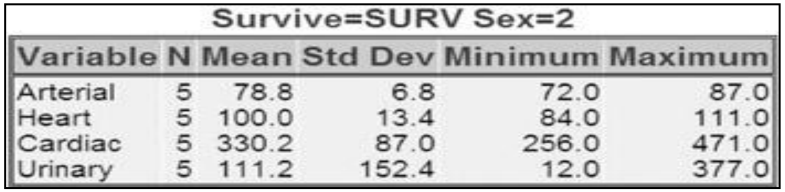
Variable	N	Mean	Std Dev	Minimum	Maximum
Arterial	4	92.5	10.5	83.0	103.0
Heart	4	111.0	53.4	54.0	183.0
Cardiac	4	176.8	75.2	95.0	260.0
Urinary	4	98.0	186.1	0.0	377.0

Survive=DIED Sex=2

Variable	N	Mean	Std Dev	Minimum	Maximum
Arterial	6	94.2	27.3	72.0	145.0
Heart	6	103.7	16.7	81.0	130.0
Cardiac	6	318.3	102.6	156.0	424.0
Urinary	6	100.3	155.7	0.0	405.0

Survive=SURV Sex=1

Variable	N	Mean	Std Dev	Minimum	Maximum
Arterial	5	77.2	12.2	61.0	88.0
Heart	5	109.0	32.0	77.0	149.0
Cardiac	5	298.0	139.8	66.0	410.0
Urinary	5	100.8	60.2	44.0	200.0



Survive=SURV Sex=2

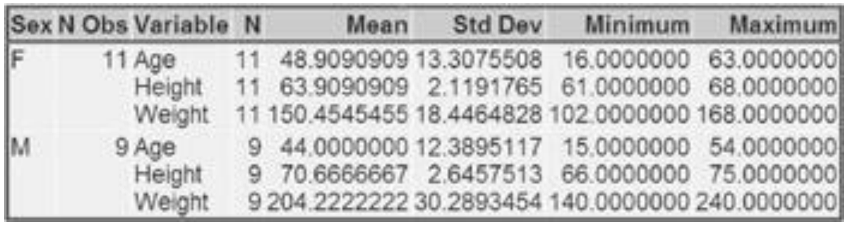
Variable	N	Mean	Std Dev	Minimum	Maximum
Arterial	5	78.8	6.8	72.0	87.0
Heart	5	100.0	13.4	84.0	111.0
Cardiac	5	330.2	87.0	256.0	471.0
Urinary	5	111.2	152.4	12.0	377.0

PROC MEANS: OUTPUT Statement

Specifying the STATISTIC= Option

You can specify which statistics to produce in the output data set. To do so, you must specify the statistic and then list all of the variables. The variables must be listed in the same order as in the VAR statement. You can specify more than one statistic in the OUTPUT statement.

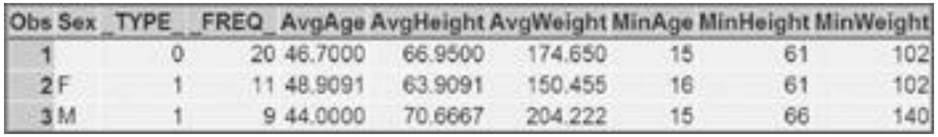
PROC MEANS in SAS LIST WINDOW



Sex	N Obs	Variable	N	Mean	Std Dev	Minimum	Maximum
F	11	Age	11	48.9090909	13.3075508	16.0000000	63.0000000
		Height	11	63.9090909	2.1191765	61.0000000	68.0000000
		Weight	11	150.4545455	18.4464828	102.0000000	168.0000000
M	9	Age	9	44.0000000	12.3895117	15.0000000	54.0000000
		Height	9	70.6666667	2.6457513	66.0000000	75.0000000
		Weight	9	204.2222222	30.2893454	140.0000000	240.0000000

```
proc means data=clinic.diabetes;
     class sex;
     var age height weight;
     output out=work.sum_gender
        mean=AvgAge AvgHeight AvgWeight
        min=MinAge MinHeight MinWeight;
run;
```

PROC MEANS OUTPUT TO SAS DATASET



Obs	Sex	_TYPE_	_FREQ_	AvgAge	AvgHeight	AvgWeight	MinAge	MinHeight	MinWeight
1		0	20	46.7000	66.9500	174.650	15	61	102
2	F	1	11	48.9091	63.9091	150.455	16	61	102
3	M	1	9	44.0000	70.6667	204.222	15	66	140

To see the contents of the output data set, submit the following PROC PRINT step.

PROC FREQ: Basics

The FREQ procedure is a descriptive procedure as well as a statistical procedure. It produces one-way and *n*-way frequency tables.

You can use the FREQ procedure to create cross-tabulation tables that summarize data for two or more categorical variables by showing the number of observations for each combination of variable values.

General form, basic FREQ procedure:

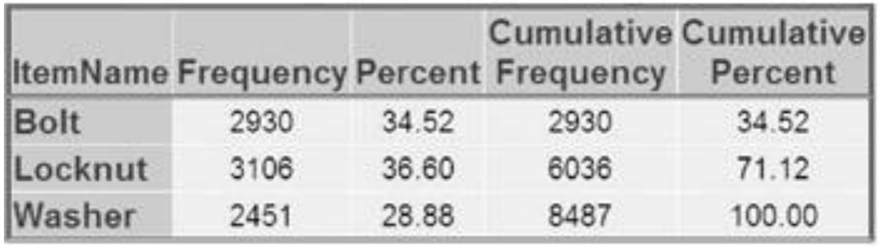
```
PROC FREQ <DATA=SAS-data-set>;
RUN;
```

By default, PROC FREQ creates a one-way table with the frequency, percent, cumulative frequency, and cumulative percent of every value of all variables in a data set.

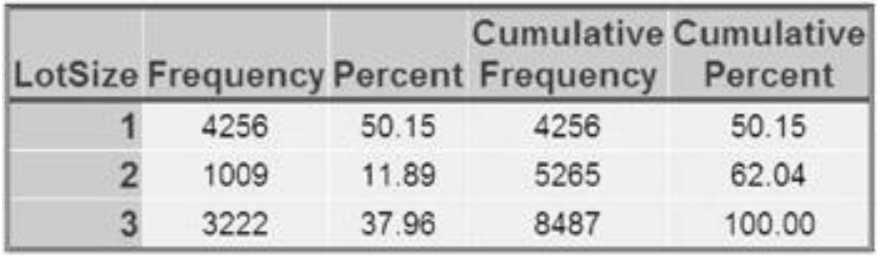
PROC FREQ: Example- Frequency Table

For example, the following FREQ procedure creates a frequency table for each variable in the data set Parts. Widgets. All the unique values are shown for ItemName, LotSize, and Region.

```
proc freq data=parts.widgets;
run;
```



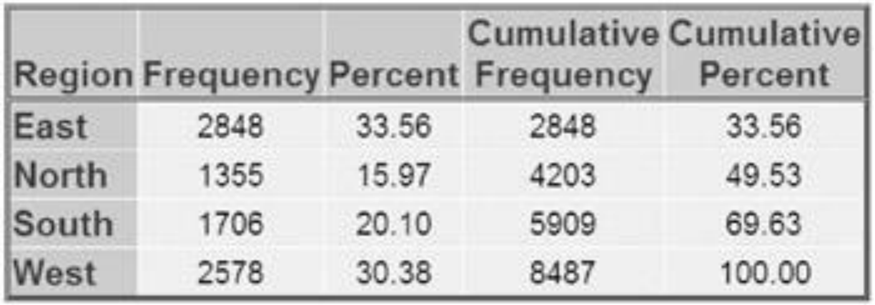
ItemName	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Bolt	2930	34.52	2930	34.52
Locknut	3106	36.60	6036	71.12
Washer	2451	28.88	8487	100.00



LotSize	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	4256	50.15	4256	50.15
2	1009	11.89	5265	62.04
3	3222	37.96	8487	100.00

To create a frequency table for a specific variable, use TABLE statement,

```
proc freq data=parts.widgets;
   TABLE Region;
run;
```



Region	Frequency	Percent	Cumulative Frequency	Cumulative Percent
East	2848	33.56	2848	33.56
North	1355	15.97	4203	49.53
South	1706	20.10	5909	69.63
West	2578	30.38	8487	100.00

PROC FREQ: Cross-tabulation Syntax

Creating Two-Way Tables
It is often helpful to crosstabulate frequencies with the values of other variables. For example, census data is typically crosstabulated with a variable that represents geographical regions.

Syntax

```
TABLES variable-1 *variable-2 <* ... variable-n>;
```

variable-1 specifies table rows and ***variable-2*** specifies table columns.
When crosstabulations are specified, PROC FREQ produces tables with cells that contain
column cell frequency
cell percentage of total frequency
cell percentage of row frequency
cell percentage of frequency.

We will revisit this two-way table later this semester with examples!

PROC UNIVARIATE: Syntax

PROC UNIVARIATE produces descriptive statistics on continuous variables just like proc means, but many more of them, and also can produce some univariate plots.

Useful in so many ways

-Instant(near-instant) analysis of large data sets

-Can manipulate variables for relational comparison

-Can provide hard-proof for QA flags

-Can view in SAS or easily output to Excel (Word, PDF)

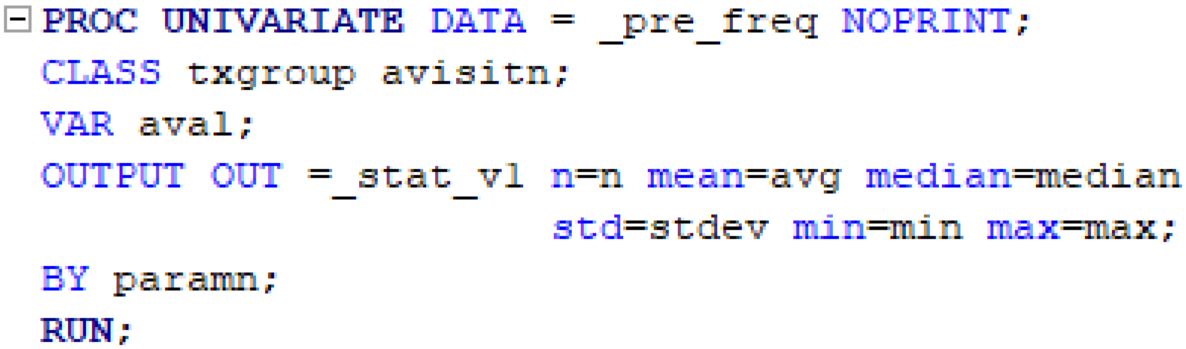
-Easily Isolate outliers

PROC UNIVARIATE: Syntax

```
PROC UNIVARIATE DATA= SASdataset
                PLOTS
                FREQ
                NORMAL
                PCTLDEF= value
                MU0= value value ... ;
    BY var-1 ... var-n;
    CLASS var-1 ... var-n;
    VAR variables;
    FREQ variable;
    HISTOGRAM < variable(s) >;
    PROBPLOT < variable(s) >;
    QQPLOT < variable(s) > ;
    OUTPUT OUT= SASdataset keyword= names...;
 RUN;
```

PROC UNIVARIATE: Example

Below is a basic example of a PROC UNIVARIATE outputting to a new dataset _STAT_V1. Note that the data had been previously sorted by PARAMN,TXGROUP and AVISITN.



```
PROC UNIVARIATE DATA = _pre_freq NOPRINT;
CLASS txgroup avisitn;
VAR aval;
OUTPUT OUT =_stat_v1 n=n mean=avg median=median
                     std=stdev min=min max=max;
BY paramn;
RUN;
```