STAT COMPUTING
BANA 6043

Lecture 2

More about SAS Data Management/Introduction to SAS Procedures

You should know …

- The structure of SAS programs (DATA steps and PROC steps)
- How to input data (CARDS, DATALINES, INFILE statements)
- How to create new variables, delete variables (SET, KEEP, DROP statements)
- How to select a subset of observations (IF… THEN; ELSE statements)
- A few simple procedures (PROC PRINT, UNIVARIATE, SORT)

- How to debug your program (read messages in the log window and use Google search)

Combining Data Set

There are several ways to combine data sets in SAS

- Concatenating
- Interleaving
- One-to-one merging

We will utilize two data sets, YR1998 and YR1999, to illustrate each method.

Example

YR1998

Day	Temp	Weather
Jan 1	33	sun
Jan 2	35	sun
Jan 3	45	clouds
Jan 5	28	snow

YR1999

Day	Temp
Jan 1	25
Jan 2	20
Jan 3	35
Jan 6	40
Jan 8	39

- Input these two data sets for future use.
  (How can you input the “Day” variable?)

```
DATA YR1998;
INPUT Day $ 1-5 Temp Weather $ ;          /*uses column input*/
CARDS;
Jan 1 33 sun
Jan 2 35 sun
Jan 3 45 clouds
Jan 5 28 snow
;
RUN;
DATA YR1999;
INPUT Day $ 1-5 Temp;
CARDS;
Jan 1 25
……
;
RUN;
```

The SAS System

Obs	Day	Temp	Weather
1	Jan 1	33	sun
2	Jan 2	35	sun
3	Jan 3	45	clouds
4	Jan 5	28	snow

The SAS System

Obs	Day	Temp
1	Jan 1	25
2	Jan 2	20
3	Jan 3	35
4	Jan 6	40
5	Jan 8	39

Method 1

- **Concatenating data sets**

```
DATA COMBINED;
    SET YR1998 YR1999;
```

- We use a SET statement to concatenate data sets.
- The total number of observations in the combined data set equals the sum of the observations in the data sets that are being combined.
- The number of variables in the combined data set is equal to the number of different variables in the data sets to be combined.
- The observations in the first data set are read first.

Run the code and check the result.

The SAS System

Obs	Day	Temp	Weather
1	Jan 1	33	sun
2	Jan 2	35	sun
3	Jan 3	45	clouds
4	Jan 5	28	snow

The SAS System

Obs	Day	Temp
1	Jan 1	25
2	Jan 2	20
3	Jan 3	35
4	Jan 6	40
5	Jan 8	39

The SAS System

Obs	Day	Temp	Weather
1	Jan 1	33	sun
2	Jan 2	35	sun
3	Jan 3	45	clouds
4	Jan 5	28	snow
5	Jan 1	25	
6	Jan 2	20	
7	Jan 3	35	
8	Jan 6	40	
9	Jan 8	39	

Think about this…

- In the previous example, the variables in the YR1999 (Day & Temp) are in fact included in the variables in the YR1998 (Day, Temp & Weather).

- What if the variables in two data sets are totally different, but we still use Method 1 to concatenate them? Can you make up an example to explore it yourself ?

Method 2

- **Interleaving data sets**
- Suppose we want a weather record in the order of dates.

```
PROC SORT data=YR1998;
        BY Day;          /*sorts within the first data set*/
PROC SORT data=YR1999;
        BY Day;          /*sorts within the second data set*/
DATA combined2;
        SET YR1998 YR1999;
        BY Day;          /*combines in the order of dates*/
```

Try it yourself!

The SAS System

Obs	Day	Temp	Weather
1	Jan 1	33	sun
2	Jan 2	35	sun
3	Jan 3	45	clouds
4	Jan 5	28	snow

The SAS System

Obs	Day	Temp
1	Jan 1	25
2	Jan 2	20
3	Jan 3	35
4	Jan 6	40
5	Jan 8	39

The SAS System

Obs	Day	Temp	Weather
1	Jan 1	33	sun
2	Jan 1	25	
3	Jan 2	35	sun
4	Jan 2	20	
5	Jan 3	45	clouds
6	Jan 3	35	
7	Jan 5	28	snow
8	Jan 6	40	
9	Jan 8	39	

Tips and Tricks

- Interleaved data set has the same number of observations as concatenated data set, but the order is different.

- In order to use a BY statement, each individual data set must first be sorted by the same variable.

- There must be a common variable in each data set.

Method 3

- **One-to-one merging of data sets**

```
DATA NEW;
    MERGE YR1998 YR1999;
```

We use MERGE statement to merge the first observation from dataset-1 with the first observation from dataset-2, the second with the second, all the way to the end of the dataset with more observations.

To-do list:

- Input the SAS code.
- Check the output and figure out how SAS process the data.
- What if the two data sets have common variables.

The SAS System

Obs	Day	Temp	Weather
1	Jan 1	33	sun
2	Jan 2	35	sun
3	Jan 3	45	clouds
4	Jan 5	28	snow

The SAS System

Obs	Day	Temp
1	Jan 1	25
2	Jan 2	20
3	Jan 3	35
4	Jan 6	40
5	Jan 8	39

The SAS System

Obs	Day	Temp	Weather
1	Jan 1	25	sun
2	Jan 2	20	sun
3	Jan 3	35	clouds
4	Jan 6	40	snow
5	Jan 8	39	

Tips and Tricks

- The number of observations in the final data set is the same as the individual dataset with **most** observations.

- The number of variables is the total number of variables in the datasets minus the number of overlapping variables.

- If two variables from different datasets have the same name, then **the values from the last dataset (listed in the MERGE statement) will replace the values from the previous datasets**.

SAS functions

- Some frequently used functions are already stored in SAS for easy use.

Example

X >>>> log(X), X={2,89,34,60}

```
DATA ONE;
  INPUT X;
  Y=LOG(X);    /*defines a new variable in data step*/
CARDS;
…..
```

- Complete the codes on your computer.

The SAS System

Obs	X	Y
1	2	0.69315
2	89	4.48864
3	34	3.52636
4	60	4.09434

SAS functions

SAS has many ready-to-use function for working with data values

- Mathematical functions
- Probability functions
- Descriptive statistics functions
- Character functions

Mathematical functions

Function	**Description**
Arithmetic	
abs(…)	Returns the absolute value of argument.
exp(…)	Returns the number e to the power of argument.
sqrt(…)	Returns the positive square root of argument.
log(…)	Returns the natural log of argument.
Trigonometric	
sin(…)	Returns the sine of argument.
cos(…)	…..
arcos(…)	
tan(…)	

Probability functions

Function	Description
probnorm(x)	Returns P(X<=x), where X is a standard normal random variable.
probchi(x,df)	Returns P(X<=x), where X is a chi-square random variable, df is the degree of freedom.
probbnml(p,n,m)	Returns P(X<=m), where X is a binomial random variable, p is the probability of success and n is the number of trials.
poisson(m,n)	Returns P(X<=n), where X is a Poisson random variable and m is the mean.

Descriptive statistics functions

Functions	Description
mean(A,B,C…)	Returns the arithmetic mean.
std(A,B,C…)	Returns the standard deviation.
sum(A,B,C…)	Returns the summation.
min(A,B,C…)	
max(A,B,C…)	
median(A,B,C…)	
range(A,B,C…)	Returns the range (max-min)

Exercise 1

Compute the positive squared root of the following values when possible. {4, 0, -9, 32, -13, -98, 100}. Make your output in the format as below.

VALUE	ROOT
4	2.0000
0	0.0000
-9	.
32	5.6569
-13	.
-98	.
100	10.0000

Answer 1

```
DATA A;
  INPUT VALUE @@;
DATALINES;
4 0 -9 32 -13 -98 100
;
RUN;

DATA B;
  SET A;
  IF VALUE>=0 THEN ROOT=SQRT(VALUE);
  ELSE ROOT= . ;
RUN;
```

Answer 2

```
DATA A;
  INPUT VALUE @@;
  IF VALUE>=0 THEN ROOT=SQRT(VALUE);
  ELSE ROOT= . ;
DATALINES;
4 0 -9 32 -13 -98 100
;
RUN;

PROC PRINT DATA=A;
RUN;
```

Exercise 2

- Calculate the cumulative distribution function (CDF) of the standard normal variable at the points -1, 0, 1, 1.5, 2, 4.

- Hint:

probnorm(x)	Returns P(X<=x), where X is a standard normal random variable.

SAS codes

```
DATA CDF_NORMAL;
 INPUT X @@;
 CDF=PROBNORM(X);
DATALINES;
 -1 0 1 1.5 2 4
;
RUN;
PROC PRINT DATA=CDF_NORMAL;
RUN;
```

The SAS System

Obs	X	CDF
1	-1.0	0.15866
2	0.0	0.50000
3	1.0	0.84134
4	1.5	0.93319
5	2.0	0.97725
6	4.0	0.99997

Exercise 3

- Calculate the point mass function of the binomial random variable (Binomial(n=10,p=0.4)) at the points 2, 5, 6, 9.

- Hint:

probbnml(p,n,m) Returns P(X<=m), where X is a binomial random variable, p is the probability of success and n is the number of trials.

Answer

```
DATA PMF_BIN;
 INPUT M_LOWER M_UPPER;
 CDF_LOWER=PROBBNML(0.4,10,M_LOWER);
 CDF_UPPER=PROBBNML(0.4,10,M_UPPER);
 PMF=CDF_UPPER-CDF_LOWER;
 M=M_UPPER;
DATALINES;
 1 2
 4 5
 5 6
 8 9
;
RUN;

PROC PRINT DATA=PMF_BIN;
 VAR M PMF;       /* Please also try removing this line to  see  the full set of variables */
RUN;
```

The SAS System

Obs	M_LOWER	M_UPPER	CDF_LOWER	CDF_UPPER	PMF	M
1	1	2	0.04636	0.16729	0.12093	2
2	4	5	0.63310	0.83376	0.20066	5
3	5	6	0.83376	0.94524	0.11148	6
4	8	9	0.99832	0.99990	0.00157	9

Exercise 4

- Read the following SAS code. Write down the output without running it.

```
DATA SUMMARY;
  INPUT TEST1 TEST2 TEST3;
  TEST_MIN=MIN(TEST1,TEST2,TEST3);
  TEST_MAX=MAX(TEST1,TEST2,TEST3);
  TEST_MEAN=MEAN(TEST1,TEST2,TEST3);
DATALINES;
88 93 91
85 74 68
87 96 79
83 88 85
;
RUN;
PROC PRINT DATA=SUMMARY;
RUN;
```

The SAS System

Obs	TEST1	TEST2	TEST3	TEST_MIN	TEST_MAX	TEST_MEAN
1	88	93	91	88	93	90.6667
2	85	74	68	68	85	75.6667
3	87	96	79	79	96	87.3333
4	83	88	85	83	88	85.3333

Break ….

LABEL & TITLE statements

- Example

```
DATA ONE;
   INPUT NAME $ @@;
LABEL   NAME= student name;
TITLE   Survey Data;
TITLE2   FROM NJ;
CARDS;
John  Jeffrey Tom
;
RUN;
PROC PRINT DATA=ONE LABEL;
RUN;
```

LABEL & TITLE statements

- Example

```
DATA ONE;
   INPUT NAME $ @@;
LABEL   NAME= student name;
TITLE   Survey Data;
TITLE2   FROM NJ;
CARDS;
John  Jeffrey Tom
;
RUN;
PROC PRINT DATA=ONE LABEL;
RUN;
```

Tips and Tricks

- The LABEL statement assigns labels to variables. It gives variables aliases.

- LABEL is an *option* in PROC PRINT. SAS will not use these aliases, but original names, if you do not put this option in PROC PRINT.

- The text of the label has to be contained in single quotes or double quotes if it contains a single quote or apostrophe inside.

  LABEL NAME = “students’ name”

  This rule is also valid in TITLE statements.

Tips and Tricks (continuing)

- LABEL (TITLE) statement can be put in either DATA step or PROC step. If found in a DATA step, the label (TITLE) is valid throughout the program. If found in a PROC step, the label (TITLE) is valid only for that procedure.

- TITLE statements appears at the top of the output. You can create more than one title by numbering the TITLE statements TITLE1, TITLE2, and so on.

- TITLE statements are in effect until changed in subsequent TITLE statements

Check this…

```
DATA ONE;
   INPUT NAME $ @@;
CARDS;
John  Jeffrey Tom
;
RUN;
PROC PRINT DATA=ONE;
LABEL NAME= “students’ name”;
RUN;
```

LABEL statement is placed in the PROC step

```
PROC PRINT DATA=ONE;
RUN;
```

Can we have labels if we print it again?

PROC UNIVARIATE

- PROC UNIVARIATE generates descriptive statistics: mean, standard diviation, quantiles, minimum value and maximum value.

- General Form

```
PROC UNIVARIATE DATA=…  <OPTIONS>;
        BY VARIABLES;
        VAR VARIABLES;
```

Example

The ‘FOOTBALL’ data set consists of two variables, ‘TEAM’ and ‘SCORE’. What if we want SAS to produce statistics for every single team?

Team	Score
Cincinnati	18
UOhio	27
UOhio	39
Cincinnati	16
Cincinnati	29
UOhio	42

```
DATA FOOTBALL;
  INPUT TEAM $ SCORE @@;
CARDS;
Cincinnati 18 UOhio 27 UOhio 39
Cincinnati 16 Cincinnati 29 UOhio 42
;
RUN;
PROC SORT DATA=FOOTBALL;
  BY TEAM;              /*data has to be sorted first if  BY statement is used*/
RUN;                    /*in PROC UNIVARIATE*/
PROC UNIVARIATE DATA=FOOTBALL;
  BY TEAM;               /* tells SAS to sort data by TEAM*/
  VAR SCORE;            /* tells SAS to produce statistics of SCORE*/
RUN;
```

The UNIVARIATE Procedure
Variable: SCORE

TEAM=Cincinna

Moments			
N	3	Sum Weights	3
Mean	21	Sum Observations	63
Std Deviation	7	Variance	49
Skewness	1.57434402	Kurtosis	.
Uncorrected SS	1421	Corrected SS	98
Coeff Variation	33.3333333	Std Error Mean	4.04145188

Basic Statistical Measures			
Location		Variability	
Mean	21.00000	Std Deviation	7.00000
Median	18.00000	Variance	49.00000
Mode	.	Range	13.00000
		Interquartile Range	13.00000

Tests for Location: Mu0=0				
Test		Statistic	p Value	
Student's t	t	5.196152	Pr > |t|	0.0351
Sign	M	1.5	Pr >= |M|	0.2500
Signed Rank	S	3	Pr >= |S|	0.2500

Quantiles (Definition 5)	
Quantile	Estimate
100% Max	29
99%	29
95%	29
90%	29
75% Q3	29
50% Median	18
25% Q1	16
10%	16
5%	16
1%	16
0% Min	16

Extreme Observations			
Lowest		Highest	
Value	Obs	Value	Obs

Tips and Tricks

- PROC UNIVARIATE can produce statistics for one variable. It is allowed to sort the data set into several subsets and statistics can be produced for each subset.

- If BY statement is used in PROC UNIVARIATE, the data set should have been sorted preceding this procedure.

- Options:

  NORMAL option: testing for normality.

  PLOT option: generating a stem-and-leaf plot, a box plot and a normal probability plot

Compare two sets of codes

- Set 1

```
PROC UNIVARIATE DATA=AAUP;      /*AAUP data used in Homework */
 VAR AS_full;
 Run;
```

- Set 2

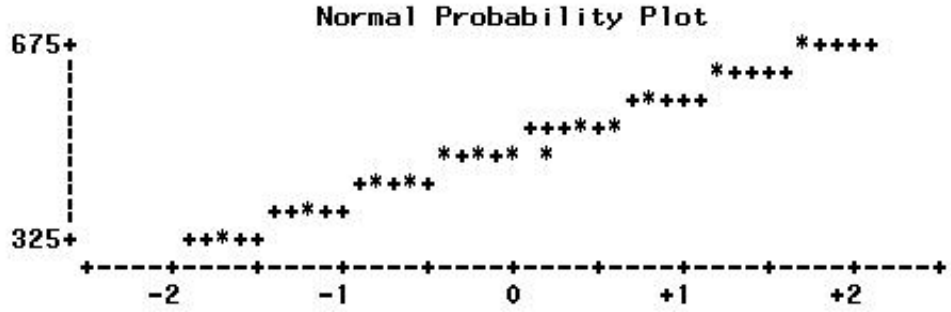
```
PROC UNIVARIATE DATA=AAUP NORMAL PLOT;        /* Two options added */
 VAR AS_full;
 Run;
```

Stem	Leaf	#	Boxplot
6	9	1	0
6	1	1	¦
5	8	1	¦
5	013	3	+------+
4	567	3	*--+--*
4	44	2	+------+
3	6	1	¦
3	4	1	¦

Multiply Stem.Leaf by 10**+2

Tests for Normality

Test		--Statistic---		-----p Value------
Shapiro-Wilk	W	0.963171	Pr < W	0.8017
Kolmogorov-Smirnov	D	0.148867	Pr > D	>0.1500
Cramer-von Mises	W-Sq	0.047983	Pr > W-Sq	>0.2500
Anderson-Darling	A-Sq	0.273461	Pr > A-Sq	>0.2500



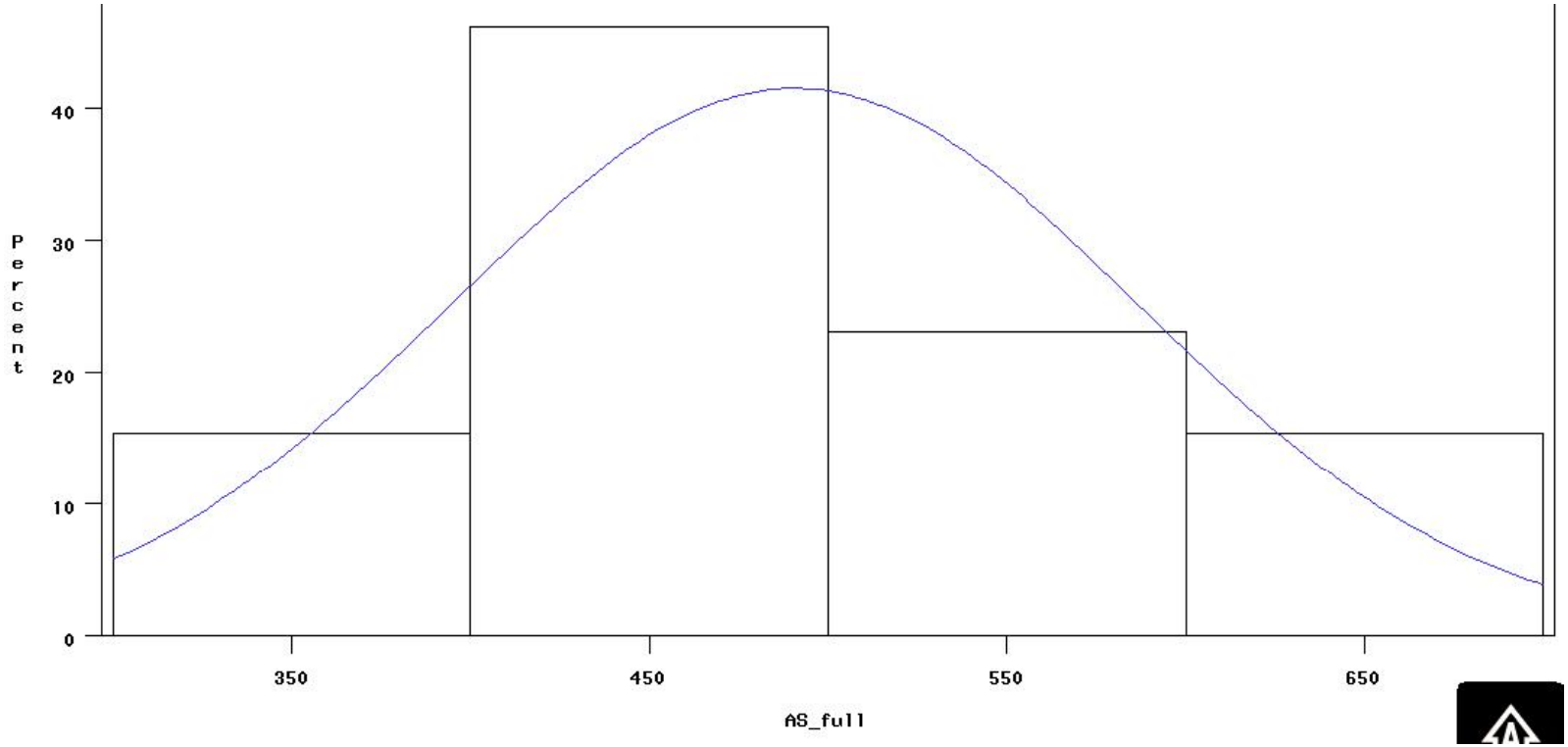
Compare another two sets of codes

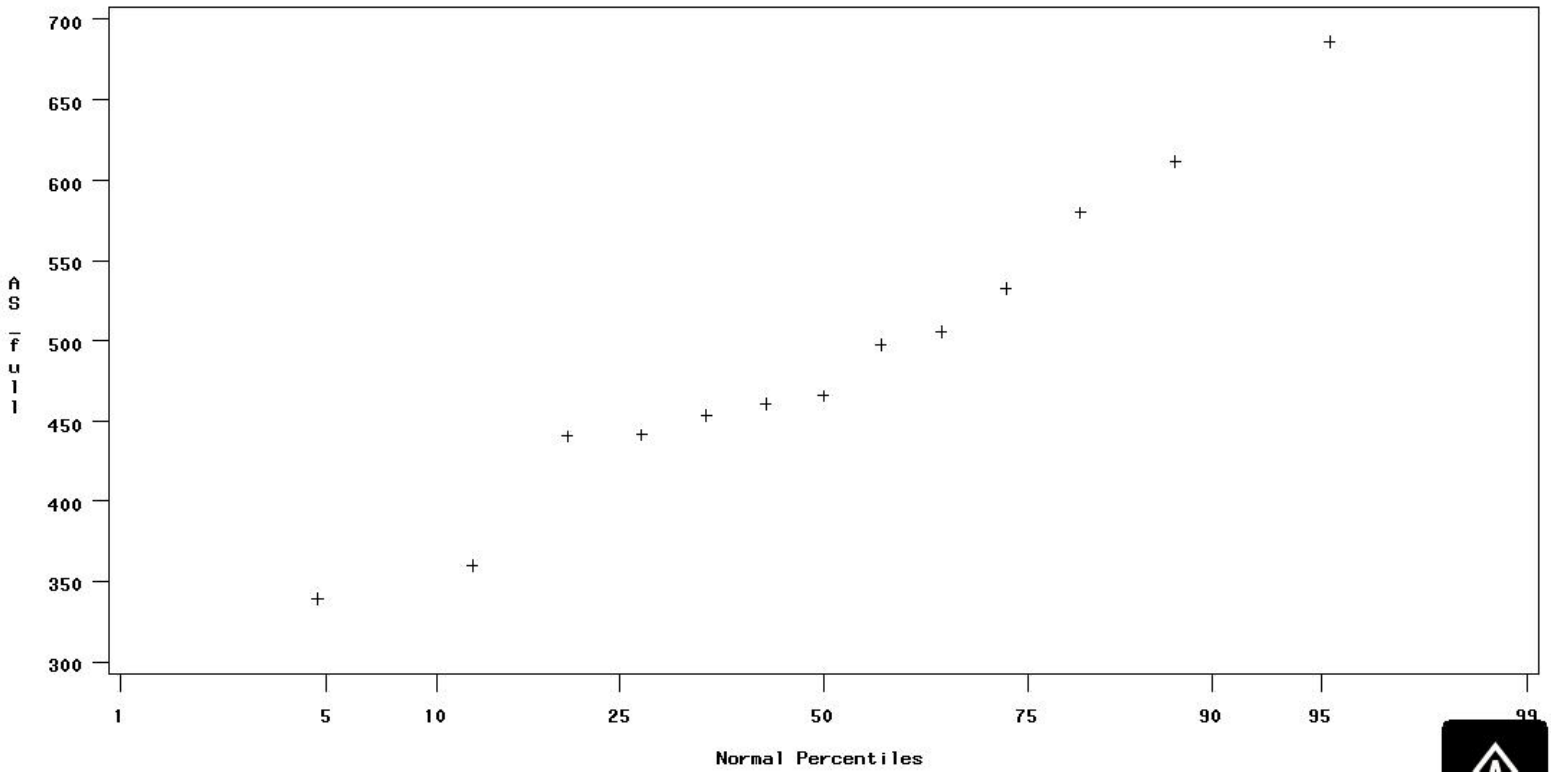
- Set 1

```
PROC UNIVARIATE DATA=AAUP;      /*AAUP data used in Homework */
 VAR AS_full;
 Run;
```

- Set 2

```
PROC UNIVARIATE DATA=AAUP;
 HISTOGRAM AS_full / NORMAL;      /*  HISTOGRAM  variables / <OPTIONS>  */
 PROBPLOT AS_full / NORMAL;       /*  PROBPLOT  variables / <OPTIONS>  */
Run;
```


Percent
40
30
20
10
0
350
450
550
650
AS_full


700
650
600
550
500
450
400
350
300
AS_full
1
5
10
25
50
75
90
95
99
Normal Percentiles

PROC MEANS

- PROC MEANS outputs the basic descriptive statistics in a more concise way than UNIVARIATE.

- By default, it generates the sample size, the mean, the standard deviation and the minimum and maximum values. It does not generate plots or percentiles.

- **General Form** (similar to UNIVARIATE)

  PROC MEANS DATA=… <OPTIONS>;

  BY VARIABLES;

  VAR VARIABLES;

Exercise 5

The data set “CLINIC” consists of two variables, “TYPE” and “SCORE”. “TYPE” refers to what patients take. “SCORE” is a kind of health score of patients.

TYPE	SCORE	TYPE	SCORE
drug	8	drug	9
drug	10	placebo	7
placebo	5	placebo	6
drug	9	placebo	6

Step 1. Input the data set. Label “TYPE” and “SCORE” as “drug or placebo” and “health score” respectively.

Step 2. Calculate the means of health score for patients taking drug and for patients taking placebo respectively.

Answer

```
DATA CLINIC;
            INPUT TYPE $ SCORE @@;
   LABEL TYPE=drug or placebo
               SCORE=health score;
CARDS;
drug 8 drug 9 drug 10 placebo 7 placebo 5 placebo 6 drug 9 placebo 6
;
RUN;
PROC PRINT DATA=CLINIC LABEL;
RUN;
PROC SORT DATA=CLINIC;
   BY TYPE;
RUN;
PROC MEANS DATA=CLINIC;
   BY TYPE;
   VAR SCORE;
RUN;
```

---------- drug or placebo=drug ----------

The MEANS Procedure

Analysis Variable : SCORE health score

N	Mean	Std Dev	Minimum	Maximum
4	9.0000000	0.8164966	8.0000000	10.0000000

---------- drug or placebo=placebo ----------

Analysis Variable : SCORE health score

N	Mean	Std Dev	Minimum	Maximum
4	6.0000000	0.8164966	5.0000000	7.0000000

Options for PROC MEANS

- n, min, max, mean, std

  --- These are the default values if none are specified.

- nmiss, range, sum, var, stderr, t, probt, q1, median, q3, qrange

  --- These options can also be specified in PROC MEANS

Example

```
PROC MEANS DATA=CLINIC N MEAN STD RANGE;
  BY TYPE;
  VAR SCORE;
RUN;
```

PROC FREQ

- **PROC FREQ** generates tables for data that are in categories.

- **General Form**

```
PROC FREQ DATA=…;
    TABLE   A   B*C / <OPTIONS>;
```

- For one variable, a one-way table summarizes all the values of the variable, including how many observations each value has and the percent for each value.

- For two variables, a two-way table contains cell frequencies, cell percent of total, cell percent of row total and cell percent of column total.

Example

STATE	TYPE	SCORE
NJ	drug	8
PA	drug	10
NJ	placebo	5
NJ	drug	9

STATE	TYPE	SCORE
PA	drug	9
PA	placebo	7
NJ	placebo	6
NJ	placebo	6

Question:

In the SAS system, you already have a data set “clinic”, which contains the two variables “TYPE” and “SCORE”. How would you create the above data set with minimum input effort?

```
DATA CLINIC2;
 SET CLINIC;
 INPUT STATE $ @@;
DATALINES;
 NJ PA PA PA NJ NJ NJ NJ
;
RUN;

PROC FREQ DATA=CLINIC2;
  TABLE SCORE STATE STATE*TYPE;
RUN;
```

The FREQ Procedure

health score

SCORE	Frequency	Percent	Cumulative Frequency	Cumulative Percent
5	1	12.50	1	12.50
6	2	25.00	3	37.50
7	1	12.50	4	50.00
8	1	12.50	5	62.50
9	2	25.00	7	87.50
10	1	12.50	8	100.00

STATE	Frequency	Percent	Cumulative Frequency	Cumulative Percent
NJ	5	62.50	5	62.50
PA	3	37.50	8	100.00

Table of STATE by TYPE

STATE TYPE(drug or placebo)

Frequency Percent Row Pct Col Pct	drug	placebo	Total
NJ	1 12.50 20.00 25.00	4 50.00 80.00 100.00	5 62.50
PA	3 37.50 100.00 75.00	0 0.00 0.00 0.00	3 37.50
Total	4 50.00	4 50.00	8 100.00

Options for TABLE statements

A few important options:

- chisq --- computes the chi-square statistic for testing for independence or homogeneity in two-way tables
- exact --- performs Fisher's exact test for tables larger than 2 X 2.
- expected --- computes the expected counts for two-way tables.

Example:

```
PROC FREQ DATA=CLINIC2;
        TABLE STATE*TYPE / CHISQ EXPECTED;
RUN;
```

The FREQ Procedure

Statistics for Table of STATE by TYPE

Statistic	DF	Value	Prob
Chi-Square	1	4.8000	0.0285
Likelihood Ratio Chi-Square	1	6.0863	0.0136
Continuity Adj. Chi-Square	1	2.1333	0.1441
Mantel-Haenszel Chi-Square	1	4.2000	0.0404
Phi Coefficient		-0.7746	
Contingency Coefficient		0.6124	
Cramer's V		-0.7746	

WARNING: 100% of the cells have expected counts less than 5. Chi-Square may not be a valid test.

Fisher's Exact Test

Cell (1,1) Frequency (F)	1
Left-sided Pr <= F	0.0714
Right-sided Pr >= F	1.0000
Table Probability (P)	0.0714
Two-sided Pr <= P	0.1429

Sample Size = 8

Table of STATE by TYPE

STATE TYPE(drug or placebo)

Frequency Expected Percent Row Pct Col Pct	drug	placebo	Total
NJ	1 2.5 12.50 20.00 25.00	4 2.5 50.00 80.00 100.00	5 62.50
PA	3 1.5 37.50 100.00 75.00	0 1.5 0.00 0.00 0.00	3 37.50
Total	4 50.00	4 50.00	8 100.00