EXST SAS Lab
Lab #8: More data step and t-tests

Objectives

1. Input a text file in column input
2. Output two data files from a single input
3. Modify datasets with a **KEEP** statement or option
4. Prepare two different types of **BOXPLOT**
5. Produce several versions of paired t-tests

The Example:

The data for this example is the percent of women in the labor force in selected US cities for the two years, 1972 and 1968. The small data set will be input to SAS as part of the SAS program, not as an external data set. There are several potential issues with the input of this data. First, the name of the city frequently has imbedded blanks, periods, comas and even a slash. If a CSV file was to be used, these would not be a problem, with the possible exception of the comma. However, a CSV file saved from EXCEL would address this by enclosing the city names in double quotes. Input could then be done in SAS with the **DLD** option.

City	1972	1968
N.Y.	.45	.42
L.A.	.50	.50
Chicago	.52	.52
Philadelphia	.45	.45
Detroit	.46	.43
San Francisco	.55	.55
Boston	.60	.45
Pitt.	.49	.34
St. Louis	.35	.45
Connecticut	.55	.54
Wash., D.C.	.52	.42
Cinn.	.53	.51
Baltimore	.57	.49
Newark	.53	.54
Minn/St. Paul	.59	.50
Buffalo	.64	.58
Houston	.50	.49
Patterson	.57	.56
Dallas	.64	.63

In this particular case the odd symbols will be handled with alternative programming in the **INPUT** statement. The statements below show the input statement and the first two lines of data. I have also added a "ruler" in the form of a comment statement to locate the column numbers.

```
   input City $ 1-15 @16 Year1972 Year1968;
*---+----1----+----2----+----3----+----4----+----5;
datalines;
N.Y.            .45    .42
L.A.            .50    .50
```

The input statement reads the first variable, **CITY**, as a character string by reading columns 1 through 15. Since the variable is a character string, all of the odd symbols will be read accurately, and since the columns are specified, the variable length will be known.

After the SAS input statement reads the first 15 columns it will continue its usual input mode starting at column 16. If the input statement simply listed the remaining two variables after the "**1-15**" , they would be read correctly since they are separated by spaces. An additional option was added here for demonstration purposes. The "**@16**" on the input line causes the input statement to start reading the next variable in column 16 after reading the first 15 columns. This option is useful if you wish to skip some columns in the data set, but in this case SAS would have resumed reading in column 16 anyway.

The length of variables in SAS

Setting variable lengths in SAS is a little complicated. Numeric variables use up to 8 "bytes", a size that allows up to 15 digits for continuous variables (sometimes referred to as double precision). Numbers can be stored with fewer bytes, as few as two or three depending on the operating system, but with a loss of precision. SAS provides a table showing the loss of precision for continuous numbers (second column) and the largest integer that can be stored **exactly** for discrete numbers (third column) for a given number of bytes (first column).

Bytes	Significant Digits	Largest Integer
3	3	8,192
4	6	2,097,152
5	8	536,870,912
6	11	137,438,953,472
7	13	35,184,372,088,832
8	15	9,007,199,254,740,992

Since many integer variables have values that never exceed 8000, they can be reproduced exactly with 3 bytes instead of the default 8 bytes. This can reduce the size of large databases and speed up sorting and processing considerably. The length can be specified in a **LENGTH** statement. For example, the statement below would modify the length of variables **MONTH**, **DAY**, **YEAR**, **STATIONNO** and **AGE** to all be represented with only 3 bytes.

```
LENGTH MONTH DAY YEAR STATIONNUMBER AGE 3;
```

The best placement of the length statement is after the **DATA** statement, but before the **INFILE** and **INPUT** statements.

The default for **numeric** variables is 8 bytes, but the length for **character** variables is more complicated. Theoretically, SAS will judge the number of bytes needed for a character string based on the first few observations of the dataset for each variable. However, for variables with larger character strings, or highly variable string lengths, it is much safer to declare a length equal to, or greater than, the maximum length. Allowing one byte for each character, the **LENGTH** statement for a 15 character city name and a 12 character state name would be,

```
LENGTH CITY $ 15 STATE $ 12;
```

Another problem occurs when creating character variables. For example, if you have sex coded as a variable called **SEX**, with 1 for male and 2 for female, and you want to create a variable called **GENDER** with the values "male" and "female" you might try the following.

```
IF SEX EQ 1 THEN GENDER = 'MALE'; ELSE GENDER = 'FEMALE';
```

Or you could try,

```
GENDER = 'MALE'; IF SEX EQ 2 THEN GENDER = 'FEMALE';
```

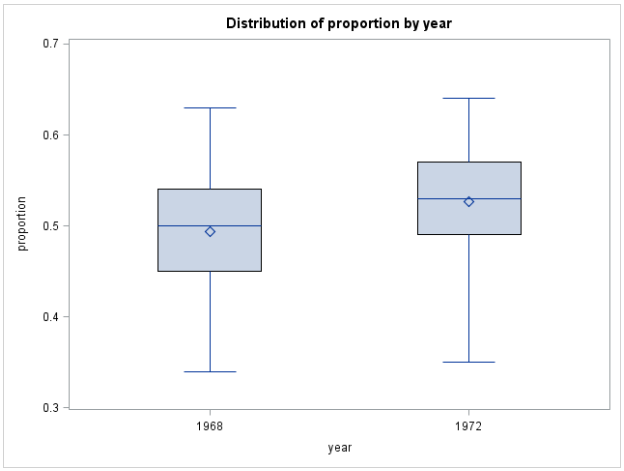
Both work, but when SAS first creates the variable **GENDER** for males, it characterizes the length needed as 4 bytes. Then, when it needs to set a value of **GENDER** to **female**, it only allows 4 characters, so **GENDER** is recorded as "**fema**" for females. Ergo, if you are going to create a character variable, either set the longest one first or declare the length in a length statement.

Outputting two datasets in a DATA step

There are analyses where it is best to have multiple levels of a variable in separate columns. For example, if the data set has been taken on 30 sample sites in 4 seasons, should these values be placed on separate columns (spring, summer, fall, winter) or should each sample observation be placed on a separate line. This option might be referred to as the multivariate option where, in EXCEL, there would be 30 rows of data (one for each site) and the values taken in each season would be placed in separate columns with the season as the variable name. The second option, sometimes termed the univariate option, would require 120 rows in EXCEL, first the 30 sites with a variable identifying the

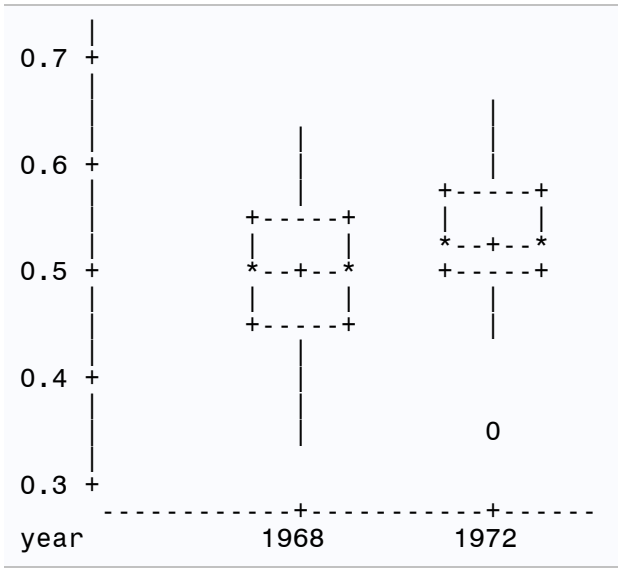
values as **spring** values, then 30 sites repeated anew in 30 additional rows with a variable specifying that those are **summer** values, etc. There are some procedures that require the multivariate arrangement and some that require the univariate arrangement. In this week's assignment we will prepare both.

More about boxplots



Previously we prepared boxplots. In the particular version prepared by **PROC BOXPLOT**, the lower and upper limits of the box are the first and third quartile respectively. The line through the middle of the box is the second quartile or the 50th percentile, also called the median. The whiskers that extend out from the box show the range of the data from the lowest value to the highest (called the minimum and maximum values or the 0th and 100th percentile). The circle in the box represents the location of the mean. These boxplots are high resolution graphics and are only available in the HTML output.

The **UNIVARIATE** procedure will also prepare side by side boxplots if **PROC UNIVARIATE** is run with the **plot** option and a **BY** statement. The interpretation is similar, but in contrast to the boxplot procedure, the **UNIVARIATE** procedure creates whiskers that stretch 1.5 interquartile distances above and below the median. An interquartile distance is the difference between the third and the first quartile. Observations beyond the 1.5 interquartile distance are marked with a "0" if they are between 1.5 and 3 interquartile distances and are marked with a "*" outside this distance. SAS notes that the extreme values, marked 0 or * could be possible outliers. Outliers are values that are anomalous, and possibly erroneous.



The example analysis

This week's example examines the proportion of women in the workforce for two years during a period of very high inflation. The desired analyses are t-tests, so there is some repetition of the data step and analytical techniques from previous assignments.

The data is coded in three columns with a city name, 1968 rate and 1972 rate in separate columns. A calculation of the difference in the proportions between the two years is to be added in the data step. This dataset is similar to the "multivariate" style of data, and is appropriate for some of the procedures needed for this assignment. However, other procedures will require a "univariate" style dataset. It is possible to output two or more datasets in a single data step.

In the example below the original data is already in the "multivariate" style, so it can be read and output immediately as the dataset "**LABORFORCE**" with the aforementioned variables. The only addition is the calculation of a difference between the years. Therefore, immediately following the input statement and the calculation of the difference, the original data is output to the dataset called **LABORFORCE** with an output statement.

The **keep** statement for the **LABORFORCE** dataset allows only the 4 variables to stay on the dataset. Any other variables created will not be in the **LABORFORCE** dataset.

```
data LaborForce (keep = City diff Year1972 Year1968)
    Univariate (keep = City year proportion);
  Label City    = 'City in the United States'
       Year1972 = 'Labor Force Participation rate of women in 1972'
       Year1968 = 'Labor Force Participation rate of women in 1968';
  input City $ 1-15 @16 Year1972 Year1968;
       diff = Year1972 - Year1968;
  output laborforce;
  year = 1972; proportion = year1972; output Univariate;
  year = 1968; proportion = year1968; output Univariate;
*---+----1----+----2----+----3----+----4----+----5;
datalines;
N.Y.          .45   .42
L.A.          .50   .50
```

The second dataset created is called "univariate". It will have the variable called **CITY** plus two additional variables called **YEAR** and **PROPORTION** which must be created. This creation will require outputting two data lines for each original data line, one for each year. The first line output has the variable **YEAR** set to 1972 and the value of **PROPORTION** set equal to the value of the variable **YEAR1972**. This is followed by an **OUTPUT UNIVARIATE;** statement that outputs the first line to that dataset. Similarly, the second line of output has the variable **YEAR** set to 1968 and the value of **PROPORTION** set equal to the value of the variable **YEAR1968**. This is followed again by an **OUTPUT UNIVARIATE;** that outputs the second line to that dataset.

Once the data input as finished, the **UNIVARIATE** data is sorted and used to do a **PROC BOXPLOT.** This procedure produced the first of the boxplots above. This was followed by a **PROC UNIVARIATE** with almost all output suppressed except the plots. This will result in several plots including the side by side boxplots shown in the second boxplot above.

```
proc sort data=Univariate; by year; run;
proc boxplot data=Univariate; plot proportion*year; run;

proc univariate data=Univariate plot; by year;
   Title2 'proc univariate examination BY year';
   var proportion;
   ods exclude ExtremeObs ExtremeValues Modes
       Moments MissingValues Quantiles TestsForLocation;
options ls=64 ps=35; run; options ls=90 ps=56;
```

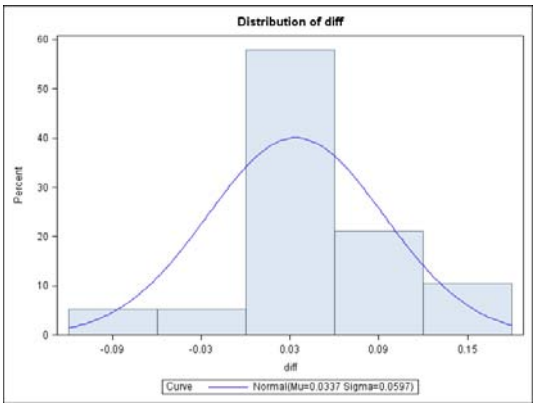
Both of these analyses require the univariate style data set. The rest of the analyses below will employ the multivariate style dataset called "**LABORFORCE**".

The second part of the example program involves solving a paired t-test using several different procedures and options, similar to a previous assignment. First, using **PROC UNIVARIATE** on the variable **DIFF**, do a two tailed test of hypothesis of the difference against zero (i.e. $H_0 : \delta = 0$ where $H_0 : \delta = \mu_{72} - \mu_{68}$). A value of $\alpha = 0.05$ is used as a decision criteria.

The next **PROC UNIVARIATE** is done with the addition of plots and confidence intervals and for tests of normality, but suppresses most other output. A histogram statement is included of the variable **DIFF**

with a comparison requested against a normal distribution. This graphic will only be available in the html output or the results viewer.

```
proc univariate data=LaborForce plot normal CIBasic;
   Title2 'Paired t-test with proc univariate';
   Title3 'Two-tailed hypothesis';
   var diff;
   ods exclude BasicMeasures MissingValues Quantiles
               ExtremeValues;
   histogram diff / normal; run;
```



The histogram statement produces a histogram of the variable diff and superimposes a normal distribution for the calculated mean and variance of the observed distribution. This plot can be used to determine where the greatest discrepancies occur when the test of normality is rejected. The normal probability plot performs a similar function. Notice that where the theoretical normal curve is above the histogram, on the left of the plot, the asterisks of the normal probability plot are above the theoretical values designated by the "+++" signs. In the middle of the chart where the observed values exceed expectations based on the normal curve, the asterisks fall below the plus signs. The right of the chart fits pretty well, as reflected by the asterisks occurring amid the plus signs up to near the end. At the extreme right, where the normal curve again falls below the observed values, the asterisks again occur above the plus signs. So, when normality is rejected, the function of these plots is to help determine what aspects of the observed distribution do not conform to the normal distribution.

```
                    Normal Probability Plot
 0.175+                                                  *   * ++++++
      |                                                *++++++++
      |                                          ++*+*+*+*++
 0.025+                     * * * ***+***+**
      |                 *++++++++
      |    ++++*+++
-0.125++++
       +----+----+----+----+----+----+----+----+----+----+----+
            -2        -1         0        +1        +2
```

The remaining task is to use a **PROC TTEST** to do a one-tailed test of the difference between the values for 1972 and 1968, similar to the previous assignment. Conduct two t-tests, one for the variable **DIFF** and one for the original variables.

```
PROC ttest data=LaborForce sides=u;
   VAR diff;
   Title2 'Paired t-test if DIFF with Proc TTEST';
   Title3 'One-tailed hypothesis';
RUN;

PROC ttest data=LaborForce sides=u;
   paired Year1972 * Year1968;
   Title2 'Paired t-test with Proc TTEST';
   Title3 'One-tailed hypothesis';
RUN;
```

As with that previous **PROC TTEST** assignment, the results of the two approaches are likely to be identical.

Assignment 8

The dataset gives the price, in cents per pound, received by fishermen and vessel owners for various species of fish and shellfish in 1970 and 1980. The data is from Moore, David S., and George P. McCabe (1989). *Introduction to the Practice of Statistics*. We want to test the hypothesis that the price of fish did not change between 1970 and 1980 (e.g. test the difference where $H_0 : \delta = 0$ and $H_0 : \mu_{80} - \mu_{70}$.

Use $\alpha = 0.05$ as a decision criteria.

Type_Fish	Price_1970	Price_1980
COD	13.1	27.3
FLOUNDER	15.3	42.4
HADDOCK	25.8	38.7
MENHADEN	1.8	4.5
OCEAN PERCH	4.9	23.0
SALMON, CHINOOK	55.4	166.3
SALMON, COHO	39.3	109.7
TUNA, ALBACORE	26.7	80.1
CLAMS, SOFT-SHELLED	47.5	150.7
CLAMS, BLUE HARD-SHELLED	6.6	20.3
LOBSTERS, AMERICAN	94.7	189.7
OYSTERS, EASTERN	61.1	131.3
SEA SCALLOPS	135.6	404.2
SHRIMP	47.6	149.0

Answer all questions about hypothesis tests by stating the outcome (REJECT the null hypothesis or FAIL to reject the null hypothesis) and Give a P-value where a relevant P-value is available. Turn in your log and the list output or results viewer output for relevant sections only. You may write answers to questions on the log, or on a separate page.

Task 1: Produce two datasets, one with the original variables (fish type, price in 1970 and price in 1980) and a second dataset with variables for the fish type, year and price. The second dataset should have twice as many observations as the first. (2 points)

Task 2: Obtain side by side boxplots using both **PROC BOXPLOT** and **PROC UNIVARIATE** suppress most output from the **PROC UNIVARIATE**. (1 points)

Question 1: Does either boxplot suggest any possible outliers for this data set? (1 point)

Task 3: Run a two-tailed t-test against the on the difference in price between the two years using **PROC UNIVARIATE**. Include a histogram with a normal curve. (1 point)

Question 2: Is there a significant difference between the means in the two tailed test? (1 point)

Question 3: Is there evidence that the data departs from a normal distribution? If so, describe the evidence. (1 point)

Task 4: Suppose the investigators were actually interested in a one tailed test, hypothesizing an increase in price from 1970 to 1980. Use **PROC TTEST** to conduct this test on the variable **DIFF**. Note that if you calculated diff as the 1980 price minus the 1970 price you would be testing against the upper bound. (1 point)

Task 5: Repeat the test in Task 3 using **PROC TTEST** and the two original variables instead of the calculated difference. (1 point)

Question 4: Is there a significant difference between the means when done as a one tailed test for either of the two one-tailed tests? (1 point)