Analyzing Counts and Tables, Merging, Concatenating and Transposing Data Sets

STAT 3505

Week 6 (February 22, 2024)

Gunes Fleming Ph.D.

Analyzing Counts and Tables: **PROC FREQ**

- PROC FREQ is a multi-purpose procedure in SAS for analyzing count data.
- It can be used to obtain frequency counts for one or more single variables or to create two-way tables (crosstabulations) from two variables.
- PROC FREQ can also be used to perform statistical tests on count data.

Analyzing Counts and Tables: **PROC FREQ**

- PROC FREQ can also be used to perform statistical tests on count data.
- Syntax:
  - **PROC FREQ** <options(s)>; <statements> TABLES requests </ options> ;
  - Common Options
    - **DATA =** Specifies which data set to use.
    - **ORDER=** Specifies the order results is listed in the output table (FREQ, FORMATTED, ORDER (p164).
    - **PAGE** Specifies that only one table will appear per page. (Not for HTML)
    - **COMPRESS** Begins next table on same page when possible. (Not for HTML)

Analyzing Counts and Tables: **PROC FREQ**

- PROC FREQ can be used for
  - Chi-Square
  - Relative Risk
  - CMH Test
  - Fisher's Exact Test
  - KAPPA
  - etc.

PROC FREQ Statements

- Common Statements
  - **BY**variable(s): Creates tables for each BY group
  - **EXACT:** Produces exact test forselected statistics
  - **OUTPUT=** Creates an output data set containing statistics from an analysis
  - **WEIGHT var:** Identifies a weight variable that contains summarized counts

  - The TABLES statement is required for all of the examples in this section. Its format is:

    TABLES <variable-combinations/options>;

    where variable-combinations specify frequency or crosstabulation tables.

TABLES Statement

Description	TABLE statement
Specify counts for a single table	TABLE A;
Specify a crosstabulation between two variables	TABLE A*B;
Specify several tables	TABLE A*B B*C X*Y; Also, TABLE A*(B C D); is the same as TABLE A*B A*C A*D;
Use a range of variables in a TABLE statement	TABLE (A - - C)*X; is the same as TABLE A*X B*X C*X;

Options for TABLES Statement

- Options for the TABLES statement follow a slash (/).

  For example,

  ```
  TABLES A*B / CHISQ;
  ```

- Requests that the chi-square and related statistics be reported for the crosstabulation A*B.

Options for TABLES Statement

- **AGREE** Request KAPPA statistic (inter-rater reliability) To include significance tests, add the following statement:
  - **TEST** option. For example: TABLES A*B / /AGREE; TEST KAPPA;
- **CHISQ** Requests chi-square and related tests
- **RELRISK** Requests relative risk calculations
- **FISHER** Requests Fisher's Exact test for tables greater than 2x2
- **SPARSE** Requests all possible combinations of variable levels
- **MISSING** Requests that missing values be treated as non-missing
- **CELLCHI2** Displays the contribution to chi-square for each cell
- **NOCOL** Suppresses column percentages for each cell
- **NOCUM** Suppresses cum. freq. and cum. % in one-way table
- **NOFREQ** Suppresses frequency count for each cell
- **NOPERCENT** Suppresses row %, and column % in tables
- **NOPRINT** Suppresses table display
- **NOROW** Suppresses row percentages

Testing Goodness-of-Fit in a One-Way Table

- A goodness-of-fit test of a single population is a test to determine if the distribution of observed frequencies in the sample data closely matches the expected number of occurrences under a hypothetical distribution for the population.
- The observations are assumed to be independent, and each data value can be counted in one and only one category.
- It is also assumed that the number of observations is fixed.
- The hypotheses being tested are:

  H_0: The population follows the hypothesized distribution.

  H_a: The population does not follow the hypothesized distribution.

- A chi-square statistic is calculated, and a decision can be made based on the p-value associated with that statistic

Testing Goodness-of-Fit in a One-Way Table

- Example:

```
PROC FREQ data = carsdata2 order=freq; *descending order;
   TABLES nationality/ nocum chisq testp=(0.7 0.15 0.1 0.05); * expected proportions;
RUN;
```

(This test is Pearson chi-square goodness-of-fit test.)

- The TESTP= and TESTF= options in PROC FREQ specify the null hypothesis (expected) proportions or frequencies for a chi-square goodness-of-fit test on a one-way table.
- Note that you must use the ORDER=DATA option to ensure that the hypothesized ratios listed in the TESTP= statement match up correctly with the categories in the table.
- If the TESTP= option is omitted, SAS will assume the proportions within the category are equal. For a categorical variable with 4 possible values, the SAS default would be TESTP = (0.25 0.25 0.25 0.25).

Testing Goodness-of-Fit in a One-Way Table

- Example:

```
PROC FREQ data = carsdata2 order=freq; *descending order;
   TABLES nationality/ nocum chisq testp=(0.7 0.15 0.1 0.05); * expected proportions;
RUN;
```

- Result:

The FREQ Procedure

Nationality	Frequency	Percent	Test Percent
AMERICAN	61028	83.62	70.00
OTHER ASIAN	8033	11.01	15.00
TOP LINE ASIAN	3722	5.10	10.00
MANUAL	200	0.27	5.00

Chi-Square Test for Specified Proportions	
Chi-Square	7722.3995
DF	3
Pr > ChiSq	<.0001

- Notice that in this case, the p-value for the chi-square test is less than 0.05, which leads us to reject the null hypothesis and conclude there is evidence to conclude that the data does not follow the hypothesized distribution..

Another Example: Mendel's Pea Experiment

- We will use data from an experiment conducted by the nineteenth-century scientist Gregor Mendel. According to a genetic theory, crossbred pea plants show a 9:3:3:1 ratio of yellow smooth, yellow wrinkled, green smooth, and green wrinkled offspring.
- Out of 250 plants, under the theoretical ratio (distribution) of 9:3:3:1, you would expect:

$(9/16) \times 250 = 140.625$ yellow smooth peas (56.25%)

$(3/16) \times 250 = 46.875$ yellow wrinkled peas (18.75%)

$(3/16) \times 250 = 46.875$ green smooth peas (18.75%)

$(1/16) \times 250 = 15.625$ green wrinkled peas (6.25%)

- After growing 250 of these pea plants, Mendel observed 152 have yellow smooth peas, 39 have yellow wrinkled peas, 53 have green smooth peas, 6 have green wrinkled peas. Do these offspring support the hypothesized ratios? Run a goodness-of-fit test to assess whether the observed phenotypic frequencies seem to support the theory.

Another Example: Mendel's Pea Experiment

- Program:

```
proc freq data=genedata order=data;
  WEIGHT NUMBER;
  TITLE 'GOODNESS OF FIT ANALYSIS - MENDEL EXPERIMENT';
  TABLES COLOR / NOCUM PLOTS=NONE CHISQ TESTP=(0.5625 0.1875 0.1875 0.0625);
run;
```

- Notice here the WEIGHT statement. The WEIGHT statement names a numeric variable that provides a weight for each observation in the input data set. The WEIGHT statement is most commonly used to input cell count data.
- Results:

GOODNESS OF FIT ANALYSIS - MENDEL EXPERIMENT

The FREQ Procedure

COLOR	Frequency	Percent	Test Percent
YELLOWSMOOTH	152	60.80	56.25
YELLOWWRINKLE	39	15.60	18.75
GREENSMOOTH	53	21.20	18.75
GREENWRINKLE	6	2.40	6.25

Chi-Square Test for Specified Proportions	
Chi-Square	8.9724
DF	3
Pr > ChiSq	0.0297

Sample Size = 250

- The p-value for the chi-square test is less than 0.05, which leads us to reject the null hypothesis and conclude there is evidence to conclude that the peas do not come from a population having the 9:3:3:1 phenotypic ratios.

Proc Freq Output Data Sets

- PROC FREQ produces two types of output data sets that you can use with other statistical and reporting procedures. You can request these data sets as follows:

- Specify the `OUT=` option in a TABLES statement. This creates an output data set that contains frequency or crosstabulation table counts and percentages

- Specify an OUTPUT statement. This creates an output data set that contains statistics. More details can be found in SAS documentation.

Analyzing Two-Way Tables

- Recall: To create a cross-tabulation table using PROC FREQ for relating two variables, use the TABLES statement with both variables listed and separated by an asterisk (*), (for example, A * B).

- A cross-tabulation table is formed by counting the number of occurrences in a sample across two grouping variables. The number of columns in a table is usually denoted by c and the number of rows by r. Thus, a table is said to have r x c cells.

- Using Proc Freq, we can conduct a test of independence between two variables:

H_0: The variables are independent (no association between them).

H_a: The variables are not independent.

Analyzing Two-Way Tables

- Example: Data come from a study performed by Karl Pearson in 1909 involving the relationship between criminal behavior and drinking alcoholic beverages. The category “Coining” refers to counterfeiting. For the DRINKER variable, 1 means yes and 0 means no.

```
PROC FREQ DATA=DRINKERS;WEIGHT COUNT;
     TABLES DRINKER*CRIME/CHISQ;
TITLE 'Chi Square Analysis of a Contingency Table';
RUN;
```

- Results: The chi-square value is 48.7 and $p<0.001$. Thus, you reject the null hypothesis of no association (independence) and conclude that there is evidence of a relationship between drinking status and type of crime committed.

Merging/ Concatenating Data Sets

- Concatenating/ Appending data sets with SET statement:
  - Appending data sets combines records from two or more data sets.
  - For example, suppose data are collected at two locations and you want to combine the data sets into one SAS data set for analysis.
  - The data sets must have at least some of the same variables in common.
  - Appending is accomplished by including multiple data set names in the SET statement.
  - Example:

```
DATA NEW;
  SET OLD1 OLD2;
RUN;
```

creates the data set NEW, which consists of the records in the OLD1 data set as well as the records from the OLD2 data set.

Merging/ Concatenating Data Sets

- Merging data sets with MERGE statement:
  - We can merge datasets by adding new variables from one data set to an existing data set.
  - To accomplish this, you must have a **matching identifier** in each data set that can be used by SAS to match the records during the merge.
  - For example, suppose you have data collected on patients taken on two separate occasions, and the data for each collection time are in separate data sets. If you want to combine the data so you can compare pre- and post-treatment values, the technique for merging the data sets using some key identifier (such as patient ID) is:

    1. Sort each data set by the key identifier.

    2. Within a DATA step, use the MERGE statement along with a BY statement to merge the data by the key identifier.

Merging/ Concatenating Data Sets

- Example:

Pre-Treatment Data

Obs	case	pretreat
1	1	1.02
2	2	2.10
3	4	2.20
4	5	1.44
5	3	1.88
6	11	1.55
7	13	1.61
8	14	2.61
9	15	1.56
10	16	0.99
11	22	1.53

Post-Treatment Data

Obs	case	posttreat
1	1	1.94
2	2	1.63
3	3	2.73
4	4	2.18
5	5	1.82
6	13	2.25
7	11	1.94
8	14	1.70
9	15	1.78
10	16	1.52
11	22	1.97

(Notice the sort for Case variable – We need to sort both datasets first.)

```
proc sort data=pre; by case; run;
proc sort data=post; by case; run;

data prepost;
  merge pre post;
  by case;
  diff = posttreat - pretreat;
run;
```

- Merged dataset:

Merged Pre- and Post- Treatment Data

Obs	case	pretreat	posttreat	diff
1	1	1.02	1.94	0.92
2	2	2.10	1.63	-0.47
3	3	1.88	2.73	0.85
4	4	2.20	2.18	-0.02
5	5	1.44	1.82	0.38
6	11	1.55	1.94	0.39
7	13	1.61	2.25	0.64
8	14	2.61	1.70	-0.91
9	15	1.56	1.78	0.22
10	16	0.99	1.52	0.53
11	22	1.53	1.97	0.44

Merging/ Concatenating Data Sets

- Merging data sets with MERGE statement:
  - When datasets are merged using the MERGE statement in a DATA step, a given record in one input dataset may not have corresponding counterparts with matching BY variable values in the other input datasets.
  - However, the DATA step merge selects both records with matching BY variable values *as well as* nonmatching records from any input dataset.
  - Any variables occurring only in datasets having no records to contribute for a given BY group will simply be missing.

Merging/ Concatenating Data Sets

- Merging data sets with MERGE statement:
  - To perform matching only for matching records in the DATA step, a common approach is to use the `IN=` dataset option in conjunction with a subsetting IF statement as follows:

```
data mergeddata;
  merge olddata1(in=xxx) olddata2;
  by commonvar;
  if xxx;
run;
```

  - Here, the `IN=` option creates a temporary variable that indicates whether the corresponding dataset contributed to the current observation. We can specify any valid SAS variable name, but here we chose the name xxx.
  - Any records with a value of COMMONVAR variable that did not appear in the OLDDATA1 dataset will have a value of 0 for xxx. Those observations will fail the subsetting IF statement and will not be written to the output dataset.
  - The resulting output dataset, MERGEDDATA, will have only the matching records desired.

Summarizing Longitudinal Data

- A longitudinal study is a research design that involves repeated observations of the same variables over long periods of time.
- Longitudinal data (sometimes called panel data) is data that is collected through a series of repeated observations of the same subjects over some extended time frame.
- This type of data is especially useful for measuring change over time.
- See VITALS dataset as an example.

Transposing Data Sets: PROC TRANSPOSE

- Proc Transpose allows you to restructure the values in a data set by transposing/ re-orienting the data.
- Syntax:

```
PROC TRANSPOSE DATA=Dataset-name OUT=New-dataset-name;
    BY variable(s);
    ID variable;
    VAR variable(s);
RUN;
```

- Example:

```
Proc Transpose Data=vitals out=vital_tr;
  by subjid;
  var sbp;
  id visit;
run;
```

Transposing Data Sets: PROC TRANSPOSE

- The OUT keyword says that the transposed dataset should be created as a new dataset called Vital_tr.
- The BY statement is used to determine the row structure of the transposed dataset. You can include more than one variable in the BY statement. Your data must be sorted on your BY variables before running PROC TRANSPOSE.
  - For long-to-wide transposes, the BY variable(s) should uniquely identify each row.
  - For wide-to-long transposes, the BY variable(s) determine the row structure of the long data; that is, it determines the repetition of the rows.

Transposing Data Sets: PROC TRANSPOSE

- The ID statement assigns names to the transposed value columns that match the values in the variable listed in the ID statement.
  - For long-to-wide transposes, the ID variable(s) determine the structure of the columns in the transposed dataset. There will be one column for each unique value of the ID variable (or if multiple ID variables are present, one column for each unique combination of values).
  - For wide-to-long transposes, you typically do not need an ID variable.
- The VAR statement is where you actually tell SAS what variables you want transposed. These are the values that will appear in the cells of the transposed variables.
  - For long-to-wide datasets, there is usually one variable in the VAR statement.
  - For wide-to-long datasets, there are usually multiple variables in the VAR statement. The resulting dataset will have one row for each variable identified in the VAR statement.