EFFECTIVE SAS SKILLS

Meng Wang
09/23/2015
Graduate Statistics Club
About SAS

- SAS is an integrated system of software solutions that enables you to perform the following tasks:
  - data entry, retrieval, and management
  - report writing and graphics design
  - statistical and mathematical analysis
  - business forecasting and decision support
  - operations research and project management
  - applications development

- SAS includes DATA steps and PROC steps.
  - DATA steps: prepare SAS data sets.
    - observations in rows, variables in columns
  - PROC steps: data analysis and forecasting. (>300 procedures)
Table of contents

1. Introduction about the type of data.
2. Preparing data for analysis.
   PROC FORMAT, Data steps, Macro
3. Creating report, printing and saving data.
   PROC PRINT, Data step, Data _null_
Introduction about the type of data

• The paired sample of binary data is very common in clinical trials studies.
• Two measurements are paired when they come from the same or matched units. And there exist a natural link between these two measurements. Pairing is determined by a study’s design.
• Pairing is to reduce variability. After the pairing, the between-subject variability will be eliminated.
• Two tests:
  • McNemar’s test for proportions from paired sample (assess the significance of the difference between two correlated proportions)
  • CMH test for stratified two-sample binomial data
• So, we need to prepare the data with “right” format!
Data set as example

• 86 patients participated in a study of the effect of drug for reducing bilirubin. They were treated with the drug. The clinical laboratory results of pre- and post-study were examined, indicating the total bilirubin to be normal or abnormal (above a certain level).

• The data is a matched-pair study, i.e., based on their characteristics (age or gender), the subjects are matched.

The goal of dealing with the sample data sets

### Original data:

<table>
<thead>
<tr>
<th>subject</th>
<th>V1</th>
<th>V2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>31</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>32</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>33</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>34</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>35</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>37</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>38</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>39</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>41</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>42</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>43</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>44</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>45</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>46</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>47</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>48</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>49</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>50</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>51</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>52</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>53</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>54</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>55</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>56</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>57</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>58</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>59</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>60</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>61</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>62</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>63</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>64</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>65</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>66</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>67</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>68</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>69</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>70</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>71</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>72</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>73</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>74</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>75</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>76</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>77</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>78</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>79</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>80</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>81</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>82</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>83</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>84</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>85</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>86</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

### What we want:

**Format 1:**

<table>
<thead>
<tr>
<th>subject</th>
<th>V1</th>
<th>V2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Format 2:**

<table>
<thead>
<tr>
<th>Subject visit</th>
<th>bili</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 1</td>
<td>0</td>
</tr>
<tr>
<td>1 2</td>
<td>0</td>
</tr>
<tr>
<td>2 1</td>
<td>0</td>
</tr>
<tr>
<td>2 2</td>
<td>0</td>
</tr>
<tr>
<td>3 1</td>
<td>0</td>
</tr>
<tr>
<td>3 2</td>
<td>0</td>
</tr>
</tbody>
</table>

**With format:**

- For V1, V2:
  - 0 = 'Normal'
  - 1 = 'Abnormal'

- For visit:
  - 0 = 'Normal'
  - 1 = 'Abnormal'

- For bili:
  - 1 = 'Pre'
  - 2 = 'Post'
Setting the SAS environment

dm "output;clear;log;clear";
options ls=75 ps=100 formdlim='*' nodate nonumber nocenter;

*display manager: to clear output and log windows*;
*system options: useful option for report;
*ls: line size(numbers of col in output window);
*ps: page size;
*formdlim='*': specifies a character to delimit page breaks in SAS output;
*nodate & nonumber: date and page number are not included;
*nocenter: output left justified;
More about of DM command

Syntax
DM <window> ‘command(s)’<window><CONTINUE>;

Some useful display manager commands which work in any window include:
• clear - clear the contents of the window
• end - close the window
• endsas - end the sas session
• file "filename" - save contents of the window to filename
• prevcmd - recall previous display manager command
More about OPTIONS statement

How long the system option settings are in effect:

data one;
  set items;
run;
/*/ option applies to all subsequent steps */
options obs=5;
/*/ printing ends with the fifth observation */
proc print data=one;
  run;
/*/ To read more than five observations, you must reset the OBS= system option.*/
PROC FORMAT

General form:

- PROC FORMAT *library=libref.catalogname* ;
  VALUE  *numfmt value1='formatted-value-1' value2='formatted-value-2'*
  ........ valuen='formatted-value-n' ;
  VALUE  *$charfmt value1='formatted-value-1' value2='formatted-value-2'*
  ........ 'valuen='formatted-value-n' ;
RUN;

Without the LIBRARY=option, formats are stored in a catalog called FORMATS in the temporary WORK library and exist only for the duration of the SAS session. If the LIBRARY= option specifies only a libref, formats are permanently stored in that library in a catalog called FORMATS.

$ is included in the length of the name
Reading data using double trailing at signs @@

- @@ appears at the end of the INPUT statement.
- It tells SAS rather than advancing to a new record, hold the current input record for the execution of the next INPUT statement, even across iterations of the DATA step.
Generate a report

*Control the general format;
proc print data=bili1 noobs split='*' n;
label subject='Patient Number*============'
      V1='Pre-test*============'
      V2='Post-test*============';
title 'Bilirubin Abnormalities Following Drug Treatment';
run;

*creating a pdf file;
ods pdf file='your_file.pdf';
*...SAS programs...;
ods pdf close;

/*1. noobs: suppresses the observation number in the output.
/*2. split= : Specify the split character, which controls line breaks in column headings.
/*3. n: shows the number of observations in the report.
Generate a report

*Control the general format;
proc print data=bili1 noobs split='*' n;
label subject='Patient Number*============'
    V1='Pre-test*=========='
    V2='Post-test*============';
title 'Bilirubin Abnormalities Following Drug Treatment';
run;

*creating a pdf file;
ods pdf file='your_file.pdf';
*...SAS programs...;
ods pdf close;

/*1. noobs: suppresses the observation number in the output.
/*2. split= : Specify the split character, which controls line breaks in column headings.
/*3. n: shows the number of observations in the report.
* Setting environment;
  dm "output;clear;log;clear";
  options ls=75 ps=2000 formdlim='*' nodate nonumber nocenter;

* retrieve the data set bili and present it in an alternative format;
  libname mydrive 'C:\Users\Meng\SAS_seminar';

proc format;
  value bilifmt 0 = 'Normal'
               1 = 'Abnormal';
  value visitfmt 1 = 'Pre'
                 2 = 'Post';
run;
More about libname statement

• Syntax:
  libname libref 'path';
• e.g. libname mydrive 'C:\Users\Meng\SAS_seminar';
  data analysis;
    set mydrive.my_data;
  run;

• Note:
  Libname statement is to tell SAS where to read/write data, like a short-cut.
  The file is stored in the libref, but the file name is not included in the ‘path’.
*** use the original data input to get format 2;
data bili2;
format subject 2.0 visit visitfmt. bili bilifmt.;
input subject @;
do visit = 1 to 2;
  input bili @@;
  output;
end;
datalines;
1 0 0 2 0 0 3 0 0 4 0 0 5 0 0 6 0 1
7 1 1 8 0 0 9 0 0 10 0 0 11 0 0 12 1 0
...
;
@@ vs @

- The double trailing @@ holds a record in the input buffer across multiple iterations of the DATA step until the end of the record is reached.
- The single trailing @ releases a record when SAS returns to the top of the DATA step to process the next iteration.
Changing from Format 1 to Format 2

/* Method 1: the use of data-step */
data bili3_1 (keep = subject visit bili);
  set mydrive.bili1;
  visit = 1; bili=V1;
  output; *to create a record in the SAS data set;
  visit = 2; bili=V2;
  output;
run;

proc print data=bili3_1 noobs;
  format subject 2.0 visit visitfmt. bili bilifmt.;
run;

/*When you create more than one data set in a single DATA step, using the data set options enables you to drop or keep different variables in each of the new data sets. A DROP or KEEP statement, on the other hand, affects all of the data sets that are created. */
Changing from Format 1 to Format 2

/* Method 2: the use of macro, when you have several visits*/
%macro reform; /* write a macro function called 'reform';
data bili3_2 (keep = subject visit bili); %
  set mydrive.bili1; %
%do v = 1 %to 2; /* a macro variable &v. will be created whose vales are 1 to 2;
    visit = &v.; /* use &v. to assign the value of the variable 'treatment';
    bili=V&v.; /* Now the vales of the two responses are in V1 and V2;
    output; /* create one record for each response of the subjects;
  %end;
run;
%mend; /*end of a macro function;

%reform; /* run the macro function 'reform';

proc print data=bili3_2;
run;
Use DATA _NULL_ to write a report rather than create a data set

PUT statement in DATA step:

- Writes lines to the SAS log, to the SAS procedure output file, or to an external file that is specified in the most recent FILE statement;
- Syntax: PUT <specification(s)><_ODS_><@|@@>;

- Specification(s): specifies what is written, how it is written and where it is written. Including:
  - Variable(-list); Format(-list); -L/-C/-R: left aligns/centers/right aligns the value;

- Column pointer controls: @: move the pointer to column n
- Line pointer controls: #

E.g.:   put @12 x dollar7.2-c;
Uses the dollar7.2 format and centers the value of X starting at column 12:
$100.00
Use DATA _NULL_ to write a report rather than create a data set

data _null_; ** do not store the result as a SAS data set;**
   set bili3_2 end=lastrecord;
/*end = variable: defines a temporary variable whose value is 1 when the data step is processing the last observation; At all other times, the value of variable is 0;*/
/* use the 'by' statement to group the data
   SAS will create two automatic variables: first.### and last.### */
   by subject;
   first = first.subject;
   last = last.subject;

file print; * tell SAS to put the results on the output window; Specifies the current output file for PUT statements.

title1 "Example 2: Alternative Representation";
   title2 "";
Use DATA _NULL_ to write a report rather than create a data set

```sas
if _N_ = 1 then
do;
    put @1 "Patient No." @15 "Visit" @25 "T. Bilirubin;"
    put 35*"-";
end;
** (1) _N_: gives the number of the record;** (2) @#: tells SAS to move to Position #;
if first then *record = first;
    put @1 subject 8.0 @15 visit visitfmt. @25 bili bilifmt.; *if the first record of the subject;

if last then
do;
    put @15 visit visitfmt. @25 bili bilifmt.;
    put 35*"-";
end;

if lastrecord then
    put "Total No. of subjects: " subject;
run;
```

**Example 2: Alternative Representation**

<table>
<thead>
<tr>
<th>Patient No.</th>
<th>Visit</th>
<th>T. Bilirubin</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pre</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>Post</td>
<td>Normal</td>
</tr>
<tr>
<td>2</td>
<td>Pre</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>Post</td>
<td>Normal</td>
</tr>
<tr>
<td>3</td>
<td>Pre</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>Post</td>
<td>Normal</td>
</tr>
<tr>
<td>4</td>
<td>Pre</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>Post</td>
<td>Normal</td>
</tr>
<tr>
<td>86</td>
<td>Pre</td>
<td>Normal</td>
</tr>
<tr>
<td></td>
<td>Post</td>
<td>Normal</td>
</tr>
</tbody>
</table>

Total No. of subjects: 80
Changing from Format 2 to Format 1

/* separate the responses from two visits into two datasets, we will merge them later */
data bili4V1 bili4V2;
  set mydrive.bili2;  *subject, visit, bili;
  if visit = 1 then output bili4V1;
  if visit = 2 then output bili4V2;
run;

proc print data=bili4V1;
run;
Changing from Format 2 to Format 1 - PROC SQL

*Merge using proc sql;
proc sql;  **creating a table from query expression;
create table bili4_1 as
  select bili4V1.subject,
         bili4V1.bili as V1,
         bili4V2.bili as V2
  from bili4V1 left join bili4V2
  on bili4V1.subject=bili4V2.subject
  order by subject;
quit;

proc print data=bili4_1 noobs label;
  label V1='Pre'
         V2='Post';
  format subject 2.0 V1 V2 bilifmt.;
run;
Changing from Format 2 to Format 1 – DATA STEP

/** we may also use merge in data-step, but we need to change the variable names (and sort the data) in the two data sets that we are going to merge. We do both steps together */
proc sort data=bili4V1 out=bili4V1(rename=(bili=V1));
   by subject;
run;

proc sort data=bili4V2 out=bili4V2(rename=(bili=V2));
   by subject;
run;

data bili4_2 (keep=subject V1 V2);
   merge bili4V1 bili4V2;
      by subject;
run;
McNemar's test

/* McNemar's test: approximated chi-square distribution;
*Note: the first ods (output delivery system) statement helps to
suppress the output for the simple kappa coefficient, which is a
measure of interrater agreement.*/

ods exclude SimpleKappa; *(this statement is optional);
proc freq data=bili1;
  table V1*V2/agree norow nocol;
  title "McNemar's test with the chi-square approximation";
run;

* (optional) SAS also provides an exact test for the McNemar's test;
ods exclude CrossTabFreqs SimpleKappa;
proc freq data=bili1;
  table V1*V2;
  exact mcnemar;
  title "Exact test for McNemar's test";
run;
** CMH test **

``` Sas
** stratified 2-by-2 table;proc freq data=bili2;
table subject*visit*bili/norow nocol nopercent;
run;

*ods trace on;
ods select cmh;
proc freq data=bili2;
table subject*visit*bili/cmh;
title "Example 2: CMH test";
run;
*ods trace off;

data mydrive.bili2;
set bili2;
run;
```

<table>
<thead>
<tr>
<th>visit</th>
<th>Normal</th>
<th>Abnormal</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Post</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>visit</th>
<th>Normal</th>
<th>Abnormal</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Post</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>2</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>
Summary

• McNemar’s test and the Cochran-Mantel-Haenszel test can be used to compare two correlated proportions.

• When implementing PROC FREQ to test a sample dataset, several scenarios of data set preparation: Format 1, Format 2, from Format 1 to Format 2 and From format 2 to Format 1 were discussed.

• Useful skills about data-step, macro and PROC SQL were introduced.

• PROC PRINT and data-step skills for saving data, as well as providing report were introduced.
Thank you!

Thank Dr. Ming-Hung (Jason) Kao for helping preparing the materials!