An Easy Route to a Missing Data Report with ODS+PROC FREQ+A Data Step

Mike Zdeb, University at Albany School of Public Health, Rensselaer, NY

INTRODUCTION

- ONE OF FIRST STEPS IN DATA ANALYSIS ... A DECISION ON HOW TO HANDLE MISSING VALUES

  DELETE OBSERVATIONS WITH EXCESS MISSING DATA

  DELETE VARIABLES WITH EXCESS MISSING DATA

  SUBSTITUTE IMPUTED VALUES FOR MISSING VALUES

  TAKE NO ACTION AT ALL IF THE AMOUNT OF MISSING DATA IS INSIGNIFICANT AND NOT LIKELY TO AFFECT THE ANALYSIS

- DECISION FACILITATED BY KNOWING THE AMOUNT OF MISSING DATA

- USE AN ODS OUTPUT STATEMENT, PROC FREQ, AND SOME DATA STEP PROGRAMMING

  PRODUCE A MISSING DATA REPORT SHOWING THE PERCENTAGE OF MISSING DATA FOR EACH VARIABLE IN A DATA SET

  IDENTIFY AND DROP ALL VARIABLES/OBSERVATIONS WITH EITHER ALL OR A HIGH PERCENTAGE OF MISSING VALUES

MAKE SOME DATA FOR EXAMPLES

- USE SASHELP.CLASS ...

```
* EXAMPLE 1;
data class;
set sashelp.class;
if ranuni(987) le .5;
if ranuni(987) le .2 then
   call missing(weight);
if ranuni(987) le .1 then
   call missing(sex, age);
run;
```

Obs	Name	Sex	Age	Height	Weight
1	Alice		.	56.5	84.0
2	Jane	F	12	59.8	84.5
3	Janet	F	15	62.5	112.5
4	John	M	12	59.0	.
5	Joyce	F	11	51.3	50.5
6	Judy	F	14	64.3	90.0
7	Mary	F	15	66.5	112.0
8	Philip	M	16	72.0	.
9	Robert		.	64.8	.
10	Thomas	M	11	57.5	85.0

REVIEW

- PROC FREQ DEFAULT

```
proc freq data=class;
run;
```

Sex	Frequency	Percent	Cumulative Frequency	Cumulative Percent
F	5	62.50	5	62.50
M	3	37.50	8	100.00

Frequency Missing = 2

- TABLES OF ALL VARIABLES
- COUNTS (NO PERCENTAGES) OF MISSING DATA

- ADD A TABLES STATEMENT AND THE MISSING OPTION ...

```
proc freq data=class;
tables _all_ / missing;
run;
```

Sex	Frequency	Percent	Cumulative Frequency	Cumulative Percent
	2	20.00	2	20.00
F	5	50.00	7	70.00
M	3	30.00	10	100.00

- COUNTS AND PERCENTAGES OF MISSING DATA
- BETTER IF NON-MISSING DATA ALL IN ONE GROUP
- USE A FORMAT TO CREATE GROUPS OF MISSING AND NON-MISSING DATA

```
proc format;
value nm    low-high = 'OK'       other = 'MISSING';
value $ch   ' ' = 'MISSING'       other = 'OK';
run;
```

- LOW-HIGH in NM FORMAT AVOIDS HAVING TO SPECIFY THE RANGE OF ALL POSSIBLE MISSING NUMERIC VALUES ... REMEMBER THERE ARE 27 OF THEM (._ , . , .A through .Z)

```
proc freq data=class;
tables _all_ / missing;
format _numeric_ nm.  _character_ $ch.;
run;
```

Sex	Frequency	Percent	Cumulative Frequency	Cumulative Percent
MISSING	2	20.00	2	20.00
OK	8	80.00	10	100.00

- OK FOR A DATA SET WITH ONLY FEW VARIABLES

- WRITE RESULTS TO A DATA SET ...

```
proc freq data=class;
tables _all_ / noprint missing out=tables;
format _numeric_ nm.   _character_ $ch.;
run;
```

FORMATTED

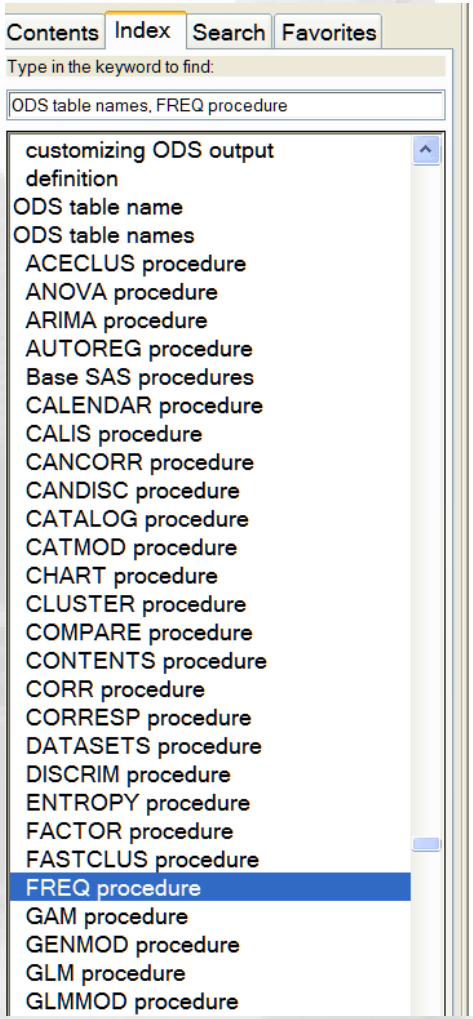
Obs	Weight	COUNT	PERCENT
1	MISSING	OK	OK
2	OK	OK	OK

UNFORMATTED

Obs	Weight	COUNT	PERCENT
1	.	3	30
2	50.5	7	70

- DOES NOT WORK ... ONLY THE LAST VARIABLE IS IN THE OUTPUT DATA SET (WEIGHT)

ALTERNATIVE … ODS OUTPUT



Measures	Measures of association	MEASURES
MHChiSq	Mantel-Haenszel chi-square exact test	MHCHI
MHChiSqMC	Monte Carlo exact test for Mantel-Haenszel chi-square	MHCHI / MC
NLevels	Number of variable levels	NLEVELS
OddsRatioCL	Exact confidence limits for the odds ratio	OR (2x2 tables)
OneWayChiSq	One-way chi-square test	CHISQ (one-way tables)
OneWayChiSqMC	Monte Carlo exact test for one-way chi-square	CHISQ / MC (one-way tables)
OneWayFreqs	One-way frequencies	(One-way table request)
OverallKappa	Overall simple kappa coefficient	AGREE (hx2x2 tables)
OverallKappas	Overall kappa coefficients	AGREE (hxrxr tables, r>2)
PdiffEquiv	Equivalence analysis for the proportion difference	RISKDIFF(EQUIV) (2x2 tables)
PdiffEquivLimits	Equivalence limits for the proportion difference	RISKDIFF(EQUIV) (2x2 tables)
PdiffEquivTest	Equivalence test for the proportion difference	RISKDIFF(EQUIV) (2x2 tables)
PdiffNoninf	Noninferiority test for the proportion difference	RISKDIFF(NONINF) (2x2 tables)
PdiffSup	Superiority test for the proportion difference	RISKDIFF(SUP) (2x2 tables)
PdiffTest	Proportion difference test	RISKDIFF(EQUAL) (2x2 tables)
PearsonChiSq	Pearson chi-square exact test	PCHI
PearsonChiSqMC	Monte Carlo exact test for Pearson chi-square exact test	PCHI / MC
PearsonCorr	Pearson correlation	PCORR
PearsonCorrMC	Monte Carlo exact test for Pearson correlation	PCORR / MC

CREATE DATA SET ... TABLES

```
proc format;
value nm     low-high = '1'     other = '0';
value $ch    ' ' = '0'          other = '1';
run;

ods listing close;
ods output onewayfreqs=tables;

proc freq data=class;
tables _all_ / missing;
format _numeric_ nm.  _character_ $ch.;
run;

ods output close;
ods listing;
```

- TABLE OF ALL VARIABLES

- "F_" VARIABLES HAVE FORMATTED VALUES (0, 1)

Table	F_Name	Name	Frequency	Percent	F_Sex	Sex	F_Age	Age	F_Height	Height	F_Weight	Weight
Table Name	1	Alice	10	100				.		.		.
Table Sex			2	20	0			.		.		.
Table Sex			8	80	1	F		.		.		.
Table Age			2	20			0	.		.		.
Table Age			8	80			1	11		.		.
Table Height			10	100				.	1	51.3		.
Table Weight			3	30				.		.	0	.
Table Weight			7	70				.		.	1	50.5

CHANGE ODS OUTPUT STATEMENT ...

ods output onewayfreqs=tables (keep=table frequency percent f_:);

Table	F_Name	Frequency	Percent	F_Sex	F_Age	F_Height	F_Weight
Table Name	1	10	100.00				
Table Sex		2	20.00	0			
Table Sex		8	80.00	1			
Table Age		2	20.00		0		
Table Age		8	80.00		1		
Table Height		10	100.00			1	
Table Weight		3	30.00				0
Table Weight		7	70.00				1

ADD A DATA STEP

Table	F_Name	Frequency	Percent	F_Sex
Table Name	1	10	100.00	
Table Sex		2	20.00	0
Table Sex		8	80.00	1

```
data report;
length var $32;
do until (last.table);
      set tables;
      by table notsorted;
      array names(*) f_:  ;
      select (names(_n_));
            when ('0') do; miss = frequency; p_miss = percent; end;
            when ('1') do; ok = frequency;  p_ok = percent; end;
      end;
end;
```

```
miss = coalesce(miss,0)          /* change missing  values  to zero */
pmiss = coalesce(pmiss, 0);
ok = coalesce(ok, 0);
p_ok = coalesce(p_ok, 0);
var  = scan(table,-1);

format miss ok comma7. p_: 5.1 ;
label     miss  = 'N_MISSING'
          ok  = 'N_OK'
          p_miss = '%_MISSING'
          p_ok  = '%_OK'
          var = 'VARIABLE';
keep var miss ok p_:  ;
run;
```

- A MISSING DATA REPORT ...

VARIABLE	N_MISSING	%_MISSING	N_OK	%_OK
Name	0	0.0	10	100.0
Sex	2	20.0	8	80.0
Age	2	20.0	8	80.0
Height	0	0.0	10	100.0
Weight	3	30.0	7	70.0

- TRY SAME SAS CODE WITH ANOTHER DATA SET ... SASHELP.HEART

VARIABLE	N_MISSING	%_MISSING	N_OK	%_OK
Status	0	0.0	5,209	100.0
DeathCause	3,218	61.8	1,991	38.2
AgeCHDdiag	3,760	72.2	1,449	27.8
Sex	0	0.0	5,209	100.0
AgeAtStart	0	0.0	5,209	100.0
Height	6	0.1	5,203	99.9
Weight	6	0.1	5,203	99.9
Diastolic	0	0.0	5,209	100.0
Systolic	0	0.0	5,209	100.0
MRW	6	0.1	5,203	99.9
Smoking	36	0.7	5,173	99.3
AgeAtDeath	3,218	61.8	1,991	38.2
Cholesterol	152	2.9	5,057	97.1
Chol_Status	152	2.9	5,057	97.1
BP_Status	0	0.0	5,209	100.0
Weight_Status	6	0.1	5,203	99.9
Smoking_Status	36	0.7	5,173	99.3

- THE SAS CODE IS "GENERIC"
- THE ONLY CHANGE IS THE NAME OF THE DATA SET

ADD VARIABLE LABELS

- CHANGE THE LENGTH STATEMENT AT THE TOP OF THE DATA STEP ...

```
length var $32 label $100;
```

- ADD A STATEMENT ...

```
var  = scan(table,-1);
* new statement;
label  = vlabelx(var);
keep var label miss ok p_:  ;
```

VARIABLE	LABEL	N_MISSING	%_MISSING	N_OK	%_OK
Status	Status	0	0.0	5,209	100.0
DeathCause	Cause of Death	3,218	61.8	1,991	38.2
AgeCHDdiag	Age CHD Diagnosed	3,760	72.2	1,449	27.8
Sex	Sex	0	0.0	5,209	100.0
AgeAtStart	Age at Start	0	0.0	5,209	100.0
Height	Height	6	0.1	5,203	99.9
Weight	Weight	6	0.1	5,203	99.9
Diastolic	Diastolic	0	0.0	5,209	100.0
Systolic	Systolic	0	0.0	5,209	100.0
MRW	Metropolitan Relative Weight	6	0.1	5,203	99.9
Smoking	Smoking	36	0.7	5,173	99.3
AgeAtDeath	Age at Death	3,218	61.8	1,991	38.2
Cholesterol	Cholesterol	152	2.9	5,057	97.1
Chol_Status	Cholesterol Status	152	2.9	5,057	97.1
BP_Status	Blood Pressure Status	0	0.0	5,209	100.0
Weight_Status	Weight Status	6	0.1	5,203	99.9
Smoking_Status	Smoking Status	36	0.7	5,173	99.3

MISSING DATA PATTERNS

- USE PROC MI ... NUMERIC VARIABLES

```
proc mi data=class;
ods select misspattern;
run;
```

Missing Data Patterns

Group	Age	Height	Weight	Freq	Percent	Group Means		
						Age	Height	Weight
1	X	X	X	6	60.00	13.000000	60.316667	89.083333
2	X	X	.	2	20.00	14.000000	65.500000	.
3	.	X	X	1	10.00	.	56.500000	84.000000
4	.	X	.	1	10.00	.	64.800000	.

USE PROC FREQ WITH A LIST OPTION ... ALL VARS

```
proc format;
value nm low-high = 'X'
                other = '.';
 value $ch ' ' = '.'
                  other = 'X';
run;
```

Name	Sex	Age	Height	Weight	Frequency	Percent
X	.	.	X	.	1	10.00
X	.	.	X	X	1	10.00
X	X	X	X	.	2	20.00
X	X	X	X	X	6	60.00

```
proc freq data=class;
table Name*Sex*Age*Height*Weight / list missing nocum;
format _numeric_ nm. _character_ $ch.;
run;
```

- USE PROC SQL IF THERE ARE LOTS OF VARIABLES ...

```
proc sql noprint;
select name into :vars separated by '*'
from dictionary.columns
where libname eq 'WORK' and memname eq 'CLASS' ;
quit;

proc freq data=class;
tables &vars / list missing nocum out=miss;
format _numeric_ nm. _character_ $ch.;
run;
```

- MACRO VARIABLE ... Name*Sex*Age*Height*Weight

- DROP VARIABLES WITH EXCESS MISSING DATA

- USE VALUES FROM THE MISSING DATA REPORT ...

```
proc sql noprint;
select var into :droplist separated by ' '
from report
where p_miss ge 50;
quit;

data heart;
set sashelp.heart(drop=&droplist);
run;
```

(data set SASHELP.HEART ... DeathCause, AgeCHDDiag, AgeAtDeath)

- DROP OBSERVATIONS WITH TOO MANY MISSING VALUES ...

```
data heart;
set sashelp.heart;
if cmiss (of _all_) ge 5 then delete;
run;
```

- CMISS FUNCTION ALLOWS BOTH CHARACTER AND NUMERIC VARIABLES (NMISS ... ONLY NUMERIC)

DROP OBSERVATIONS WITH ALL MISSING VALUES ...

```
options missing=' ';        /* CATS with numeric data */

data new;
set old;
if missing(cats(of _all_)) then delete;
run;

options missing='.';
```

string	Name	Sex	Age	Height	Weight
Jane.59.884.5	Jane		.	59.8	84.5
JudyF1464.390	Judy	F	14	64.3	90.0
JanetF1562.5112.5	Janet	F	15	62.5	112.5
ThomasM1157.5.	Thomas	M	11	57.5	.
JamesM1257.383	James	M	12	57.3	83.0
JohnM125999.5	John	M	12	59.0	99.5
JeffreyM1362.584	Jeffrey	M	13	62.5	84.0
AlfredM1469.	Alfred	M	14	69.0	.
Henry.63.5.	Henry		.	63.5	.
WilliamM1566.5112	William	M	15	66.5	112.0

NLEVELS

- PROC FREQ WITH NLEVELS OPTION ... A QUICK WAY TO DROP VARIABLES WITH ALL MISSING VALUES

- SOME DATA ...

```
data test;
input name :  $10.  gender  :  $1.  age  height weight;
datalines;
MIKE    .     21   60   .L
MARY    .     .Z   54   .H
MARK    .     45   10   .H
;
```

- NUMBER OF UNIQUE VALUES FOR EACH VARIABLE

TableVar	NLevels	NMissLevels	NNonMissLevels
name	3	0	3
gender	1	1	0
age	3	1	2
height	3	0	3
weight	2	2	0

```
ods output nlevels=tables;

proc freq data=test nlevels;
run;

ods output close;
```

WHAT IF NO MISSING DATA

```
data test;
input name : $10. gender : $1. age height weight;
datalines;
MIKE M 21 72 200
MARY F 56 54 120
MARK M 45 60 110
;
```

TableVar	NLevels
name	3
gender	2
age	3
height	3
weight	3

```
proc format;
value nm      ._ - .z  = .     other = 'OK' ;
value $ch     ' '  = ' '       other = 'OK';
run;

ods listing close;
ods output nlevels=tables;

proc freq data=test nlevels;
ods select nlevels;
format _numeric_ nm.   _character_ $ch.;
run;

ods output close;
ods listing;
```

TableVar	NLevels	NMissLevels	NNonMissLevels
name	3	0	3
gender	1	1	0
age	3	1	2
height	3	0	3
weight	2	2	0

* error in SQL if no missing data;

```
proc sql noprint;
select tablevar into :droplist separated by ' '
from tables
where nnonmisslevels eq 0;
quit;

data new;
set test (drop=&droplist);
run;
```

nnonmisslevels	TableVar	NLevels	NMissLevels
1	name	1	0
0	gender	1	1
1	age	2	1
0	height	1	1
0	weight	1	1

REPLACE NUMERIC MISSING WITH 0

```
data test;
input name : $10. gender : $1. age height weight;
datalines;
MIKE     .     21   60   .L
MARY     .     .Z   54   .H
MARK     .     45   10   .H
;

data test;
set test;
array nm(*) _numeric_;
do _n_ = 1 to dim(nm);
  nm(_n_) + 0;
end;
run;
```

name	gender	age	height	weight
MIKE		21	60	0
MARY		0	54	0
MARK		45	10	0

- THERE IS A SHORTER WAY

```
proc stdize data=test out=new reponly missing=0;
run;
```

- CAN USE SAME NAME FOR OUT= ... WITH ALL VALUES MISSING YOU GET A WARNING IN THE LOG THAT HAS NOTHING TO WITH THE VALUE REPLACEMENT

WHAT VALUES WERE REPLACED

```
data test;
input name :  $10.  gender  :  $1.  age  height weight;
datalines;
MIKE      .      21   60   .L
MARY     .       0     .     .H
MARK     .      45   10    0
;

proc format;
value fix low-high='0' other='1';
run;
```

```
filename nosee dummy;

data test;
file nosee;
set test;
put (_numeric_) (fix.) @;
missing = _file_;
put;
run;
```

name	gender	age	height	weight	missing
MIKE		21	60	L	001
MARY		0	.	H	011
MARK		45	10	0	000

```
proc stdize data=test out=new reponly missing=0;
run;
```

name	gender	age	height	weight	missing
MIKE		21	60	0	001
MARY		0	0	0	011
MARK		45	10	0	000

CONTACT INFORMATION

Mike Zdeb
msz03@albany.edu
518 402 6479