2. Simple procedures

Use of SAS
January 2011

Contents

- transformation and selection
- descriptive procedures
- sorting data

Example O'Neill et.al. (1983):

Lung function for 25 patients with cystic fibrosis.

Table 12.11 Data for 25 patients with cystic fibrosis (O'Neill *et al.*, 1983)

Sub	Age	Sex	Height	Weight	BMP	FEV_1	RV	FRC	TLC	PEmax
1	7	0	109	13.1	68	32	258	183	137	95
2	7	1	112	12.9	65	19	449	245	134	85
3	8	0	124	14.1	64	22	441	268	147	100
4	8	1	125	16.2	67	41	234	146	124	85
5	8	0	127	21.5	93	52	202	131	104	95
6	9	0	130	17.5	68	44	308	155	118	80
7	11	1	139	30.7	89	28	305	179	119	65
8	12	1	150	28.4	69	18	369	198	103	110
9	12	0	146	25.1	67	24	312	194	128	70
10	13	1	155	31.5	68	23	413	225	136	95
11	13	0	156	39.9	89	39	206	142	95	110
12	14	1	153	42.1	90	26	253	191	121	90
13	14	0	160	45.6	93	45	174	139	108	100
14	15	1	158	51.2	93	45	158	124	90	80
15	16	1	160	35.9	66	31	302	133	101	134
16	17	1	153	34.8	70	29	204	118	120	134
17	17	0	174	44.7	70	49	187	104	103	165
18	17	1	176	60.1	92	29	188	129	130	120
19	17	0	171	42.6	69	38	172	130	103	130
20	19	1	156	37.2	72	21	216	119	81	85
21	19	0	174	54.6	86	37	184	118	101	85
22	20	0	178	64.0	86	34	225	148	135	160
23	23	0	180	73.8	97	57	171	108	98	165
24	23	0	175	51.1	71	33	224	131	113	95
25	23	0	179	71.5	95	52	225	127	101	195

Data has been read into the sas-dataset:

`p:\sas\data\other\pemax.sas7bdat`

Definition of new variables

We want to study body mass index, `bmi`.

```
libname mysas 'p:\sas\data\other';

data temp;      /*temporary data set*/
 set mysas.pemax;
 bmi=weight/(height/100)**2;
run;

proc print data=temp;
run;
```

Obs	age	sex	height	weight	fev1	pemax	bmi
1	7	1	109	13.1	32	95	11.0260
2	7	2	112	12.9	19	85	10.2838
3	8	1	124	14.1	22	100	9.1701
.	.	.	.	.	.	.	.

Transformations

- **Arithmetics**
  - The usual operators: + - * /
  - Raising to a power: **, e.g.. `x**2`
  - Square root: `sqrt(x)`
  - Logarithms: `log(x), log10(x), log2(x)`
- Relations:

  ```
  =  <  >  <= >= <> (unequal)
  eq lt gt le ge ne (alternative notation)
  ```

- Logical operators:

  ```
  and   or   not
  ```

Other types of variable definitions

```
data mysas.pemax;
  set mysas.pemax; /*original file changed*/

if sex=1 then csex='m';
if sex=2 then csex='f';

if 0<bmi<12 then fat=1;
if 12<=bmi<18 then fat=2;
if 18<=bmi then fat=3;
run;

proc print data=mysas.pemax;
  var csex age bmi fat;
run;
```

Obs	csex	age	bmi	fat
1	f	7	11.0260	1
2	m	7	10.2838	1
.	.	.	.	.
.	.	.	.	.
14	m	15	20.5095	3

If I do not want to change 'mysas.pemax', I can make a new data set:

```
data temp;
  set mysas.pemax;

if sex=1 then csex='m';
if sex=2 then csex='f';

if 0<bmi<12 then fat=1;
if 12<=bmi<18 then fat=2;
if 18<=bmi then fat=3;
run;

proc print data=temp;
  var csex age bmi fat;
run;
```

Obs	csex	age	bmi	fat
1	f	7	11.0260	1
2	m	7	10.2838	1
.	.	.	.	.
.	.	.	.	.
14	m	15	20.5095	3

Measures of location, centre

- Average

$$\bar{x} = \frac{1}{n}(x_1 + \cdots + x_n)$$

  - can be interpreted as the centre of gravity
  - is heavily influenced by outlying observations

- Median
  - the observation in 'the middle'
  - is not influenced by outlying observations (**robust**)

Measures of variation

- Variance, standard deviation

$$s^2 = \frac{1}{n-1}\Sigma(x_i - \bar{x})^2$$

standard deviation $= \sqrt{\textbf{variance}}$

- Quantiles
  - median: 50% quantile
  - quartiles: 25%, 50% and 75% quantiles

Calculation of summary statistics in SAS

```
proc means data=temp;
run;
```

```
The MEANS Procedure

Variable    N            Mean         Std Dev         Minimum         Maximum
------------------------------------------------------------------------------
age        25      14.4800000       5.0589854       7.0000000      23.0000000
sex        25       1.4400000       0.5066228       1.0000000       2.0000000
fev1       25      34.7200000      11.1971723      18.0000000      57.0000000
pemax      25     109.1200000      33.4369058      65.0000000     195.0000000
bmi        25      15.3422331       3.8633242       9.1701353      22.7777778
------------------------------------------------------------------------------
```

These are **default**,

others may be chosen as **options**

Options in PROC MEANS

```
/*Some of the keywords available with PROC MEANS:
      N - number of observations
      MEAN - mean value
      MIN - minimum value
      MAX - maximum value
      SUM - total of values
      NMISS - number of missing values
      MAXDEC=n - set maximum number of decimal places */

statistic-keyword(s) specifies which statistics to compute
and the order to display them in the output.
The available keywords in the PROC statement are

 Descriptive statistic keywords
 CLM RANGE CSS SKEWNESS|SKEW CV STDDEV|STD KURTOSIS|KURT STDERR
 LCLM SUM MAX SUMWGT MEAN UCLM MIN USS N VAR NMISS

Quantile statistic keywords
    MEDIAN|P50 Q3|P75 P1 P90 P5 P95 P10 P99 Q1|P25 QRANGE
```

If we want to see the medians:

```
proc means data=temp median;
  var age bmi fev1;
run;

The MEANS Procedure

Variable          Median
------------------------
age           14.0000000
bmi           14.8660771
fev1          33.0000000
------------------------
```

Oops: Now, we got **only** the median!

```
proc means data=temp N mean median;
  var age bmi fev1;
run;

The MEANS Procedure

Variable     N            Mean          Median
---------------------------------------------
age         25      14.4800000      14.0000000
bmi         25      15.3422331      14.8660771
fev1        25      34.7200000      33.0000000
---------------------------------------------
```

Alternative procedure: UNIVARIATE

```
proc univariate data=mysas.pemax;
  var bmi;
run;
```

a lot of output...

```
The UNIVARIATE Procedure
Variable:  bmi

                          Moments

N                          25    Sum Weights               25
Mean                15.3422331    Sum Observations    383.555827
Std Deviation       3.86332415    Variance            14.9252735
Skewness            0.27214922    Kurtosis            -0.7599282
Uncorrected SS      6242.80947    Corrected SS        358.206564
Coeff Variation     25.1809768    Std Error Mean      0.77266483

              Basic Statistical Measures

    Location                    Variability

Mean     15.34223     Std Deviation            3.86332
Median   14.86608     Variance                14.92527
Mode        .         Range                   13.60764
                      Interquartile Range      5.36231
```

Tests for Location: Mu0=0

Test	-Statistic-		-----p Value------	
Student's t	t	19.85626	Pr > |t|	<.0001
Sign	M	12.5	Pr >= |M|	<.0001
Signed Rank	S	162.5	Pr >= |S|	<.0001

Quantiles (Definition 5)

Quantile	Estimate
100% Max	22.77778
99%	22.77778
95%	22.31516
90%	20.50953
75% Q3	17.98454
50% Median	14.86608
25% Q1	12.62222
10%	10.35503
5%	10.28380
1%	9.17014
0% Min	9.17014

Extreme Observations

------Lowest------		-----Highest-----	
Value	Obs	Value	Obs
9.17014	3	19.4021	18
10.28380	2	20.1995	22
10.35503	6	20.5095	14
10.36800	4	22.3152	25
11.02601	1	22.7778	23

Categorical variables: PROC FREQ

Can be used to illustrate distributions of categorical variables, e.g.

```
proc freq data=temp;
  tables csex sex fat;
run;
```

The FREQ Procedure

csex	Frequency	Percent	Cumulative Frequency	Cumulative Percent
f	11	44.00	11	44.00
m	14	56.00	25	100.00

sex	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	14	56.00	14	56.00
2	11	44.00	25	100.00

fat	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	6	24.00	6	24.00
2	13	52.00	19	76.00
3	6	24.00	25	100.00

Filtering data - selecting subsets

- In `DATA`-step
  - regarding observations: `WHERE, IF, DELETE`
  - regarding variables: `DROP, KEEP`
- In procedures
  - Regarding observations: `WHERE`
  - Regarding variables: `VAR`-statement
    (depending on procedure)

How to use WHERE

If we only want to look at the girls:

```
data temp1;                  /* temporary data set */
 set mysas.pemax;
 where csex='f';
run;

proc print data=temp1;
  var csex age bmi;
run;

Obs     csex     age      bmi

  1      f         7    10.2838
  2      f         8    10.3680
  3      f        11    15.8894
  4      f        12    12.6222
  ..............
```

Alternative ways of writing,

using **IF** or **DELETE**

Look out, if data contains missing values.

```
data temp2;
 set mysas.pemax;
 if csex='f';
run;

----------------------------------------

data temp3;
 set mysas.pemax;
 if csex ne 'm';
run;

----------------------------------------

data temp4;
 set mysas.pemax;
 if csex='m' then delete;
run;
```

temp2 includes only girls,

temp3, temp4 includes grils *and* children with information on 'csex'.

Filterings may be combined:

```
data temp5;
 set mysas.pemax;
 where csex='f' and age>12;
run;

proc print data=temp;
  var csex age bmi;
run;
```

Obs	csex	age	bmi
1	f	13	13.1113
2	f	14	17.9845
3	f	15	20.5095
4	f	16	14.0234
5	f	17	14.8661
6	f	17	19.4021
7	f	19	15.2860

Note: Observations with age missing are included with:

```
data pemax6;
set mysas.pemax;
where csex='f' and age<12;
run;
```

How to use DROP and KEEP

Now that we have `bmi`,
we may not need `height` and `weight`:

The most efficient is to simply delete them:

```
data temp7;
 set mysas.pemax;
 drop height weight;
run;
```

You can also specify which variables you want to keep:

```
data temp8;
 set mysas.pemax;
 keep bmi csex;
run;
```

Using WHERE in procedures

If we for a specific procedure only want to look at the girls (but continue to work with all data):

```
proc print data=mysas.pemax;
  where csex='f';
  var csex age bmi;
run;
```

```
Obs     csex     age      bmi

  1       f        7    10.2838
  2       f        8    10.3680
  3       f       11    15.8894
  4       f       12    12.6222
  ............
```

Sorting of data

- Often used because other procedures require sorted data
- Example

```
proc sort data=account;
   by town descending debt;
run;
```

- data are sorted by `town` (small values first).
- for each value of `town` data are sorted by `debt` (high values first - because of descending statement).

BY statement

- Used in many procedures (MEANS, REG, GLM, ...)
- Runs analyses within groups
- Requires that data are sorted
- Remember to get rid of missings, or they will form a separate group
- Example

```
proc sort data=new.fitness;
  by group;
run;
proc means data=new.fitness;
  var maxpulse;
  by group;
run;
```

Output

```
------------------------------------ Experimental group=0 ------------------------------------

                                The MEANS Procedure

                    Analysis Variable : maxpulse Maximum heart rate

          N            Mean         Std Dev         Minimum         Maximum
         10     177.5000000       6.1689185     168.0000000     186.0000000


------------------------------------ Experimental group=1 ------------------------------------

                    Analysis Variable : maxpulse Maximum heart rate

          N            Mean         Std Dev         Minimum         Maximum
         10     172.7000000       9.1899704     164.0000000     192.0000000


------------------------------------ Experimental group=2 ------------------------------------

                    Analysis Variable : maxpulse Maximum heart rate

          N            Mean         Std Dev         Minimum         Maximum
         11     171.3636364      10.9660634     155.0000000     188.0000000
```

The Juul data set

Anders Juul et al., Dep. GR, Rigshosp.

Serum IGF-I (Insulin-like Growth Factor) reference data set

Age	N	Source
0–5	44	Circumcision, hernia operation
5–20	833	4 schools in the Copenhagen area
20+	153	Hospital staff

AGE	age
MENARCHE	1. mentrual period occurred (1/2, 2 for yes)
SEXNR	1 for boys, 2 for girls
SIGF1	Serum IGF-I
TANNER	Puberty stage a.m. Tanner (1–5)
TESTVOL	Testicular volume

Exercise: Simple procedures

In your p-drive, you can find the file `'\juul2.sas7bdat'` containing the Juul data with variables

- Age (years)
- Height (cm)
- Menarche (No/Yes: 1/2)
- Sexnr (M/F: 1/2)
- Serum IGF1, growth hormone (μg/ml)
- Tanner stage (1–5)
- Testis volume (ml)
- Weight (kg)

1. Read the data it into SAS using a libname statement.
2. Calculate means and standard deviations
3. Use PROC FREQ to determine the distribution of the categorical variables
4. Make a new variable giving the BMI for each person
5. Determine the BMI distribution in each of the Tanner stages (e.g. using a BY-statement).