SAS course

March, 2009

Reading data into SAS

from file
data lines directly in program
import from, e.g., Excel

Special considerations

character variables
data seperation
missing values

Lung function: 25 patients with cystic fibrosis.

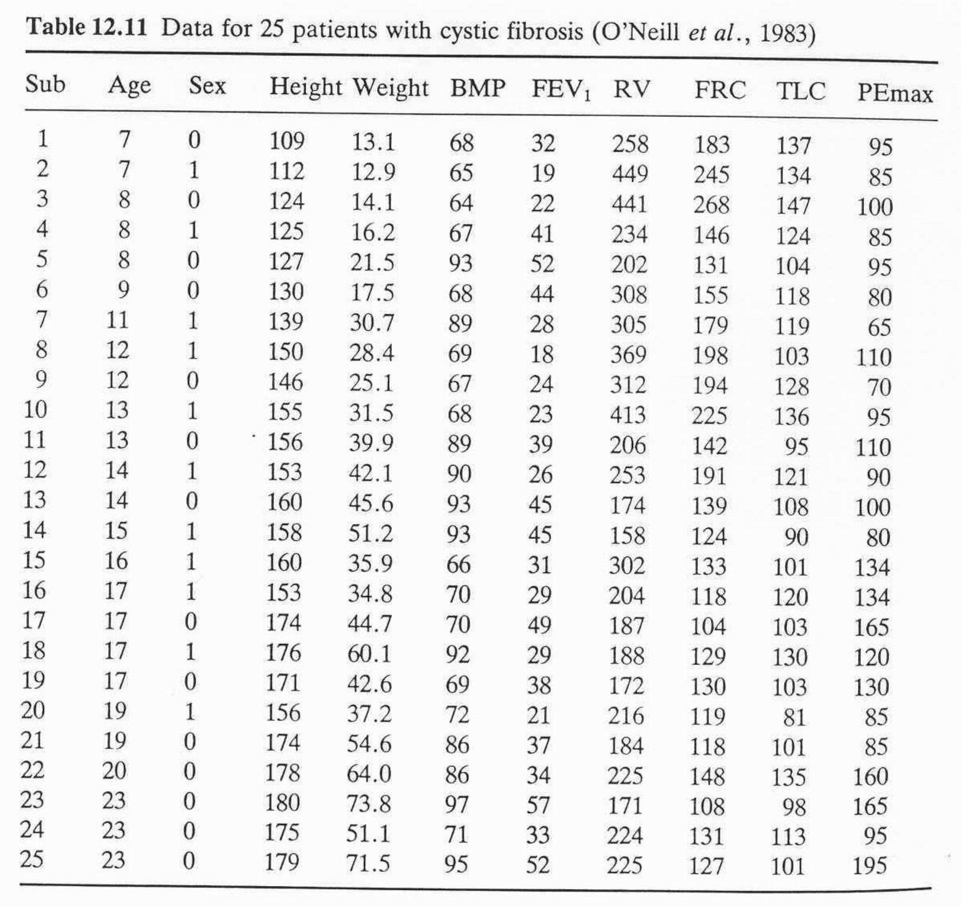

Table 12.11 Data for 25 patients with cystic fibrosis (O'Neill *et al*., 1983)

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

Some of these data are in the text file `pemax.txt` on the T-drive

```
age sex height weight fev1 pemax
  7 1 109 13.1 32  95
  7 2 112 12.9 19  85
  8 1 124 14.1 22 100
  8 2 125 16.2 41  85
  8 1 127 21.5 52  95
  9 1 130 17.5 44  80

  : :   :    :  :   :

 19 2 156 37.2 21  85
 19 1 174 54.6 37  85
 20 1 178 64.0 34 160
 23 1 180 73.8 57 165
 23 1 175 51.1 33  95
 23 1 179 71.5 52 195
```

Reading in data from a text file

In the program editor, we write the program lines:

```
data sasuser.pemax;
  infile ’T:\pemax.txt’ firstobs=2;
  input age sex height weight fev1 pemax;
run;
```

Note: the option `firstobs=2` tells SAS that line 2 is the first line that contains data.

```
NOTE: 25 records were read from the infile ’pemax.txt’.
      The minimum record length was 21.
      The maximum record length was 21.
NOTE: The data set SASUSER.PEMAX has 25 observations and 6
variables. NOTE: DATA statement used:
      real time           0.11 seconds
      cpu time            0.01 seconds
```

We now have the permanent SAS data set ’pemax’ in the ’sasuser’ library.

Data lines directly in program

```
data sasuser.pemax;
 input age sex height weight fev1 pemax;
 datalines;
  7 1 109 13.1 32  95
  7 2 112 12.9 19  85
  8 1 124 14.1 22 100

 23 1 179 71.5 52 195
 ;
run;
```

Reading in character variables

```
data sasuser.pemax;
 input age sex $ height weight fev1 pemax;
 datalines;
  7 male   109 13.1 32  95
  7 female 112 12.9 19  85
  8 male   124 14.1 22 100

 23 male   179 71.5 52 195
 ;
run;
```

Include '$' after each character variable.

Semicolon separated data

Until now, data have been nicely separated by blanks. Now it looks a bit different.....

```
age;sex;height;weight;fev1;pemax
7;male,109;13.1;32;95
7;female;112;12.9;19;85 8;male;124;14.1;22;100
8;female;125;16.2,41;85 ..... .....
```

```
data sasuser.pemax;
  /* we now specify a list of possible delimiters*/
  infile 'T:\pemax2.txt' firstobs=2 dlm=';,';
  input age sex $ height weight fev1 pemax;
run;
```

Using 'dlm' both comma and semicolon are regarded as delimiters. Period is not a good delimiter. Why??

Formatted input

Now, the values are not separated at all! This is often useful for many binary observations, e.g. questionnaire data.

```
 7M10913.132 95
 7F11212.919 85
 8M12414.122100

23M17971.552195
```

SAS code:

```
data sasuser.pemax;
 infile 'T:\pemax3.txt';
 input age 1-2 sex $ 3 height 4-6 weight 7-10 fev1 11-12 pemax 13-15;
run;
```

Missing values

Should be coded using '.' (period)

When looking at data-files take care if you see 9,-9, 99, 999 etc.

example:

```
data sasuser.pemax;
 input age sex height weight fev1 pemax;
 datalines;
  7 1 . 13.1 32  95
  7 2 112 12.9 19  85
  8 . 124 14.1 22 100
  .....

  .....
 ;
run;
```

Exercise: Reading in some 'ugly' data.

In the file `'orig_juul.txt'` (P-drive, juul directory) we have the original data from Anders Juul's investigation of growth hormone. Data are comma separated and appears in the following order:

`age bmi genital height hsds hv igfbp3 mammae menarche pubestan sex sigf1 tanner testvol weight`

1. Read in the data into SAS
2. Check, that you have 1340 observations and 15 variables
3. We do not want to use the variables **`hsds, hv, pubestan, mammae`** and `genital`. Omit these from the data set.
4. Compute summary statistics: mean, median, number of missing observations, minimum and maximum.
5. Are there any missing values?
6. How are they represented?
7. Are there any strange values, which might actually be missing values? If so, make them into proper missing values, and calculate the summary statistics once more. Compare to the previous results.

Files from external programs

In general when you have files from Excel, SPSS or other programs we recommend the program *StatTransfer*, which can be used for converting almost any data-file to a SAS data set.

Alternatively, most programs will allow you to print data in a text file. This file can then be read into SAS using the previosly described methods.

Importing Excel sheets

A data set with information about adverse events from a clinical trial is in `ae.xls` on the P-drive

```
proc import out=work.adverse datafile= "P:\ae.XLS" dbms=excel replace;
     getnames=yes;
run;
```

Files from external programs

Can often be handled using 'Import Wizard'

1. Select File → Import Data
2. Select type of file to import
3. Specify where to put the generated sas-dataset (e.g. WORK)
4. Save automatically generated SAS code (PROC IMPORT)