SAS Input and Output
A collection demonstrating commonly used features.

SAS contains a very extensive system for importing and exporting data. These are well described in the manuals, but the mere extent of the documentation is for many a barrier for using the facilities. In the present collection of SAS programs we have presented some useful examples that may be useful for newcomers.

Many of the input tasks may be solved using the graphical user interface (GUI) provided by the SAS version you are using. Many others simply by copy-paste operations from the data- or program source you are using. However, data are often not organized in such a way that it may be directly used as input for the relevant SAS procedures. Many of the programs presented here are addressing this problem, and the solutions are independent of the SAS version you are using.

The examples here are run using SAS Enterprise under Windows, but almost none of the specific facilities of that version is used. The most important exception is the 'compiler' generated coloring of the SAS program: data are shaded yellow, statements blue, comments green etc. These colors are kept here since they are found to be useful in structuring the programs.

S.E.&O. Knut Conradsen, August 2017.

Contents

Read one observation/line.

By observation we here mean all data corresponding to one observational unit, in the example corresponding values of flight number, temperature, and thermal distress. We address the problem of reading data with one observation per dataline.

Data

The data are documenting the presence or absence of primary O-ring thermal distress in the 23 shuttle launches preceding the Challenger mission. The focus of these data is to determine if there is a relationship between the temperature at launch time and o-ring thermal distress. The variables in the data set are the flight number (FLT), the temperature at launch (TEMP), and an indicator variable for whether or not there was thermal distress during the launch (TD) (0=no distress, 1=distress).
Source: SAS Institute Inc.

Flight number	1	2	3	4	5	6	7	8	9	10	11	12
Temperature	66	70	69	68	67	72	73	70	57	63	70	78
Thermal distress	0	1	0	0	0	0	0	0	1	1	1	0

Flight number	13	14	15	16	17	18	19	20	21	22	23
Temperature	67	53	67	75	70	81	76	79	75	76	58
Thermal distress	0	1	0	0	0	0	0	0	1	0	1

Program

```
data Challenger;
input flt temp td;
datalines;
1  66  0
2  70  1
3  69  0
4  68  0
5  67  0
6  72  0
7  73  0
8  70  0
9  57  1
10  63  1
11  70  1
12  78  0
13  67  0
14  53  1
15  67  0
16  75  0
17  70  0
18  81  0
19  76  0
20  79  0
21  75  1
22  76  0
23  58  1
;
run;
```

```
Title 'The o-ring data from the Challenger missions';
proc print data=Challenger;
run;
```

Output

The o-ring data from the Challenger missions

Obs	flt	temp	td
1	1	66	0
2	2	70	1
3	3	69	0
4	4	68	0
5	5	67	0
6	6	72	0
7	7	73	0
8	8	70	0
9	9	57	1
10	10	63	1
11	11	70	1
12	12	78	0
13	13	67	0
14	14	53	1
15	15	67	0
16	16	75	0
17	17	70	0
18	18	81	0
19	19	76	0
20	20	79	0
21	21	75	1
22	22	76	0
23	23	58	1

■

Read several observations/line

When each dataline contains several observations, for instance data on several individuals, we must use the double at-sign (@@) (called a double trailing in SAS) is to hold the line for multiple iterations of the data step. In the example below, each line contains values of eight variabkles for two individuals except for the last line, which only contains data for one individual.

Program

```
data fitness;
input Age Weight Oxygen RunTime RestPulse RunPulse MaxPulse @@;
datalines;
44 89.47 44.609 11.37 62 178 182 40 75.07 45.313 10.07 62 185 185
44 85.84 54.297  8.65 45 156 168 42 68.15 59.571  8.17 40 166 172
38 89.02 49.874  9.22 55 178 180 47 77.45 44.811 11.63 58 176 176
40 75.98 45.681 11.95 70 176 180 43 81.19 49.091 10.85 64 162 170
44 81.42 39.442 13.08 63 174 176 38 81.87 60.055  8.63 48 170 186
44 73.03 50.541 10.13 45 168 168 45 87.66 37.388 14.03 56 186 192
45 66.45 44.754 11.12 51 176 176 47 79.15 47.273 10.60 47 162 164
54 83.12 51.855 10.33 50 166 170 49 81.42 49.156  8.95 44 180 185
51 69.63 40.836 10.95 57 168 172 51 77.91 46.672 10.00 48 162 168
48 91.63 46.774 10.25 48 162 164 49 73.37 50.388 10.08 67 168 168
57 73.37 39.407 12.63 58 174 176 54 79.38 46.080 11.17 62 156 165
52 76.32 45.441  9.63 48 164 166 50 70.87 54.625  8.92 48 146 155
51 67.25 45.118 11.08 48 172 172 54 91.63 39.203 12.88 44 168 172
51 73.71 45.790 10.47 59 186 188 57 59.08 50.545  9.93 49 148 155
49 76.32 48.673  9.40 56 186 188 48 61.24 47.920 11.50 52 170 176
52 82.78 47.467 10.50 53 170 172
;
run;

Title "The fitness data set";
proc print data=fitness;
run;
```

Partial Output

The fitness data set

Obs	Age	Weight	Oxygen	RunTime	RestPulse	RunPulse	MaxPulse
1	44	89.47	44.609	11.37	62	178	182
2	40	75.07	45.313	10.07	62	185	185
3	44	85.84	54.297	8.65	45	156	168
4	42	68.15	59.571	8.17	40	166	172
5	38	89.02	49.874	9.22	55	178	180
6	47	77.45	44.811	11.63	58	176	176
⋮							
29	49	76.32	48.673	9.40	56	186	188
30	48	61.24	47.920	11.50	52	170	176
31	52	82.78	47.467	10.50	53	170	172

■

Read type=corr data

In many cases the original data will not be available, only the sufficient statistics (assuming Gaussian distributions), i.e. empirical (sample versions of) means, variances, and covariances, or functions thereof: standard deviations and correlations. Many analyses may of cause be performed using those (e.g. principal components), but it requires special input statements using the type=cov or type=corr option depending on whether we have covariances or correlations.

Program

```
/*Program for reading number of observations, means, standard deviations
and correlations into the correct data type*/

data cement(type=corr);
infile datalines missover;
input _type_ $ _name_ $ BLAINE XSO3 LSR STRGTH3 STRGTH7 STRGTH28 ;
datalines;
N        nobs       198 198 198 198 198 198
MEAN .            3095.71717 1.94126 40.29268 263.89394 382.28788 515.27778
STD    .           234.22485 0.25448 8.55410 36.88332 36.70451 32.67124
CORR BLAINE       1
CORR XSO3          0.49077  1
CORR LSR           -0.19604 -0.52069  1
CORR STRGTH3     0.80042  0.47398 -0.17769  1
CORR STRGTH7     0.73643  0.37344 -0.07175  0.89557  1
CORR STRGTH28   0.51413  0.27931 -0.08501  0.60826  0.74561  1
;

Title 'Cement Data';
proc print data=cement;
run;
```

Output

Cement Data

Obs	_type_	_name_	BLAINE	XSO3	LSR	STRGTH3	STRGTH7	STRGTH28
1	N	nobs	198.00	198.000	198.000	198.000	198.000	198.000
2	MEAN		3095.72	1.941	40.293	263.894	382.288	515.278
3	STD		234.22	0.254	8.554	36.883	36.705	32.671
4	CORR	BLAINE	1.00	.	.	.	.	.
5	CORR	XSO3	0.49	1.000	.	.	.	.
6	CORR	LSR	-0.20	-0.521	1.000	.	.	.
7	CORR	STRGTH3	0.80	0.474	-0.178	1.000	.	.
8	CORR	STRGTH7	0.74	0.373	-0.072	0.896	1.000	.
9	CORR	STRGTH28	0.51	0.279	-0.085	0.608	0.746	1.000

■

Reading external SAS dataset

The permanent SAS dataset o_ring.sas7bdat is stored in "C:\Users\knco\Documents\Multstat-2014". It is imported and placed into the temporary dataset "Challenger". The data are the same as in the section ‘Read one observation/line’.

When using SAS University Edition, the dataset must be placed in the ‘myfolder’ that was created during installation or in a subfolder to ‘myfolder’. This is the only folder on the harddrive that SAS UE can access!

Program

```
LIBNAME alfa "C:\Users\knco\Documents\Multstat-2014";
data Challenger;
set alfa.o_ring;
run;

Title 'The o-ring data from the Challenger missions';
proc print data=Challenger;
run;
```

Output

The o-ring data from the Challenger missions

Obs	flt	temp	td
1	1	66	0
2	2	70	1
3	3	69	0
4	4	68	0
5	5	67	0
6	6	72	0
7	7	73	0
8	8	70	0
9	9	57	1
10	10	63	1
11	11	70	1
12	12	78	0
13	13	67	0
14	14	53	1
15	15	67	0
16	16	75	0
17	17	70	0
18	18	81	0
19	19	76	0
20	20	79	0
21	21	75	1
22	22	76	0
23	23	58	1

Storing data in permanent SAS datasets

There are two types of SAS datasets: temporary and permanent. The temporary SAS datasets only exist during the current SAS session whereas permanent SAS datasets are saved to a location on the computer and exist after exiting SAS. SAS creates a library called SASUSER thet may be used for storing permanent datasets, but it is also possible to define other locations the computer e.g. by using the LIBNAME statement or directly.

When using SAS University Edition, the location must be in the 'myfolder' that was created during installation or in a subfolder to 'myfolder'. This is the only folder on the harddrive that SAS UE can access!

Program

```
/*Reading the dataset we shall analyze*/
proc import
  datafile="C:\Users\knco\Documents\Multstat2017\CamillaData.csv"
  out=FeatureData dbms=dlm replace;
delimiter=",";
getnames=yes;
guessingrows=200;
run;

/*Generating a new temporary dataset Feature*/
data Feature;
set FeatureData;
x=sqrt(abs(slope));
drop Slope Inter DeltaX;
run;

/*Storing the Feature dataset as a permanent dataset sasuser.Feature in the
SAS library SASUSER*/
data sasuser.Feature;
set Feature;
run;

/*Storing the Feature dataset as a permanent dataset Feature1.sas7bdat in the folder
"Multistat2017" using the LIBNAME statement. The LIBNAME statement performs two tasks:
It creates and names a new SAS library, and it links the new SAS library to a directory on the
computer.*/
LIBNAME alfa "C:\Users\knco\Documents\Multstat2017";
data alfa.Feature1;
set Feature;
run;

/*Storing directly in the Feature2.sas7bdat dataset*/
data "C:\Users\knco\Documents\Multstat2017\Feature2.sas7bdat";
```

```
set Feature;
run;

Title 'The first 10, 8, 6, and 4 observations from the Feature dataset addressed in four ways';
proc print data=sasuser.Feature (Obs=10);run;
Title; /*Clears the Title statement so that we only get one heading*/
proc print data=alfa.Feature1 (Obs=8);
proc print data="C:\Users\knco\Documents\Multstat2017\Feature1.sas7bdat" (Obs=6);
proc print data="C:\Users\knco\Documents\Multstat2017\Feature2.sas7bdat" (Obs=4);
run;
```

Output

The first 10, 8, 6, and 4 observations from the Feature dataset addressed in four ways

Obs	Piece	Pos	Sample	Wavelength	x
1	1	P2	1	500	0.028666
2	1	P2	2	500	0.028720
3	1	P3	3	500	0.028794
4	1	P3	4	500	0.028737
5	1	P1	5	500	0.028961
6	1	P1	6	500	0.028854
7	1	P1	7	500	0.028972
8	1	P1	8	500	0.028973
9	1	P1	9	500	0.028607
10	1	P1	10	500	0.028683

Obs	Piece	Pos	Sample	Wavelength	x
1	1	P2	1	500	0.028666
2	1	P2	2	500	0.028720
3	1	P3	3	500	0.028794
4	1	P3	4	500	0.028737
5	1	P1	5	500	0.028961
6	1	P1	6	500	0.028854
7	1	P1	7	500	0.028972
8	1	P1	8	500	0.028973

Obs	Piece	Pos	Sample	Wavelength	x
1	1	P2	1	500	0.028666
2	1	P2	2	500	0.028720
3	1	P3	3	500	0.028794
4	1	P3	4	500	0.028737
5	1	P1	5	500	0.028961
6	1	P1	6	500	0.028854

Obs	Piece	Pos	Sample	Wavelength	x
1	1	P2	1	500	0.028666

2	1	P2	2	500	0.028720
3	1	P3	3	500	0.028794
4	1	P3	4	500	0.028737

■

Read ‘one x - several y’ values per line

Program for reading two-dimensional observations with several, possibly different numbers of y-values for each x-value. For illustrative purposes we also plot corresponding values of x and y
The data are taken from: Fig. 6 in H. P. Hacker and H. S. Pearson (1944): The Growth, Survival, Wandering and Variation of the Long-Tailed Field Mouse, Apodemus Sylvaticus. Biometrika, Vol. 33, No. 2 (Aug., 1944), pp. 136-162.

Program

```
data apodemus;
infile cards missover;
input weight organ @;
do until (organ=.);
output;
input organ @;
end;
cards;
16.0  1.6
17.5  0.7
18.5  1.0  1.4
19.0  1.7  1.8
19.5  1.2  1.3  1.5
20.0  1.5  1.7  1.8  1.9  2.0
20.5  1.1  1.2  1.3  1.4  1.4  1.4  1.6  2.1
21.0  1.5  1.8  1.8  2.2
21.5  1.6  1.7  1.8  1.8  2.1
22.0  1.5  1.5  1.6  1.7  1.7  1.8  1.9  1.9  2.2  2.3  2.7
22.5  1.3  1.6  1.6  1.7  1.7  1.8  1.8  1.9  2.0  2.0  2.0  2.0  2.0  2.1  2.1  2.1  2.1  2.2  2.3  2.3  2.4
23.0  1.3  1.7  1.9  2.0  2.0  2.0  2.0  2.2  2.3  2.3  2.4  2.4  2.6
23.5  1.8  1.9  2.0  2.2  2.2  2.2  2.3  2.4
24.0  1.8  1.8  1.9  2.3  2.3  2.3  2.4  2.4  2.4  2.4  2.6  3.0
24.5  1.8  1.8  1.9  2.0  2.0  2.0  2.1  2.2  2.2  2.3  2.3  2.3  2.5
25.0  1.4  1.8  1.9  2.0  2.4  2.4
25.5  1.1  1.8  1.9  2.0  2.0  2.0  2.1  2.5  2.6  2.7
26.0  2.0  2.4  2.6  2.8
26.5  2.1  2.2  2.4  2.4  2.4  2.4  2.6  3.0
27.0  2.2  2.3  2.4  2.6  2.9
27.5  2.2  2.4  2.4  2.4
28.0  2.6
29.0  2.4
29.5  2.7
;
run;

Title 'Mouse data, first 20 observations';
```

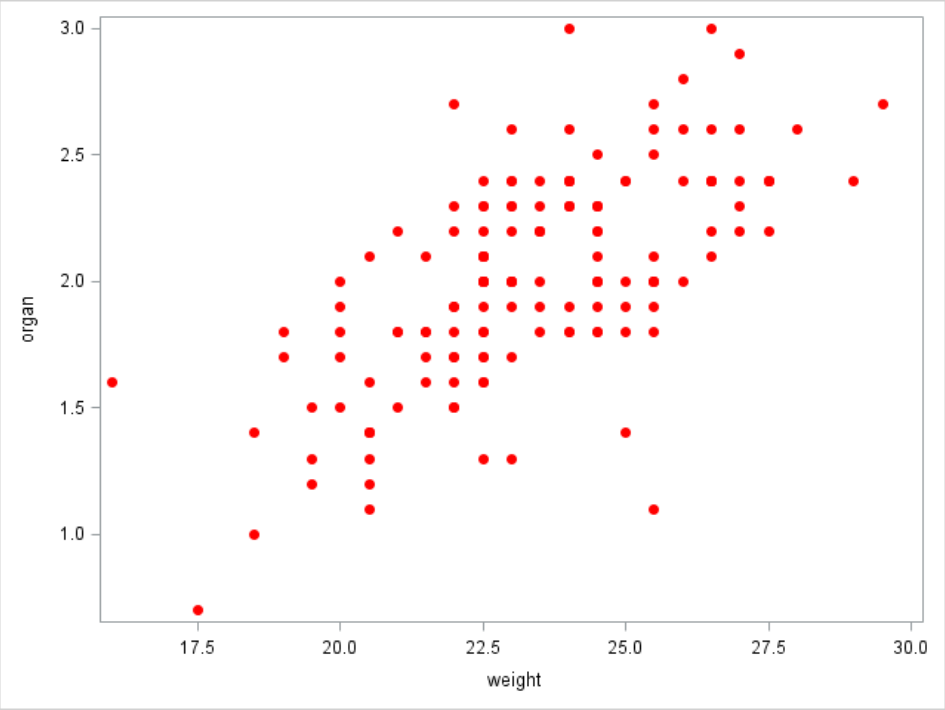
```
proc print data=apodemus (obs=20);
run;

proc sgplot data=apodemus;
  scatter x=weight y=organ/ markerattrs=(symbol=circlefilled color=red);
run;
```

Output

Mouse data, first 20 observations

Obs	weight	organ
1	16.0	1.6
2	17.5	0.7
3	18.5	1.0
4	18.5	1.4
5	19.0	1.7
6	19.0	1.8
7	19.5	1.2
8	19.5	1.3
9	19.5	1.5
10	20.0	1.5
11	20.0	1.7
12	20.0	1.8
13	20.0	1.9
14	20.0	2.0
15	20.5	1.1
16	20.5	1.2
17	20.5	1.3
18	20.5	1.4
19	20.5	1.4
20	20.5	1.4



Input One Way ANOVA

A. Balanced case.

We consider the corresponding values of lithology and spectral reflectance for 9 land plots (obtained using the multispectral scanner from the Landsat satellite)

Lithology	Spectral reflectance
LIT2	36, 43, 42
LIT5	44, 50, 30
LIT6	29, 33, 36

We want to analyze whether the reflectance mean levels may be considered equal using a one sided analysis of variance (One Way ANOVA). We copy the data from the table and paste them into the program reading the data.

Program 1.

```
/*Since the data are comma separated, we must include a dlm option in the infile statement.
The transition between cells are pasted as a tabulator, but since we only have one cell per
lithology we do not need to take this into account. An example where this does not work is gine
in the 'Input Two Way ANOVA' section.*/

data Landsat1;
infile datalines dlm = ',';
```

```
input lithology $ 1-4  @;
do index=1 to 3;
input reflect @;
output;
end;
datalines;
LIT2    36, 43, 42
LIT5    44, 50, 30
LIT6    29, 33, 36
;
run;

Title 'Landsat Data, Landsat1';
proc print data=Landsat1;
run;
```

Output

Landsat Data, Landsat1

Obs	lithology	index	reflect
1	LIT2	1	36
2	LIT2	2	43
3	LIT2	3	42
4	LIT5	1	44
5	LIT5	2	50
6	LIT5	3	30
7	LIT6	1	29
8	LIT6	2	33
9	LIT6	3	36

B. Unbalanced case.

We now consider the situation where we have different numbers of observations for each lithology value as shown in the table below.

Lithology	Spectral reflectance
LIT2	36, 43
LIT5	44, 50, 30
LIT6	29

Program 2

```
data Landsat2;
infile datalines dlm = ',' missover;
input lithology $ 1-4 reflect @;
/*The variable index is not necessary, but it is often useful as a pointer to the specific plot
number for each lithology.*/
index=1;
do until (reflect = .);
output;
index=index+1;
input reflect @;
end;
datalines;
LIT2    36, 43
LIT5    44, 50, 30
LIT6    29
;
run;

Title 'Reduced Landsat Data, Landsat2';
proc print data=Landsat2;
var lithology index reflect;
run;
```

Output

Reduced Landsat Data, Landsat2

Obs	lithology	index	reflect
1	LIT2	1	36
2	LIT2	2	43
3	LIT5	1	44
4	LIT5	2	50
5	LIT5	3	30
6	LIT6	1	29

■

Input Two Way ANOVA

We present three programs for reading data with a structure corresponding to balanced and unbalanced two sided ANOVAs and one for reading a balanced two sided MANOVA

Balanced case

		Crusher		
		1	2	3
Mixer	1	28, 52, -24, 80	-66, -60, 2, 120	-84, 18, 32, -40
	2	-58, 28, 58, -10	34, -12, -4, 120	-82, -20, -40, -52
	3	36, 116, 68, 50	72, -24, 62, 56	-54, -7, -32, 60

Table 1. Values (coded) of the strength of 36 concrete samples using three different mixers and three different crushers in a balanced, two-way layout.

Program 1

```
/*The input data is organized with one line of four strength
observations for each mixer-crusher combionation*/

data cement;
input mixer crusher @;
do index=1 to 4;
input strength @;
output;
end;
datalines;
1  1    28 52 -24 80
1  2   -66 -60 2 120
1  3   -84 18 32 -40
2  1   -58 28 58 -10
2  2   34 -12 -4 120
2  3 -82 -20 -40 -52
3  1   36 116 68 50
3  2    72 -24 62 56
3  3   -54 -7 -32 60
;
proc print data=cement;
run;
```

Program 2

```
/*The input data was copied from Table 1 and pasted into the program after 'datalines;'The
commas after the last observations in the six cells were added manually. When copying from a
table, there may be tabulator characters or similar must be removed from the pasted input.
This was not done here, only adding delimiter commas.*/

data cement0;
infile datalines dlm = ',';
input strength @@;
datalines;
28, 52, -24, 80,        -66, -60, 2, 120,       -84, 18, 32, -40
-58, 28, 58, -10,       34, -12, -4, 120,       -82, -20, -40, -52
36, 116, 68, 50,        72, -24, 62, 56,        -54, -7, -32, 60
;
run;

/*Now we generate the values of mixer, crusher and repetition number (index) by nested do
loops. Those values are then combined (merged) with the strength values read with the part
above.*/

data cement1;
do i=1 to 3;
  do j=1 to 3;
    do k=1 to 4;
      mixer=i;
      crusher=j;
      index=k;
      output;
    end;
  end;
end;
drop i j k;
run;

data cement;
merge cement0 cement1;
run;

Title 'The cement data with one observation/line';
proc print data=cement;
var mixer crusher index strength;
run;
```

Output, balanced case, identical for programs 1 and 2.

The cement data with one observation/line

Obs	mixer	crusher	index	strength
1	1	1	1	28
2	1	1	2	52
3	1	1	3	-24
4	1	1	4	80
5	1	2	1	-66
6	1	2	2	-60
7	1	2	3	2
8	1	2	4	120
9	1	3	1	-84
10	1	3	2	18
11	1	3	3	32
12	1	3	4	-40
13	2	1	1	-58
14	2	1	2	28
15	2	1	3	58
16	2	1	4	-10
17	2	2	1	34
18	2	2	2	-12
19	2	2	3	-4
20	2	2	4	120
21	2	3	1	-82
22	2	3	2	-20
23	2	3	3	-40
24	2	3	4	-52
25	3	1	1	36
26	3	1	2	116
27	3	1	3	68
28	3	1	4	50
29	3	2	1	72
30	3	2	2	-24
31	3	2	3	62
32	3	2	4	56
33	3	3	1	-54
34	3	3	2	-7
35	3	3	3	-32
36	3	3	4	60

Unbalanced case.

		Crusher		
		1	2	3
Mixer	1	28, 52,	-66	-84, 18, 32,
	2	-58, 28, 58, -10	34, -12,	-82, -20
	3		72, -24, 62,	-54, -7, -32, 60

Table 2. Values (coded) of the strength of 21 concrete samples using three different mixers and three different crushers in an unbalanced, two-way layout. The data is a proper subset of the data in Table 1.

Program 3

```
/*The input data is organized as a subset of the input data from program 1.*/

data cementredu;
infile cards missover;
input mixer crusher strength @;
/*Introducing the counting variable "index" is not necessary, but the variable may be useful.*/
index=1;
do until (strength =.);
output;
index=index+1;
input strength @;
end;
cards;
1  1   28 52
1  2  -66
1  3  -84 18 32
2  1  -58 28 58 -10
2  2   34 -12
2  3  -82 -20
3  1
3  2   72 -24 62
3  3  -54 -7 -32 60
;
run;

Title 'Reduced Cement Data';
proc print data=cementredu;
var mixer crusher index strength;
run;
```

Output, unbalanced case.

Reduced Cement Data

Obs	mixer	crusher	index	strength
1	1	1	1	28
2	1	1	2	52
3	1	2	1	-66
4	1	3	1	-84
5	1	3	2	18
6	1	3	3	32
7	2	1	1	-58
8	2	1	2	28
9	2	1	3	58
10	2	1	4	-10
11	2	2	1	34
12	2	2	2	-12
13	2	3	1	-82
14	2	3	2	-20
15	3	1	1	.
16	3	2	1	72
17	3	2	2	-24
18	3	2	3	62
19	3	3	1	-54
20	3	3	2	-7
21	3	3	3	-32
22	3	3	4	60

MANOVA Case

An experiment was conducted to assess the effect of four nitrogen treatments in three blocks on several variables measured for Jonathan apples. Here we consider three of those variables;

- y_1: total nitrogen,
- y_2: potassium,
- y_3: calcium.

We consider the **multivariate** response:$[y_1 \quad y_2 \quad y_3]^T$. The data is presented in Table 3.

Treatment	Block	y_1	y_2	y_3
1	1	3580	8220	244
1	2	2870	7550	272
1	3	3110	8180	297
2	1	3040	10760	138
2	2	5810	11340	165
2	3	3010	9830	159
3	1	4250	11830	169
3	2	6950	12910	148
3	3	4680	11010	224
4	1	4700	8830	148
4	2	7230	10840	120
4	3	5760	9200	147

Program 4

```
/*The input data is organized as one three-dimensional observation
for each combination of treatment and block.*/

data Jonathan;
input Treatment Block y1 y2 y3;
datalines;
1 1  3580   8220  244
1 2  2870   7550  272
1 3  3110   8180  297
2 1  3040  10760  138
2 2  5810  11340  165
2 3  3010   9830  159
3 1  4250  11830  169
3 2  6950  12910  148
3 3  4680  11010  224
4 1  4700   8830  148
4 2  7230  10840  120
4 3  5760   9200  147
;
run;
```

```
Title 'The Jonathan Data set';
proc print data=Jonathan;
run;
```

Output, MANOVA case

The Jonathan Data set

Obs	Treatment	Block	y1	y2	y3
1	1	1	3580	8220	244
2	1	2	2870	7550	272
3	1	3	3110	8180	297
4	2	1	3040	10760	138
5	2	2	5810	11340	165
6	2	3	3010	9830	159
7	3	1	4250	11830	169
8	3	2	6950	12910	148
9	3	3	4680	11010	224
10	4	1	4700	8830	148
11	4	2	7230	10840	120
12	4	3	5760	9200	147

■

Reading outstat dataset

Parameters useful for making different plots may be found embedded in outstat datasets. Therefore it is of interest to read parts of columns in SAS output data set and rearrange those as observations.

```
/*The SAS procedure cancorr keeps observations and estimated parameters in the outstat data
set, here called outcement.*/
proc cancorr data=cement outstat=outcement;
var STRGTH3 STRGTH7 STRGTH28 ;
with XSO3 LSR BLAINE;
run;

Title 'The outcement data set';
proc print data=outcement;
run;
```

The outcement data set

Obs	_TYPE_	_NAME_	STRGTH3	STRGTH7	STRGTH28	XSO3	LSR	BLAINE
1	MEAN		263.894	382.288	515.278	1.941	40.293	3095.72
2	STD		36.883	36.705	32.671	0.254	8.554	234.22
3	N		198.000	198.000	198.000	198.000	198.000	198.00
4	CORR	STRGTH3	1.000	0.896	0.608	0.474	-0.178	0.80
5	CORR	STRGTH7	0.896	1.000	0.746	0.373	-0.072	0.74
6	CORR	STRGTH28	0.608	0.746	1.000	0.279	-0.085	0.51
7	CORR	XSO3	0.474	0.373	0.279	1.000	-0.521	0.49
8	CORR	LSR	-0.178	-0.072	-0.085	-0.521	1.000	-0.20
9	CORR	BLAINE	0.800	0.736	0.514	0.491	-0.196	1.00
10	CANCORR		0.807	0.242	0.009	.	.	.
11	SCORE	V1	0.926	0.062	0.030	0.000	0.000	0.00
12	SCORE	V2	-2.048	2.692	-0.643	0.000	0.000	0.00
13	SCORE	V3	-0.461	-0.468	1.390	0.000	0.000	0.00
14	SCORE	W1	0.000	0.000	0.000	0.152	0.049	0.93
15	SCORE	W2	0.000	0.000	0.000	-0.366	0.817	0.39
16	SCORE	W3	0.000	0.000	0.000	-1.262	-0.843	0.56
17	RAWSCORE	V1	0.025	0.002	0.001	0.000	0.000	0.00
18	RAWSCORE	V2	-0.056	0.073	-0.020	0.000	0.000	0.00
19	RAWSCORE	V3	-0.012	-0.013	0.043	0.000	0.000	0.00
20	RAWSCORE	W1	0.000	0.000	0.000	0.599	0.006	0.00
21	RAWSCORE	W2	0.000	0.000	0.000	-1.437	0.096	0.00
22	RAWSCORE	W3	0.000	0.000	0.000	-4.958	-0.099	0.00
23	STRUCTUR	V1	0.999	0.913	0.639	0.470	-0.171	0.80
24	STRUCTUR	V2	-0.029	0.378	0.118	-0.145	0.226	0.01
25	STRUCTUR	V3	-0.035	0.155	0.760	-0.005	-0.003	0.00
26	STRUCTUR	W1	0.806	0.737	0.516	0.582	-0.212	0.99
27	STRUCTUR	W2	-0.007	0.091	0.028	-0.600	0.931	0.05
28	STRUCTUR	W3	-0.000	0.001	0.007	-0.548	-0.296	0.11

Program 1

```
/*The CanWT data set contains the last three entries in columns 4, 5,
and 6 organized as three 'observations' with four variables.*/
data CanW;
set outcement;
  *if (_type_ = 'STRUCTUR') AND (((_name_='W1') OR (_name_='W2')) OR _name_='W3');
  if 26<=_n_<=28;
  drop _type_ _name_ XSO3 LSR BLAINE;
run;
proc transpose data=CanW out=CanWT;
run;
data CanWT;
set CanWT;
  CVW1=col1;
  CVW2=col2;
  CVW3=col3;
  drop col1 col2 col3;
run;

Title 'The CanWT data set';
proc print data=CanWT;
run;
```

Output

The CanWT data set

Obs	_NAME_	CVW1	CVW2	CVW3
1	STRGTH3	0.80631	-0.007022	-.000317468
2	STRGTH7	0.73678	0.091423	0.001423116
3	STRGTH28	0.51550	0.028487	0.006963891

```
/*The output from Program 1 is identical to the result of the following program. However, the
latter requires that the values are entered once more.*/
data CanWT;
input _NAME_ $ CVW1-CVW3;
cards;
STRGTH3   0.80631  -0.007022  -0.000317468
STRGTH7   0.73678   0.091423   0.001423116
STRGTH28  0.51550   0.028487   0.006963891
;
```

Program 2

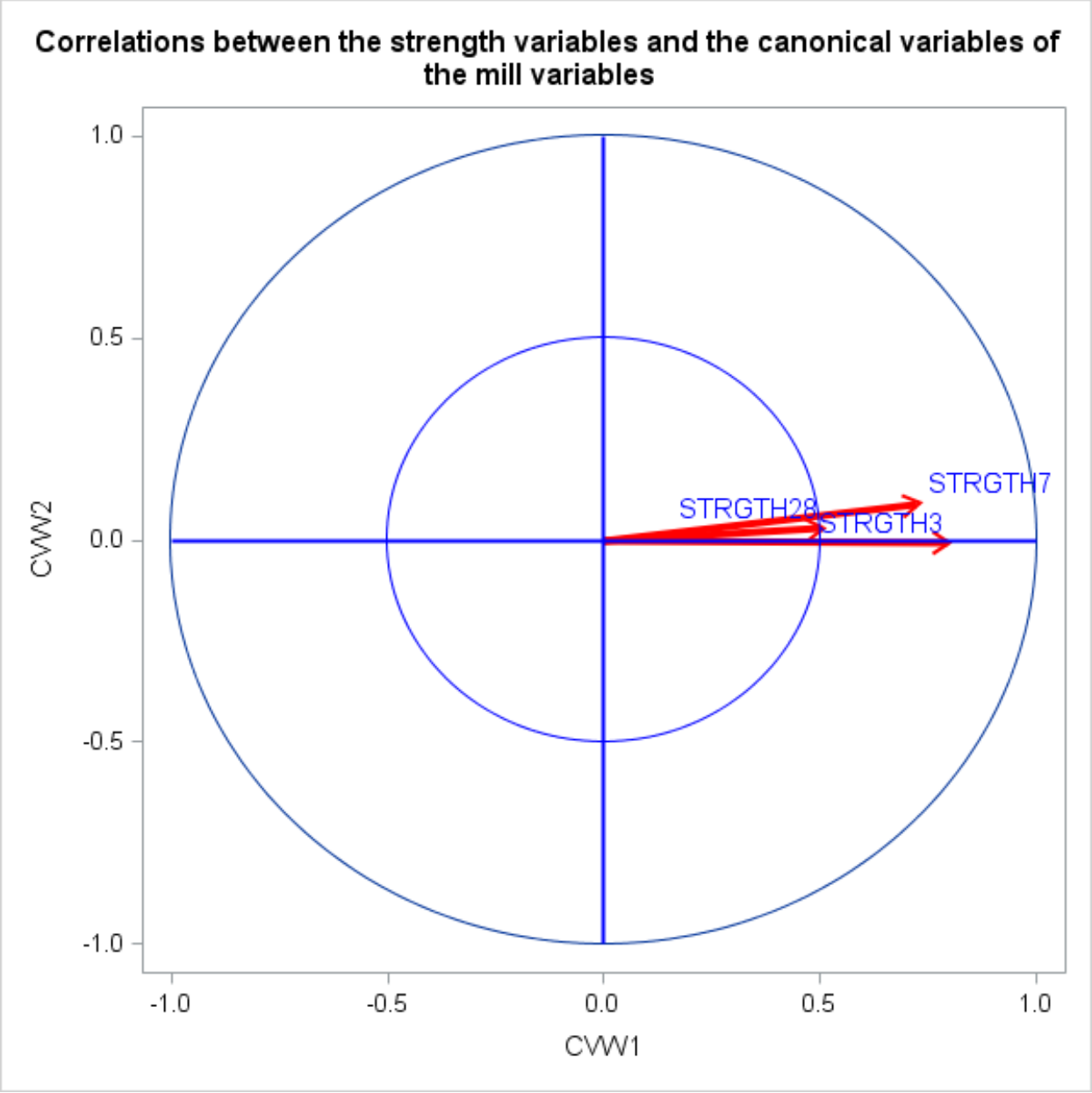
Generates a component pattern plot based on the output from Program 1.

```
ODS graphics on/ reset width=600px height=600px;
proc template;
define statgraph patternplot;
dynamic xvar yvar title;
begingraph;
entrytitle title;
  layout overlay;
    vectorplot Y=yvar X=xvar xorigin=0 yorigin=0 /
      lineattrs=(color=red thickness=4) datalabel=_name_
      datalabelattrs=(color=blue size=10)arrowheadshape=open ;
    ellipseparm
      semimajor=1 semiminor=1 slope=0 xorigin=0 yorigin=0;
    ellipseparm
      semimajor=0.5 semiminor=0.5 slope=0 xorigin=0 yorigin=0/
      outlineattrs=(color=blue thickness=1);
    lineparm
      x=0 y=0 slope=0/lineattrs=(color=blue thickness=2);
    lineparm
      x=0 y=0 slope=./lineattrs=(color=blue thickness=2);
  endlayout;
endgraph;
end;
run;

Title 'Component Pattern';
proc sgrender data=CanWT template=patternplot;
dynamic yvar="CVW2" xvar="CVW1" title="Correlations between the strength variables and
the canonical variables of the mill variables";
run;
ODS graphics off;
```

Output

Component Pattern



■

Input from csv- files and txt-files

CamillaData.csv is an Excel file saved as a **csv file**, i.e. a file with **c**omma **s**eparated **v**alues. In total there are 105 observations in the file. The first 11 lines of the file are:

```
Piece,Pos,Sample,Wavelength,Slope,Inter,DeltaX
1,P2,1,500,-0.00082176,2.8851,-0.022517
1,P2,2,500,-0.00082486,2.8877,-0.028655
1,P3,3,500,-0.0008291,2.8838,-0.024716
1,P3,4,500,-0.00082584,2.8816,-0.029507
1,P1,5,500,-0.00083873,2.882,-0.12152
1,P1,6,500,-0.00083255,2.884,-0.1209
1,P1,7,500,-0.00083938,2.8904,-0.1251
1,P1,8,500,-0.00083944,2.893,-0.12269
1,P1,9,500,-0.00081837,2.8914,-0.1204
1,P1,10,500,-0.00082271,2.8934,-0.12076
```

The names of the variables are given in the first line.

Program 1

```
proc import
  datafile="C:\Users\knco\Documents\Multstat2017/CamillaData.csv"
  out=FeatureData dbms=dlm replace;
delimiter=",";
getnames=yes;
guessingrows=200;
run;

Title 'The 10 first observations from the FeatureData data set';
proc print data = FeatureData (Obs=10);
run;
```

Output

The 10 first observations from the FeatureData data set

Obs	Piece	Pos	Sample	Wavelength	Slope	Inter	DeltaX
1	1	P2	1	500	-0.00082176	2.8851	-0.022517
2	1	P2	2	500	-0.00082486	2.8877	-0.028655
3	1	P3	3	500	-0.0008291	2.8838	-0.024716
4	1	P3	4	500	-0.00082584	2.8816	-0.029507
5	1	P1	5	500	-0.00083873	2.882	-0.12152
6	1	P1	6	500	-0.00083255	2.884	-0.1209
7	1	P1	7	500	-0.00083938	2.8904	-0.1251
8	1	P1	8	500	-0.00083944	2.893	-0.12269
9	1	P1	9	500	-0.00081837	2.8914	-0.1204
10	1	P1	10	500	-0.00082271	2.8934	-0.12076

Program 2

The file may also be entered directly using the infile statement like done below. Here we must know the number of variables.

```
data FeatureData2;
infile "C:\Users\knco\Documents\Multstat2017/CamillaData.csv" delimiter=",";
input var1-var7;
run;

Title 'The 3 first observations from the FeatureData2 data set';
proc print data = FeatureData2 (Obs=3);
run;
```

Output

The 3 first observations from the FeatureData2 data set

Obs	var1	var2	var3	var4	var5	var6	var7
1	.	.	.	.	.	.	.
2	1	.	1	500	-.00082176	2.8851	-0.022517
3	1	.	2	500	-.00082486	2.8877	-0.028655

From the output follows that the first line (probably) contains the variable names and that that variable number two is a character value. Therefore we modify the program and tells that the first data record is in line two using the firstobs option and that variable two is a character variable using the $ option.

Program 3

```
data FeatureData3;
infile "C:\Users\knco\Documents\Multstat2017/CamillaData.csv" delimiter="," firstobs=2;
input var1  var2 $  var3-var7;
run;

Title 'The 3 first observations from the FeatureData3 data set';
proc print data = FeatureData3 (Obs=3);
run;
```

Output

The 3 first observations from the FeatureData3 data set

Obs	var1	var2	var3	var4	var5	var6	var7
1	1	P2	1	500	-.00082176	2.8851	-0.02252
2	1	P2	2	500	-.00082486	2.8877	-0.02866
3	1	P3	3	500	-.00082910	2.8838	-0.02472

The advantage of using **proc import** instead of the infile statement is that proc import (according to 'Reading Delimited Text Files into SAS®9'):

- scans the first 20 records
- collects the variable names from the first row
- scans the remaining 19 rows and determines the variable types
- assigns an informat and a format to each variable
- creates an INPUT statement
- submits all of the code to the DATA step compiler, which, in turn, executes the code

Program 4

India1.txt is a text-file with observations from the Landsat satellite. Each observation consists of the six reflection values for a pixel. No information on the names are given, so **proc import** allocates the names VAR1, ... , VAR6. We use the same name in the infile version of the reading program.

```
proc import
  datafile="C:\Users\knco\Documents\MultStat2017\India1.txt"
  out=India1Landsat dbms=dlm replace;
getnames=no;
guessingrows=1000;
run;

Title 'The 5 first observations from the India1Landsat data set';
proc print data = India1Landsat (Obs=5);
run;
```

Program 5

```
data India1Landsat;
infile "C:\Users\knco\Documents\MultStat2017\India1.txt";
input VAR1-VAR6;
run;

Title 'The 5 first observations from the India1Landsat data set';
proc print data=India1Landsat2 (Obs=5);
```

Output from programs 4 and 5

The 5 first observations from the India1Landsat data set

Obs	VAR1	VAR2	VAR3	VAR4	VAR5	VAR6
1	133	67	100	77	166	115
2	135	72	108	83	173	123
3	141	76	116	88	184	130
4	142	79	122	93	189	134
5	141	78	121	94	186	133

■