# A small SAS-manual

## 1 The SAS screen

The statistical program SAS has three different windows. The OUTPUT window, the LOG window and finally the PROGRAM EDITOR.
In the PROGRAM EDITOR the SAS programs will be typed. If you want really to use SAS when you want to estimate some models, you have to put some time in learning the SAS-programming language. But in order to do some simple OLS regressions with some options, this manual will suffice.
In the LOG window the SAS-programs will be shown after they were submitted[1]. Optional faults in the SAS program will also be shown here.
In the OUTPUT window, all the output will be shown, which is generated by a SAS program.

## 2 Data sets

Working with data sets in SAS isn't that difficult, because it's possible to use the so called LIBNAME-statement. This is a reference to a physical place of a permanent data set[2]. It's also possible in SAS to work with temporary data sets, which will be deleted after the SAS session.
With the LIBNAME-statement a reference is made to a place on the hard disk or diskette were the permanent data sets are located. We can illustrate this with an example.
Suppose that the permanent data set TEST.SD2 is located in the directory C:\FILES\SAS. Instead of typing the whole path name over and over again, we can use a LIBNAME-statement to make a reference to this directory. We call this reference SAAB, but it's possible to come up with another name.
Using the following SAS-program, we refer with the name SAAB to the directory C:\FILES\SAS.

***SAS Program 1 (LIBNAME statement)***

```
Libname SAAB 'c:\files\sas';
run;
```

After submitting the program, the following will be shown in the LOG window of SAS.

[1] To submit, choose LOCALS and SUBMIT or press the icon with a S from 'SUBMIT'.
[2] Permanent SAS data sets are files with the extension .sd2.

***LOG window (After LIBNAME statement)***

```
1
2    Libname SAAB 'c:\files\sas';
NOTE: Libref SAAB was successfully assigned as follows:
      Engine:        V611
      Physical Name: C:\FILES\SAS
3    run;
```

As said before permanent data sets are files with the extension .sd2. With the help of the LIBNAME-statement it's easy to access these files within SAS. With use of the procedure PRINT, it's possible to see the contents of the data set PROEF.SD2 in the OUTPUT window. To so this we use the following SAS program:

***SAS program 2 (Showing the contents of the data set PROEF.SD2)***

```
Proc print data=saab.proef;
run;
```

After submitting this program, we have the following output in the OUTPUT window

***OUTPUT window (Partly)  (After submitting SAS program 2)***

| OBS | UREN | LOGLOON | VAKBOND | MAN | GETR | OPL | ERVARING | BLANK |
|---|---|---|---|---|---|---|---|---|
| 1 | 40.0000 | 1.99461 | 0 | 0 | 1 | 13 | 9 | 0 |
| 2 | 48.0000 | 1.03213 | 0 | 0 | 0 | 12 | 10 | 0 |
| 3 | 31.9615 | 0.42720 | 0 | 0 | 1 | 12 | 9 | 0 |
| 4 | 40.0000 | 1.30146 | 0 | 0 | 1 | 11 | 9 | 0 |
| 5 | 49.8077 | 1.77532 | 0 | 0 | 1 | 12 | 11 | 0 |
| 6 | 38.0000 | 2.02910 | 1 | 0 | 1 | 12 | 11 | 0 |
| 7 | 40.9808 | 1.43139 | 0 | 0 | 1 | 16 | 8 | 1 |
| 8 | 34.5192 | -0.85376 | 0 | 0 | 0 | 12 | 11 | 1 |
| 9 | 23.0769 | 2.71651 | 0 | 0 | 1 | 14 | 10 | 1 |
| 10 | 40.0000 | 1.60157 | 0 | 0 | 1 | 12 | 12 | 1 |

If we want change some of the variables in the data set which we are working with, it's useful to work with *temporary* data sets. With SAS program 3, the permanent data set PROEF.SD2 will be copied with use of the option SET to the *temporary* data set TEST.

***SAS program 3 (Copying to a temporary data set)***

```
Data test;
set saab.proef;
run;
```

As you can see in the program, the libname SAAB is not used in the DATA-statement. This way SAS knows that the data set TEST is temporary. If we would have used the libname SAAB in the DATA-statement (Data saab.test), the data set PROEF.SD2, would have been copied to the *permanent* data set TEST.sd2.

# 3 Ordinary Least Squares Estimation

OLS estimation in SAS will be performed by the procedure REG. As an example, we can estimate the following equation with OLS.

$$\text{UREN}_i = b_0 + b_1\text{LOGLOON}_i + b_2\text{VAKBOND}_i + b_3\text{MAN}_i + b_4\text{GETR}_i + e_i$$

To so this, we use SAS program 4.

***SAS program 4 (A OLS regression)***

```
Proc reg data=saab.proef;
model uren = logloon vakbond man getr;
run;
```

In the PROC REG statement we mention the data set which has to be used. In our case it's the data set PROEF.SD2, which is in the directory C:\FILES\SAS. In the MODEL statement, we put the equation we want to estimate. SAS automatically adds a constant term (to estimate without a constant, you have to add the option /NOINT after te model statement).

SAS program 4 generates the following output:

***OUTPUT window (Generated by SAS program 4)***

```
Model: MODEL1
Dependent Variable: UREN

                          Analysis of Variance

                             Sum of         Mean
  Source          DF      Squares       Square      F Value       Prob>F

  Model            4   1567.50211    391.87553        4.324       0.0028
  Error          108   9787.09178     90.62122
  C Total        112  11354.59389

      Root MSE       9.51952     R-square       0.1381
      Dep Mean      41.74149     Adj R-sq       0.1061
      C.V.          22.80589

                          Parameter Estimates

                   Parameter      Standard    T for H0:
Variable  DF      Estimate         Error   Parameter=0    Prob > |T|

INTERCEP   1     41.378101    3.16675351        13.066        0.0001
LOGLOON    1     -2.670460    1.72915476        -1.544        0.1254
VAKBOND    1      3.422418    2.14442556         1.596        0.1134
MAN        1      6.795788    1.83019412         3.713        0.0003
GETR       1      0.672989    1.89331473         0.355        0.7229
```

Within the PROC REG program there are several options to let SAS generate more output. Suppose we want to know the covariance matrix of the error term. By putting the option /COVB behind the MODEL statement in SAS program 4, the covariance matrix will be calculated and shown as an extra output in the OUTPUT window.
In SAS it's also possible to calculate the so called *White Covariance Matrix*. Instead of the option /COVB, we then have to use the option /ACOV in SAS program 4. SAS program 5 shows how to calculate this covariance matrix.

***SAS program 5 (Calculation of the White Covariance Matrix)***

```
Proc reg data=saab.proef;
model uren = logloon vakbond man getr /ACOV;
run;
```

This program generates the following (extra) output:

***OUTOUT window (Generated by SAS program 5)***

Consistent Covariance of Estimates

| ACOV | INTERCEP | LOGLOON | VAKBOND | MAN | GETR |
|---|---|---|---|---|---|
| INTERCEP | 7.5310212677 | -3.716695966 | -0.261926415 | 0.4667288863 | -0.575280439 |
| LOGLOON | -3.716695966 | 2.5443679345 | -0.590685331 | -1.184248035 | -0.579505322 |
| VAKBOND | -0.261926415 | -0.590685331 | 4.8344760057 | 0.5386820717 | -0.008456779 |
| MAN | 0.4667288863 | -1.184248035 | 0.5386820717 | 3.2494165512 | 0.3502030633 |
| GETR | -0.575280439 | -0.579505322 | -0.008456779 | 0.3502030633 | 2.9338661308 |

By taking the square root of the diagonal elements of these matrix, we get the so called *White Standard Errors*.
To calculate the Durbin-Watson statistic in order to test whether or not the errors have first-order autocorrelation, the option /DW has to be added to the MODEL-statement.

# 4 Estimation with LOGIT and PROBIT

First a SAS Program:

***SAS Program***

```
proc probit data=bieb.TEST;
class UNION;
model UNION = EXPER EX2 SCHOOL MAR /D=NORMAL;
output out=bieb.UIT1 p=PRED1;
run;
```

What is it what this program does? It uses the dataset TEST.SD2 in the directory whereto the libname link BIEB has been made.
The line with the CLASS-statement determines that the variable UNION is the latent variable of the binary respons model under consideration.

In the line with the MODEL-statement, the model we want to estimate is specified. In this line it is also denoted which distribution SAS has to use to estimate the model. In the example the option /D=NORMAL was used, which implies that SAS will estimate a PROBIT model. IF we would change the option into /D=LOGISTIC, we would estimate a LOGIT model.
The line with the OUTPUT-statement, writes the dataset UIT1 (to the directory whereto the libname BIEB is pointing). In the dataset there is also the variable PRED1 included. This variable denotes the prediction of every individual, on the basis of the model which was estimated before.
**IMPORTANT:** If we use PROC PROBIT, SAS estimates - ß¸ instead of ß. Be carefull about this in your interpretation of the estimation results.

Using a cross-tabular matrix, it is possible to determine the predictionquality of the model. To make such a cross-tabular matrix in SAS, we use the following program.

***SAS Program (Cross-tabular matric to determine the prediction quality of the model)***

```
data bieb.HULP1;
set bieb.UIT1;
if PRED1 >= 0.5 then UNION_P1 = 1;
if PRED1 < 0.5 then UNION_p1 = 0;
run;

proc freq data=bieb.HULP1;
tables UNION*UNION_P1;
run;
```

This program, creates the dataset HULP1 and to do this it uses the dataset UIT1. The second part of the program has as output a cross-tabular matrix in which the prediciton (according to the model) of the variable UNION has been shown against the observered variable UNION. Using this cross-tabular matrix it is possible to calculate a measure of fit of the model.