How to adapt the code samples in "Credit Risk Analytics"

by Bart Baesens, Daniel Rosch and Harald Scheule, Wiley, 2016

for SAS OnDemand for Academics

Aniruddho "Oni" Sanyal
Principal
Phoenix Computing Solutions
8 Joanne Court
Bridgewater
NJ 08807
USA

Mobile/WhatsApp (US): +1.732.407.8966

Email: asanyal@phoenixcomputing.net

Website: http://www.phoenixcomputing.net/home.html

Introduction

The best choice for this purpose, for students taking this course, in my experience, is the cloud based SOA or SAS OnDemand for Academics (https://www.sas.com/en_us/software/on-demand-for-academics.html). The registration is fairly straightforward and the interface is explained by a series of videos on YouTube from SAS institute, for example this one is good for an overview. This write-up will not get into the details of the GUI for SOA as they are fairly self-explanatory for anyone that has used some other version of SAS, e.g., SAS for Windows, SAS Enterprise Guide etc.

I will not recommend the somewhat better-known but less user-friendly SAS University Edition. This version depends on the Oracle VirtualBox for its functioning and many people (myself included) tend to have latency and other issues with it.

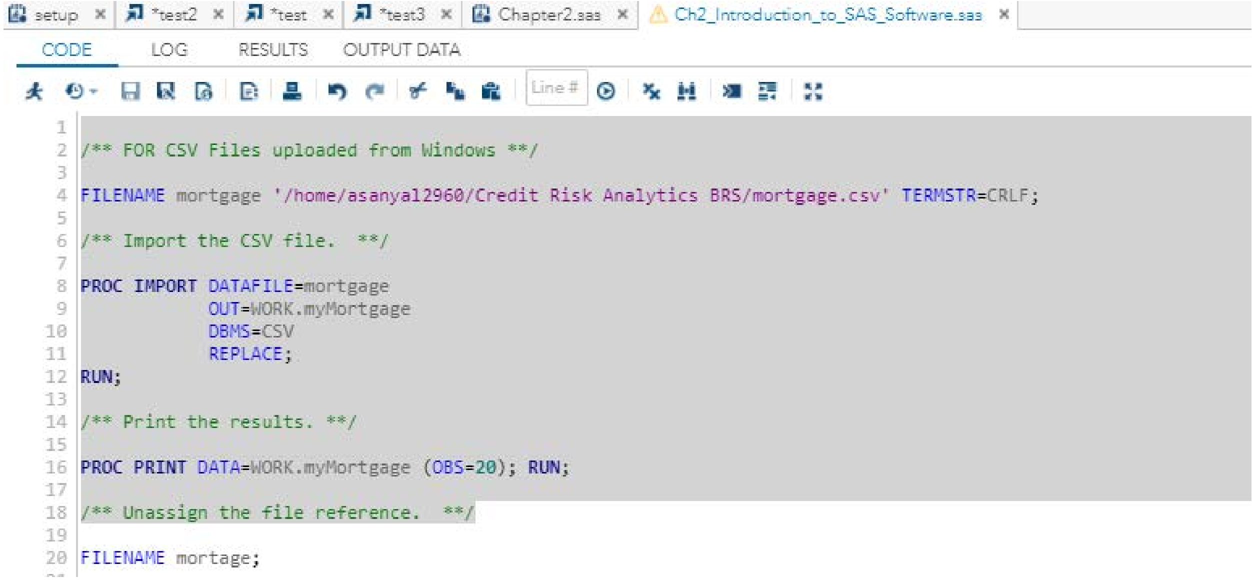
The purpose of this blurb is to show how you can get the code examples in the aforementioned book to work in the SOA environment. It will focus on the modifications that have worked for me. I have found these through trial-and-error and though there might be better ways, I am focusing on *what will invariably work for the typical SOA user*. These are important to note down as SAS technical support folks are often unable to provide you with such specific information. So, I hope to save you the research and the trial-and-error time that I had to invest.

Code modifications that work in SAS Studio

You are likely to encounter two different kinds of situations to use the data samples provided on the book's website http://www.creditriskanalytics.net/datasets.html. You are likely to encounter two different types of situations.

1. Use a LIBNAME command to access a pre-determined directory in which a SAS dataset resides (e.g., at the bottom of page 19 of the book). That sort of a LIBNAME command will not work in SOA but there is a workaround that I have found that you can use. This is what you would need to do when using the SAS dataset named *hmeq.sas7bdat* that is introduced on page 62.
2. Import a csv or an xlsx file containing the data, e.g., mortgage.csv data that is on page 20.

I will start with number 2 above first as it appears first in the book, where, on page 20 an example is laid out where a .csv file has to be imported. You have to upload the mortgage.csv file using the dedicated icon to your cloud folder. It will typically have the following structure '/home/yourSOAuserid/specific_folder.' As seen in the screenshot below, mine is '/home/asanyal2960/Credit Risk Analytics BRS/mortgage.csv' – and, please also note the syntax around it that makes it readable to SAS. Note the PROC IMPORT statement, especially the DBMS=CSV option. If it were an Excel file, you would have to type DBMS=XLS or DBMS=XLSX, as the case might be.

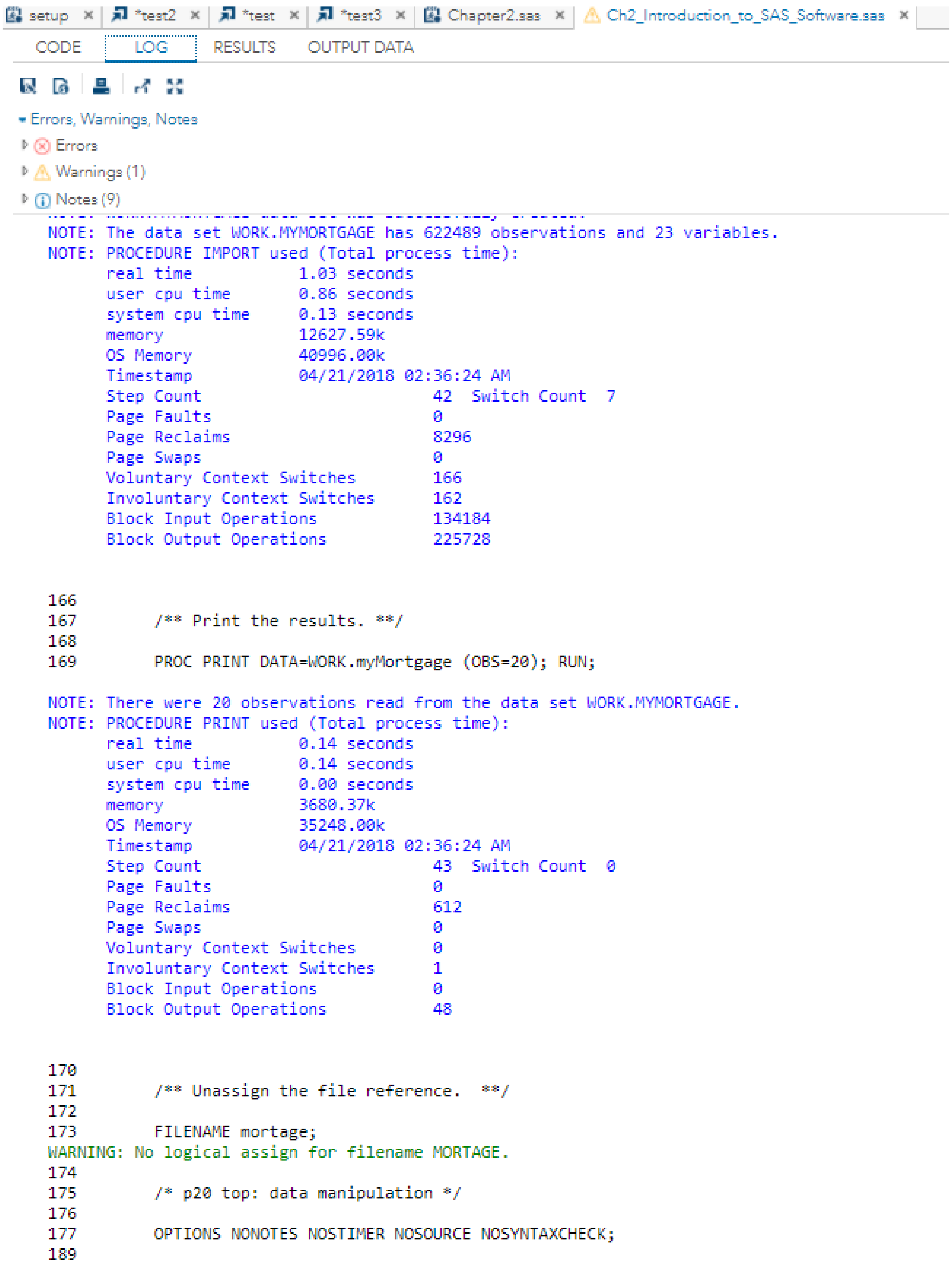


```

/** FOR CSV Files uploaded from Windows **/

FILENAME mortgage '/home/asanyal2960/Credit Risk Analytics BRS/mortgage.csv' TERMSTR=CRLF;

/** Import the CSV file.  **/

PROC IMPORT DATAFILE=mortgage
            OUT=WORK.myMortgage
            DBMS=CSV
            REPLACE;
RUN;

/** Print the results. **/

PROC PRINT DATA=WORK.myMortgage (OBS=20); RUN;

/** Unassign the file reference.  **/

FILENAME mortage;
```

For copy-paste-adapt purposes, you may use the following, obviously replacing my cloud path (highlighted) with your own. You can find this in the properties box when you right-click on the file

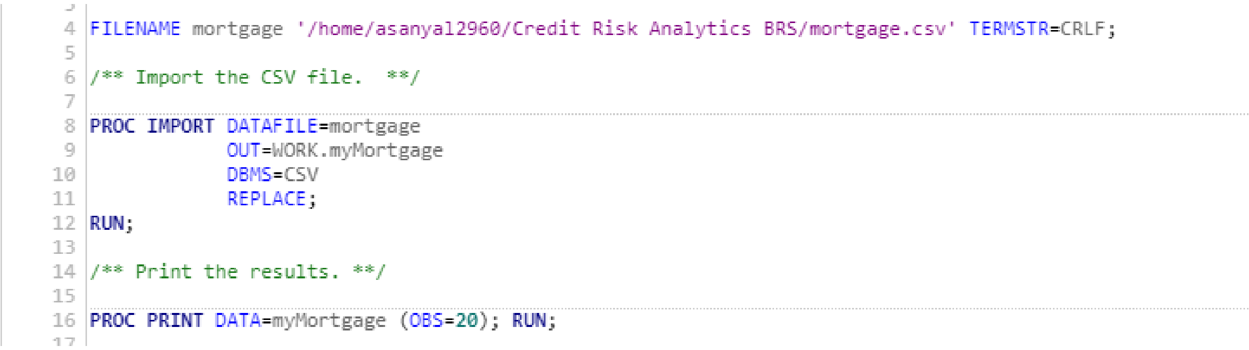
```
/** FOR CSV Files uploaded from Windows **/
FILENAME mortgage '/home/asanyal2960/Credit Risk Analytics BRS/mortgage.csv' TERMSTR=CRLF;
/** Import the CSV file.  **/

PROC IMPORT DATAFILE=mortgage
                OUT=WORK.myMortgage
                DBMS=CSV
                REPLACE;
RUN;

/** Print the results. **/
PROC PRINT DATA=WORK.myMortgage (OBS=20); RUN;
/** Unassign the file reference.  **/
FILENAME mortage;
```

This works and you should see something like the following in the log file:



```
NOTE: The data set WORK.MYMORTGAGE has 622489 observations and 23 variables.
NOTE: PROCEDURE IMPORT used (Total process time):
      real time           1.03 seconds
      user cpu time       0.86 seconds
      system cpu time     0.13 seconds
      memory              12627.59k
      OS Memory           40996.00k
      Timestamp           04/21/2018 02:36:24 AM
      Step Count                        42  Switch Count  7
      Page Faults                       0
      Page Reclaims                     8296
      Page Swaps                        0
      Voluntary Context Switches        166
      Involuntary Context Switches      162
      Block Input Operations            134184
      Block Output Operations           225728


166
167          /** Print the results. **/
168
169          PROC PRINT DATA=WORK.myMortgage (OBS=20); RUN;

NOTE: There were 20 observations read from the data set WORK.MYMORTGAGE.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.14 seconds
      user cpu time       0.14 seconds
      system cpu time     0.00 seconds
      memory              3680.37k
      OS Memory           35248.00k
      Timestamp           04/21/2018 02:36:24 AM
      Step Count                        43  Switch Count  0
      Page Faults                       0
      Page Reclaims                     612
      Page Swaps                        0
      Voluntary Context Switches        0
      Involuntary Context Switches      1
      Block Input Operations            0
      Block Output Operations           48


170
171          /** Unassign the file reference.  **/
172
173          FILENAME mortage;
WARNING: No logical assign for filename MORTAGE.
174
175          /* p20 top: data manipulation */
176
177          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
189
```

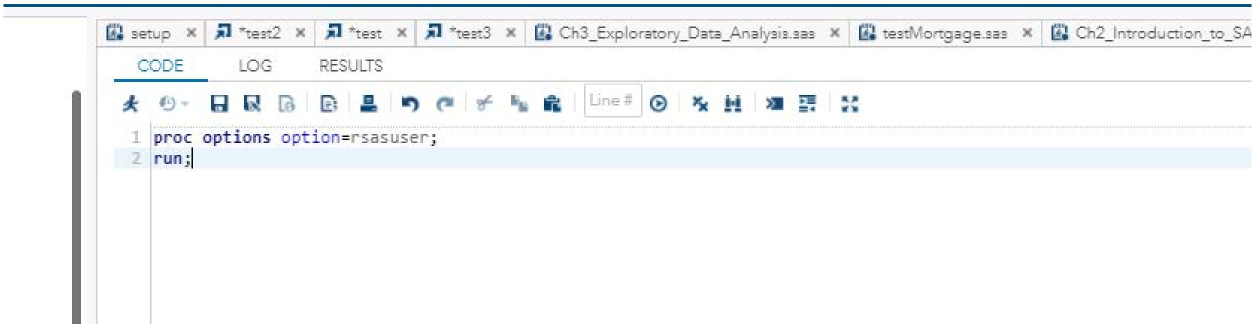
Of course, WORK.myMortgage is really literal. You can take out this ‘WORK.’ prefix and it will still work. So, the following variation will produce the exact same results



```
FILENAME mortgage '/home/asanyal2960/Credit Risk Analytics BRS/mortgage.csv' TERMSTR=CRLF;

/** Import the CSV file.  **/

PROC IMPORT DATAFILE=mortgage
            OUT=WORK.myMortgage
            DBMS=CSV
            REPLACE;
RUN;

/** Print the results. **/

PROC PRINT DATA=myMortgage (OBS=20); RUN;
```

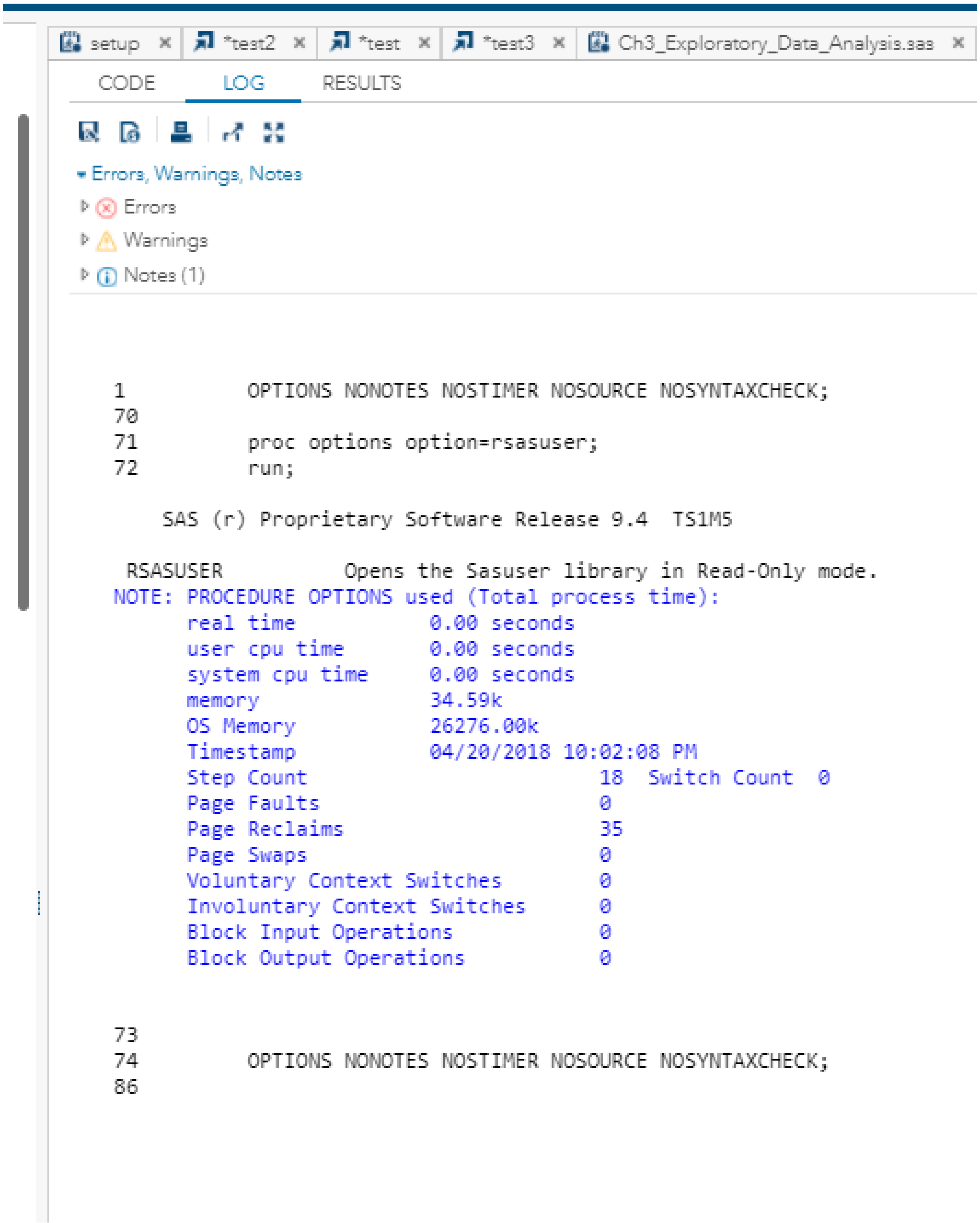
The second situation that you might face is when you have uploaded a proper SAS dataset to your cloud folder. Take the example of the code that starts at the bottom of page 62. The typical LIBNAME statement from SAS for Windows (as we see at the bottom of page 19), will not work for two reasons as SOA cannot see your hard directories such as ‘C:\Users’ and also because you will not usually have “write” access to the sasuser folder So the usual format of the LIBNAME statement will produce an error. Following are the details of checking your read/write privileges and taking appropriate action. At the command line, type

proc options option=rsasuser;
run ;



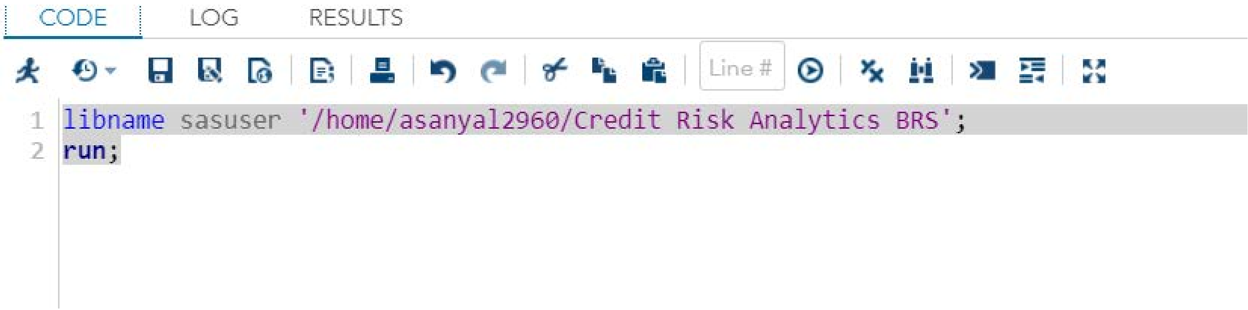
```
proc options option=rsasuser;
run;
```

If the sasuser folder is writeable, the log file will show NORSASUSER. *Typically, this will not be the case* and, therefore, you will not be able to use any of the LIBNAME commands in the book. The log file will show RSASUSER, which means that you will be a read-only user as follows:



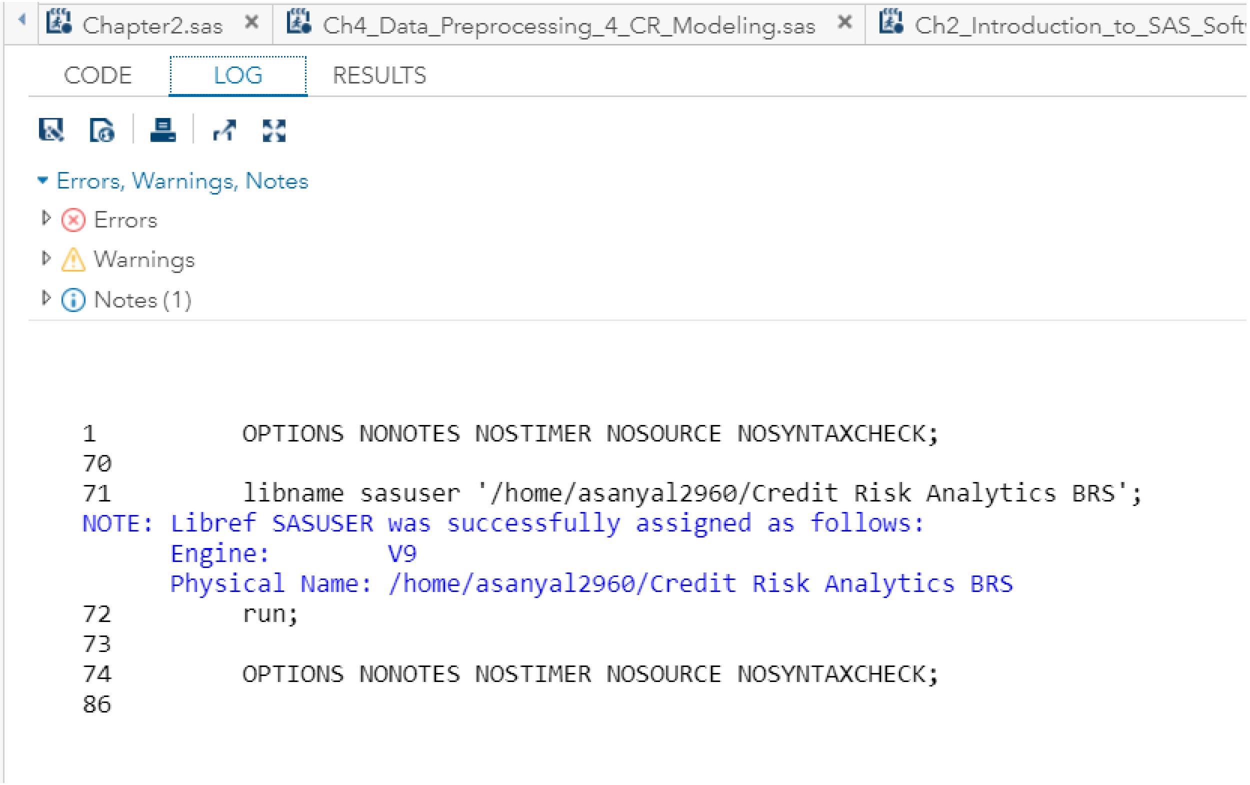
```
1          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
70
71         proc options option=rsasuser;
72         run;

    SAS (r) Proprietary Software Release 9.4  TS1M5

 RSASUSER          Opens the Sasuser library in Read-Only mode.
NOTE: PROCEDURE OPTIONS used (Total process time):
      real time           0.00 seconds
      user cpu time       0.00 seconds
      system cpu time     0.00 seconds
      memory              34.59k
      OS Memory           26276.00k
      Timestamp           04/20/2018 10:02:08 PM
      Step Count                        18  Switch Count  0
      Page Faults                       0
      Page Reclaims                     35
      Page Swaps                        0
      Voluntary Context Switches        0
      Involuntary Context Switches      0
      Block Input Operations            0
      Block Output Operations           0


73
74         OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
86
```

You could try to use the following syntax



```
libname sasuser '/home/asanyal2960/Credit Risk Analytics BRS';
run;
```

And this might work, as shown in the logfile below



```
1          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
70
71         libname sasuser '/home/asanyal2960/Credit Risk Analytics BRS';
NOTE: Libref SASUSER was successfully assigned as follows:
      Engine:        V9
      Physical Name: /home/asanyal2960/Credit Risk Analytics BRS
72         run;
73
74         OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
86
```

But this does not always work – unfortunately work and SAS technical support does not have a clear answer as to how to make it 100% reliable.

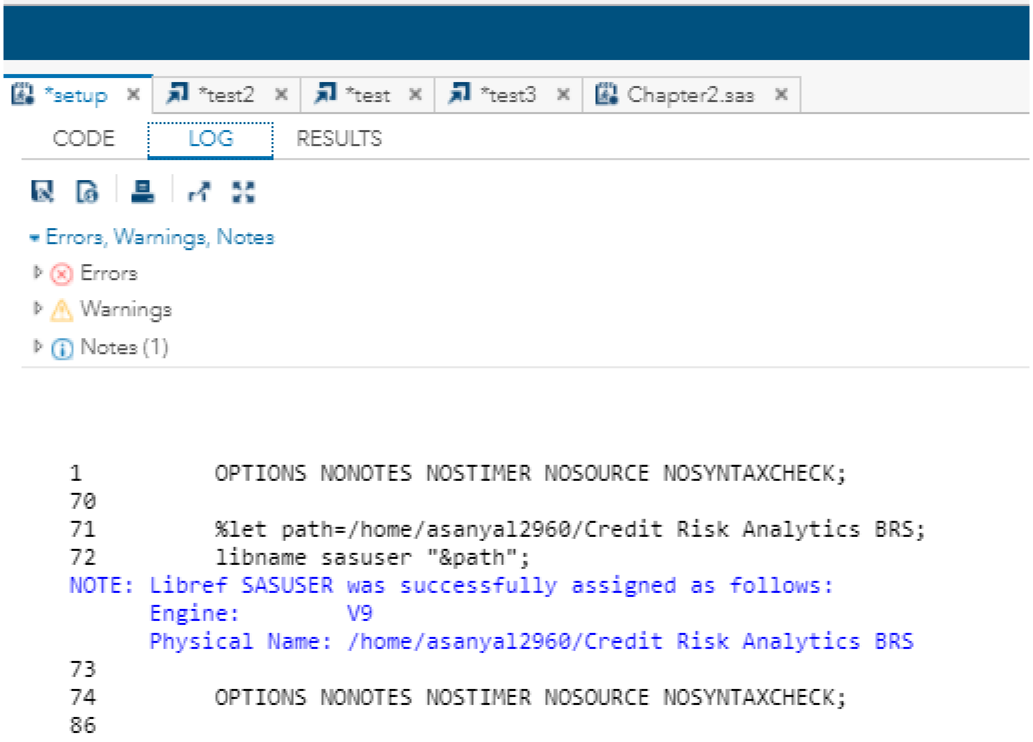
In this situation, I find the following workaround handy, as it works every time. In my case, for instance,

```
/*Credit Risk Analytics BRS*/
%let path=/home/asanyal2960/Credit Risk Analytics BRS;
libname sasuser "&path";
```

Save this file as setup.sas. As shown in the screenshot below,

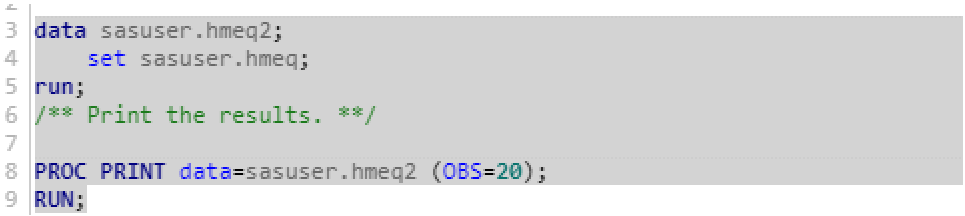
```
/*Credit Risk Analytics BRS*/
%let path=/home/asanyal2960/Credit Risk Analytics BRS;
libname sasuser "&path";
```

Run it every time, before you try the code samples of the book. This then works



```
1          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
70
71          %let path=/home/asanyal2960/Credit Risk Analytics BRS;
72          libname sasuser "&path";
NOTE: Libref SASUSER was successfully assigned as follows: 
      Engine:        V9 
      Physical Name: /home/asanyal2960/Credit Risk Analytics BRS
73
74          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
86
```

However, for whatever is already a SAS dataset, this does help quite a bit since you don’t have to keep typing in the full path. For instance, you might try the following:

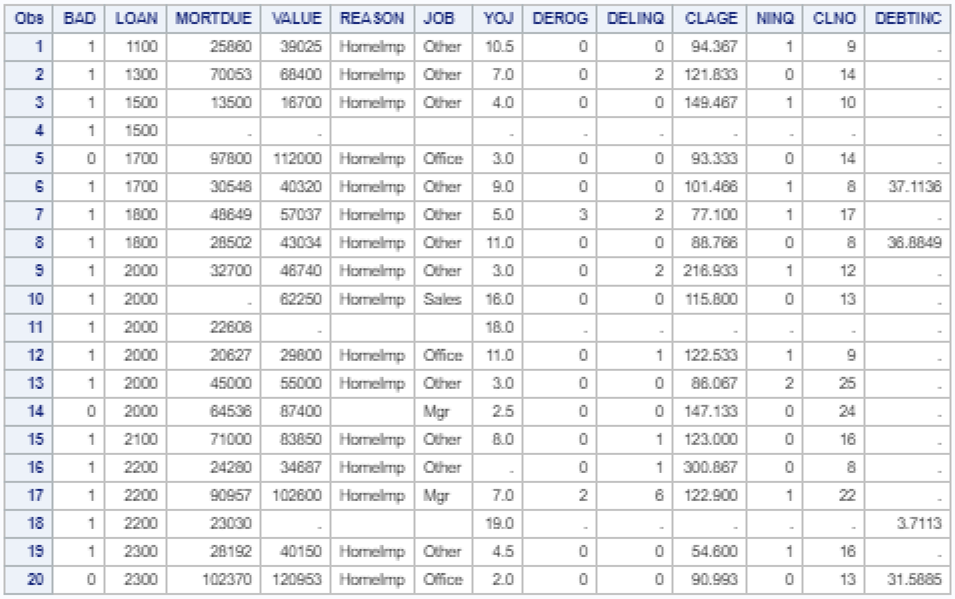


```
data sasuser.hmeq2;
    set sasuser.hmeq;
run;
/** Print the results. **/

PROC PRINT data=sasuser.hmeq2 (OBS=20);
RUN;
```

For ease of copy-paste-adapt, try the following and then modify

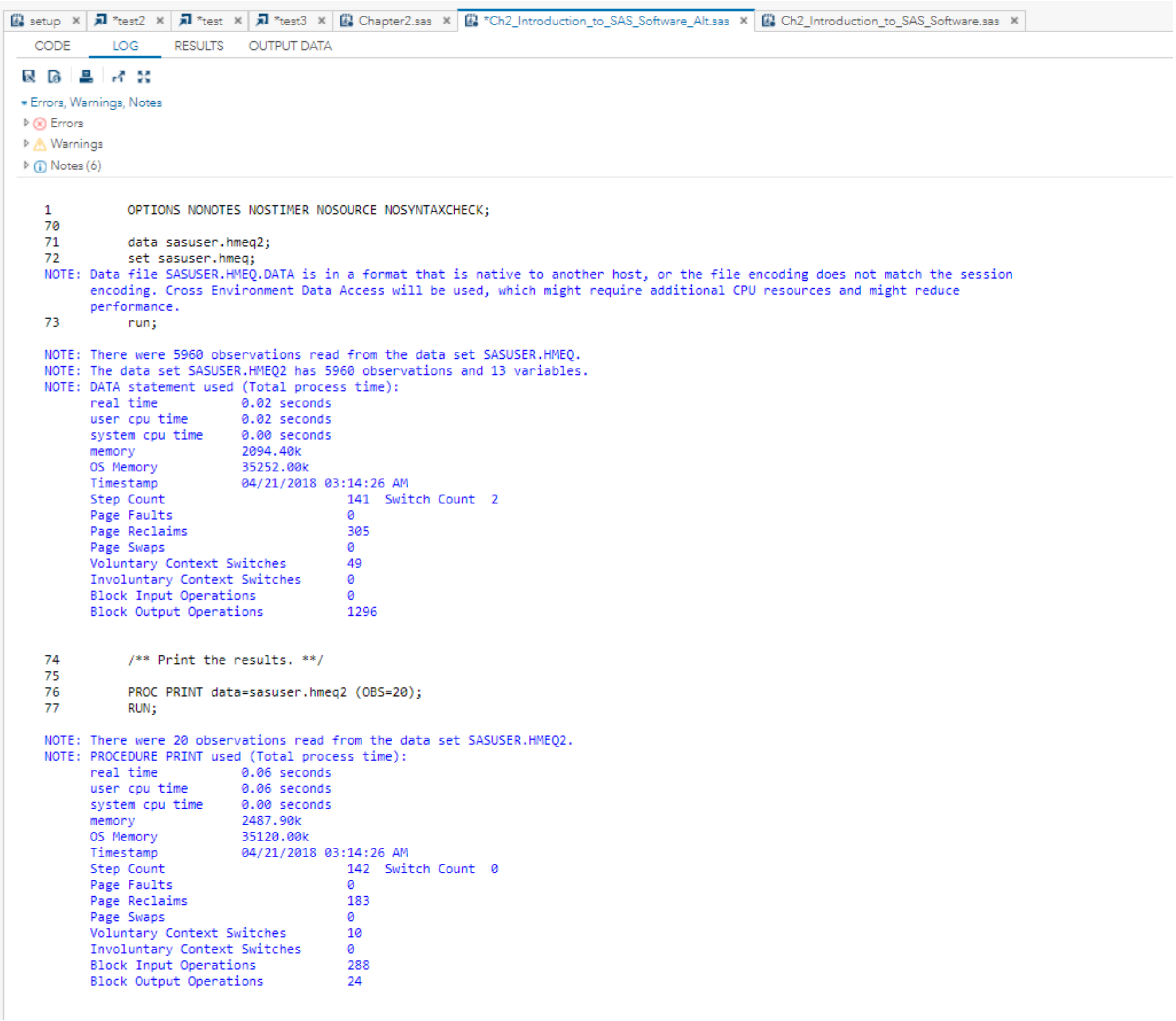
```
data sasuser.hmeq2;
            set sasuser.hmeq;
run;
/** Print the results. **/

PROC PRINT data=sasuser.hmeq2 (OBS=20);
RUN;
```

Correct results will be produced



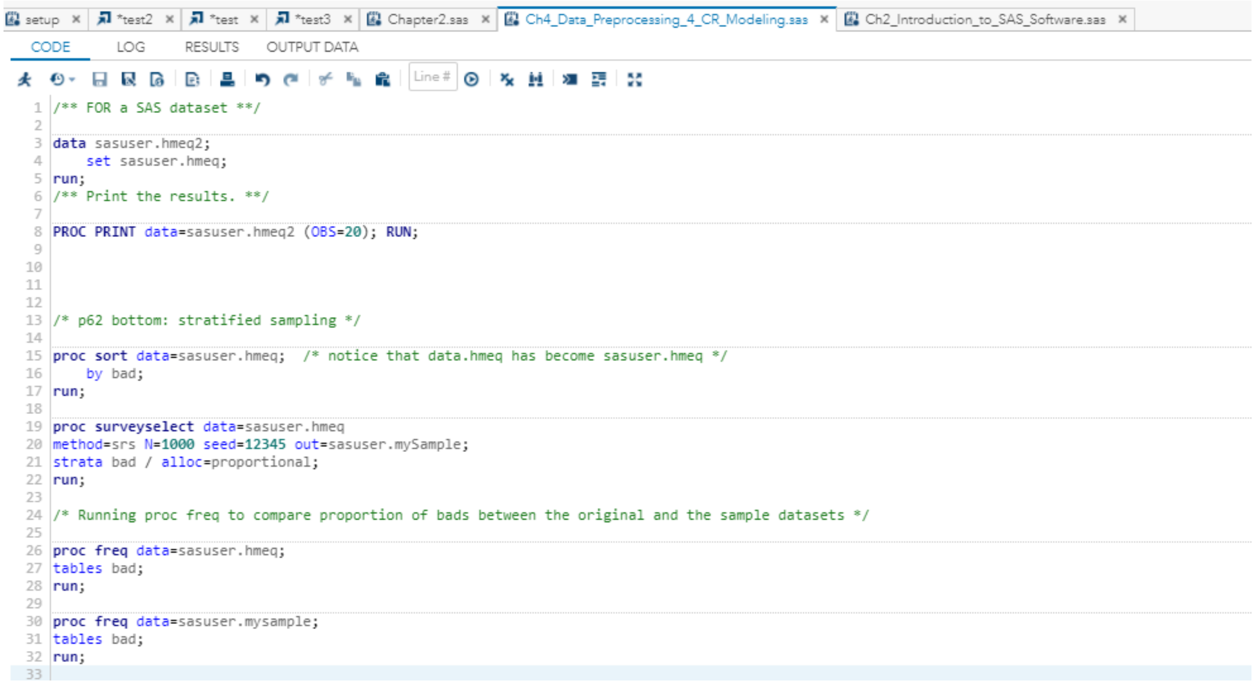
Obs	BAD	LOAN	MORTDUE	VALUE	REASON	JOB	YOJ	DEROG	DELINQ	CLAGE	NINQ	CLNO	DEBTINC
1	1	1100	25860	39025	HomeImp	Other	10.5	0	0	94.367	1	9	.
2	1	1300	70053	68400	HomeImp	Other	7.0	0	2	121.833	0	14	.
3	1	1500	13500	16700	HomeImp	Other	4.0	0	0	149.467	1	10	.
4	1	1500	.	.			.	.	.	.	.	.	.
5	0	1700	97800	112000	HomeImp	Office	3.0	0	0	93.333	0	14	.
6	1	1700	30548	40320	HomeImp	Other	9.0	0	0	101.466	1	8	37.1136
7	1	1800	48649	57037	HomeImp	Other	5.0	3	2	77.100	1	17	.
8	1	1800	28502	43034	HomeImp	Other	11.0	0	0	88.766	0	8	36.8849
9	1	2000	32700	46740	HomeImp	Other	3.0	0	2	216.933	1	12	.
10	1	2000	.	62250	HomeImp	Sales	16.0	0	0	115.800	0	13	.
11	1	2000	22608	.			18.0	.	.	.	.	.	.
12	1	2000	20627	29800	HomeImp	Office	11.0	0	1	122.533	1	9	.
13	1	2000	45000	55000	HomeImp	Other	3.0	0	0	86.067	2	25	.
14	0	2000	64536	87400		Mgr	2.5	0	0	147.133	0	24	.
15	1	2100	71000	83850	HomeImp	Other	8.0	0	1	123.000	0	16	.
16	1	2200	24280	34687	HomeImp	Other	.	0	1	300.867	0	8	.
17	1	2200	90957	102600	HomeImp	Mgr	7.0	2	6	122.900	1	22	.
18	1	2200	23030	.			19.0	.	.	.	.	.	3.7113
19	1	2300	28192	40150	HomeImp	Other	4.5	0	0	54.600	1	16	.
20	0	2300	102370	120953	HomeImp	Office	2.0	0	0	90.993	0	13	31.5885

The resulting log file will show the following – so we know there are no errors:



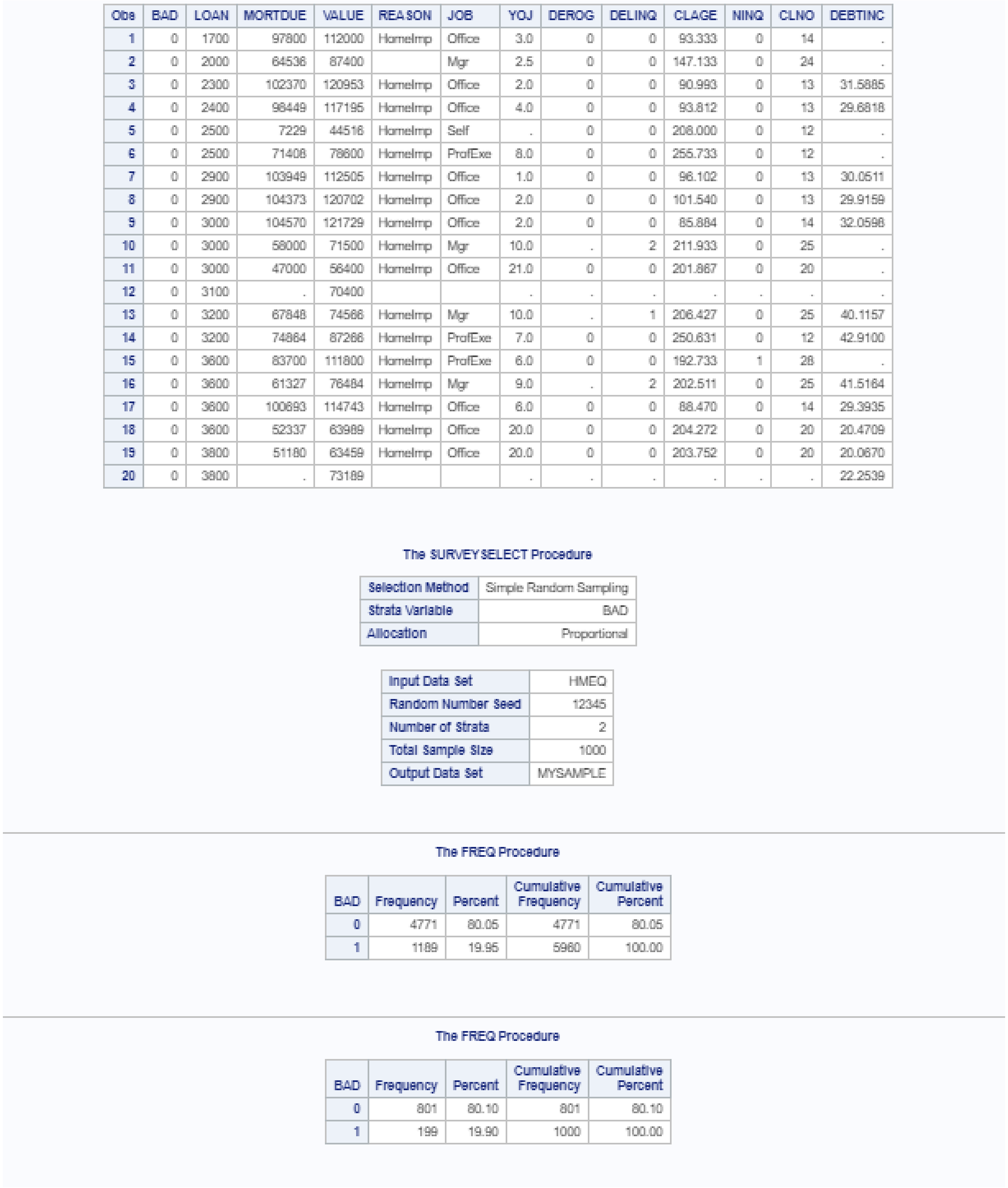
```
 1          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
 70
 71         data sasuser.hmeq2;
 72         set sasuser.hmeq;
 NOTE: Data file SASUSER.HMEQ.DATA is in a format that is native to another host, or the file encoding does not match the session
       encoding. Cross Environment Data Access will be used, which might require additional CPU resources and might reduce
       performance.
 73         run;

 NOTE: There were 5960 observations read from the data set SASUSER.HMEQ.
 NOTE: The data set SASUSER.HMEQ2 has 5960 observations and 13 variables.
 NOTE: DATA statement used (Total process time):
       real time           0.02 seconds
       user cpu time       0.02 seconds
       system cpu time     0.00 seconds
       memory              2094.40k
       OS Memory           35252.00k
       Timestamp           04/21/2018 03:14:26 AM
       Step Count                        141  Switch Count  2
       Page Faults                       0
       Page Reclaims                     305
       Page Swaps                        0
       Voluntary Context Switches        49
       Involuntary Context Switches      0
       Block Input Operations            0
       Block Output Operations           1296


 74         /** Print the results. **/
 75
 76         PROC PRINT data=sasuser.hmeq2 (OBS=20);
 77         RUN;

 NOTE: There were 20 observations read from the data set SASUSER.HMEQ2.
 NOTE: PROCEDURE PRINT used (Total process time):
       real time           0.06 seconds
       user cpu time       0.06 seconds
       system cpu time     0.00 seconds
       memory              2487.90k
       OS Memory           35120.00k
       Timestamp           04/21/2018 03:14:26 AM
       Step Count                        142  Switch Count  0
       Page Faults                       0
       Page Reclaims                     183
       Page Swaps                        0
       Voluntary Context Switches        10
       Involuntary Context Switches      0
       Block Input Operations            288
       Block Output Operations           24
```

Now to run the stratified sampling example pp 62-63


setup
*test2
*test
*test3
Chapter2.sas
Ch4_Data_Preprocessing_4_CR_Modeling.sas
Ch2_Introduction_to_SAS_Software.sas
CODE
LOG
RESULTS
OUTPUT DATA
Line #

```
/** FOR a SAS dataset **/

data sasuser.hmeq2;
    set sasuser.hmeq;
run;
/** Print the results. **/

PROC PRINT data=sasuser.hmeq2 (OBS=20); RUN;




/* p62 bottom: stratified sampling */

proc sort data=sasuser.hmeq;  /* notice that data.hmeq has become sasuser.hmeq */
    by bad;
run;

proc surveyselect data=sasuser.hmeq
method=srs N=1000 seed=12345 out=sasuser.mySample;
strata bad / alloc=proportional;
run;

/* Running proc freq to compare proportion of bads between the original and the sample datasets */

proc freq data=sasuser.hmeq;
tables bad;
run;

proc freq data=sasuser.mysample;
tables bad;
run;
```

Notice the differences in the syntax from the book examples. For example, proc sort data=data.hmeq becomes proc sort data=sasuser.hmeq. The results are as follows and obviously identical to what you have in the book.



Obs	BAD	LOAN	MORTDUE	VALUE	REASON	JOB	YOJ	DEROG	DELINQ	CLAGE	NINQ	CLNO	DEBTINC
1	0	1700	97800	112000	HomeImp	Office	3.0	0	0	93.333	0	14	.
2	0	2000	64536	87400		Mgr	2.5	0	0	147.133	0	24	.
3	0	2300	102370	120953	HomeImp	Office	2.0	0	0	90.993	0	13	31.5885
4	0	2400	98449	117195	HomeImp	Office	4.0	0	0	93.812	0	13	29.6818
5	0	2500	7229	44516	HomeImp	Self	.	0	0	208.000	0	12	.
6	0	2500	71408	78600	HomeImp	ProfExe	8.0	0	0	255.733	0	12	.
7	0	2900	103949	112505	HomeImp	Office	1.0	0	0	96.102	0	13	30.0511
8	0	2900	104373	120702	HomeImp	Office	2.0	0	0	101.540	0	13	29.9159
9	0	3000	104570	121729	HomeImp	Office	2.0	0	0	85.884	0	14	32.0598
10	0	3000	58000	71500	HomeImp	Mgr	10.0	.	2	211.933	0	25	.
11	0	3000	47000	56400	HomeImp	Office	21.0	0	0	201.867	0	20	.
12	0	3100	.	70400			.	.	.	.	.	.	.
13	0	3200	67848	74566	HomeImp	Mgr	10.0	.	1	206.427	0	25	40.1157
14	0	3200	74864	87266	HomeImp	ProfExe	7.0	0	0	250.631	0	12	42.9100
15	0	3600	83700	111800	HomeImp	ProfExe	6.0	0	0	192.733	1	28	.
16	0	3600	61327	76484	HomeImp	Mgr	9.0	.	2	202.511	0	25	41.5164
17	0	3600	100693	114743	HomeImp	Office	6.0	0	0	88.470	0	14	29.3935
18	0	3600	52337	63989	HomeImp	Office	20.0	0	0	204.272	0	20	20.4709
19	0	3800	51180	63459	HomeImp	Office	20.0	0	0	203.752	0	20	20.0670
20	0	3800	.	73189			.	.	.	.	.	.	22.2539

The SURVEYSELECT Procedure

Selection Method	Simple Random Sampling
Strata Variable	BAD
Allocation	Proportional

Input Data Set	HMEQ
Random Number Seed	12345
Number of Strata	2
Total Sample Size	1000
Output Data Set	MYSAMPLE

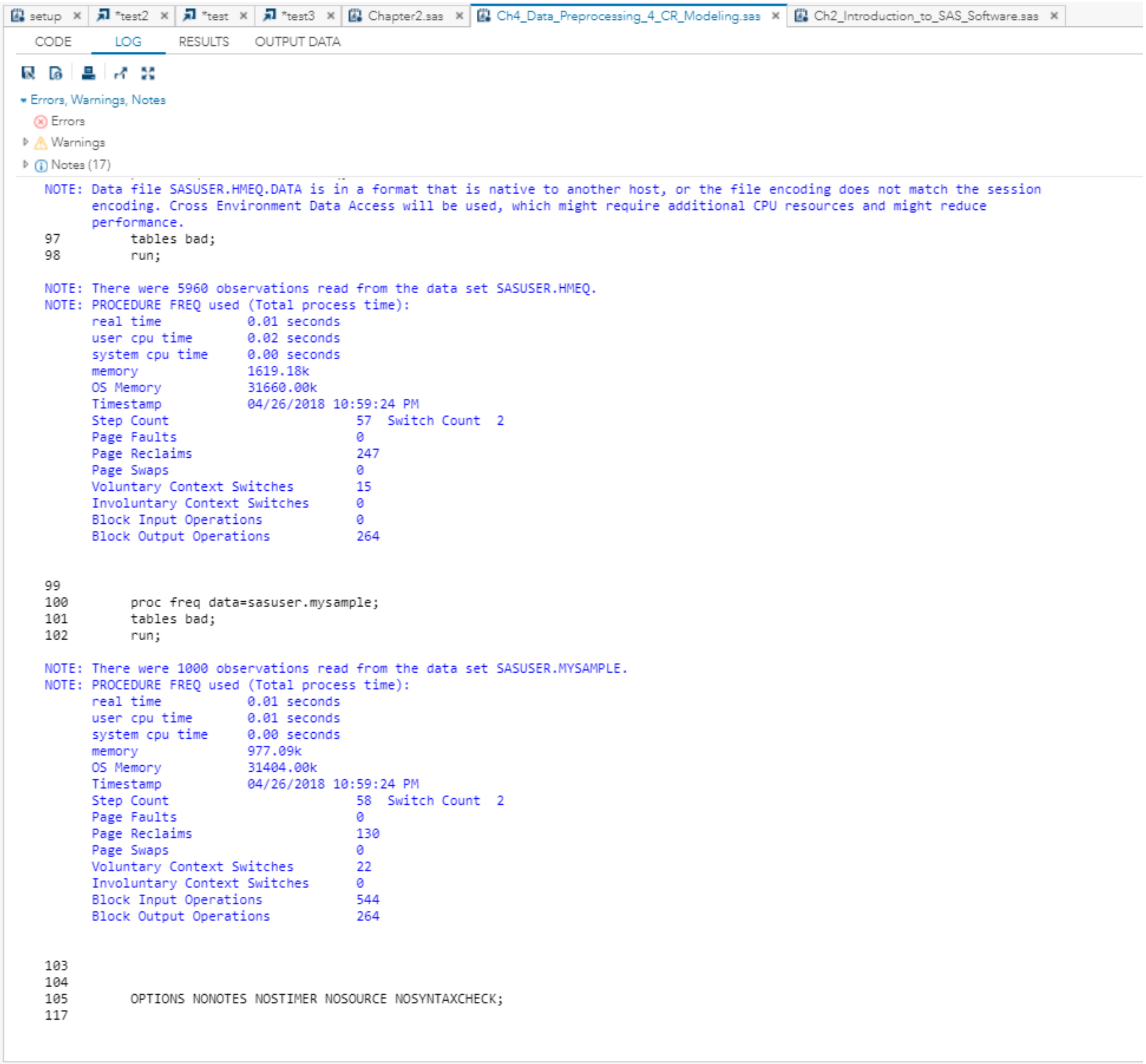
The FREQ Procedure

BAD	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	4771	80.05	4771	80.05
1	1189	19.95	5960	100.00

The FREQ Procedure

BAD	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	801	80.10	801	80.10
1	199	19.90	1000	100.00

And the log file confirms error-free execution



```
NOTE: Data file SASUSER.HMEQ.DATA is in a format that is native to another host, or the file encoding does not match the session
      encoding. Cross Environment Data Access will be used, which might require additional CPU resources and might reduce
      performance.
97          tables bad;
98          run;

NOTE: There were 5960 observations read from the data set SASUSER.HMEQ.
NOTE: PROCEDURE FREQ used (Total process time):
      real time           0.01 seconds
      user cpu time       0.02 seconds
      system cpu time     0.00 seconds
      memory              1619.18k
      OS Memory           31660.00k
      Timestamp           04/26/2018 10:59:24 PM
      Step Count                        57  Switch Count  2
      Page Faults                       0
      Page Reclaims                     247
      Page Swaps                        0
      Voluntary Context Switches        15
      Involuntary Context Switches      0
      Block Input Operations            0
      Block Output Operations           264


99
100         proc freq data=sasuser.mysample;
101         tables bad;
102         run;

NOTE: There were 1000 observations read from the data set SASUSER.MYSAMPLE.
NOTE: PROCEDURE FREQ used (Total process time):
      real time           0.01 seconds
      user cpu time       0.01 seconds
      system cpu time     0.00 seconds
      memory              977.09k
      OS Memory           31404.00k
      Timestamp           04/26/2018 10:59:24 PM
      Step Count                        58  Switch Count  2
      Page Faults                       0
      Page Reclaims                     130
      Page Swaps                        0
      Voluntary Context Switches        22
      Involuntary Context Switches      0
      Block Input Operations            544
      Block Output Operations           264


103
104
105         OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
117
```

Conclusion:

The above is a quick overview of some tips and techniques that will work for you if you wish to actually try out the code samples in the book in the SAS environment if you are familiar with the language but do not have access to a licensed edition at all times. Of course, there is freeware such as Python on R, but there is a learning curve to these languages and sometimes limitations on data volumes (as in the case of R) and a dependency on packages that are often black-boxes from little-known purveyors. If you wish to focus instead on the credit risk analytics side of it and be assured of the accuracy that comes with a highly reputable statistical software that has been around for a long time.