

(QUICK) INTRODUCTION TO SAS

ARTHUR ALLIGNOL

INTRODUCTION

WHAT IS SAS

- Developed in the early 1970s
- Used to stand for "Statistical Analysis System"
- Still a (the) prominent player in the industry

Pros and cons (according to myself)

- Good data handling (especially big data)
- Rich documentation
- Decent IDE
- Expensive
- Awful graphics (without giving more \$\$)
- Awful lot of outputs
- Real programming cumbersome (SAS developed in a time when there were still punchcards)

WHAT IS SAS

You must realize that R is written by experts in statistics and statistical computing who, despite popular opinion, do not believe that everything in SAS and SPSS is worth copying. Some things done in such packages, which trace their roots back to the days of punched cards and magnetic tape when fitting a single linear model may take several days because your first 5 attempts failed due to syntax errors in the JCL or the SAS code, still reflect the approach of "give me every possible statistic that could be calculated from this model, whether or not it makes sense". The approach taken in R is different. The underlying assumption is that the user is thinking about the analysis while doing it.

BASIC STRUCTURE OF A SAS PROGRAM

Two main components to most SAS programs

1. The data step(s):

- reads data from external sources, manipulates and combines it with other data sets

2. The procedure step(s):

- The procedure steps perform the analysis on the data, and produce (often huge amounts of) output.
- SAS is not case-sensitive (contrary to R)
- Missing values represented by a period (.)
- Each statement in SAS must end in a semicolon

SAS IDE

The screenshot displays the SAS Studio web interface. The top navigation bar includes the 'SAS Information Center' and 'SAS Studio' tabs. The address bar shows the URL: `localhost:10080/SASStudio/35/main?locale=en_US&zone=GMT%252B02%253A00&http%3A%2F%2Flocalhost%3A10080%2FSASStudio%2F35%2F=`. The left sidebar, titled 'Server Files and Folders', contains a 'Folder Shortcuts' section with a tree view of the project structure. The 'My Folders' section shows the 'sasuser.v94' folder. The main editor area, titled 'SAS Studio', displays a SAS code file named 'SASIntro.sas'. The code is written in SAS syntax and includes various data manipulation and output statements. The status bar at the bottom right indicates 'Line 80, Column 5' and 'User: sasdemo'.

Server Files and Folders

- Folder Shortcuts
 - arthur
 - WiMa_Praktikum
 - admin
 - lectures
 - figure
 - graphics
 - illustration
 - figure
 - knitr_intro.aux
 - knitr_intro.log
 - knitr_intro.pdf
 - knitr_intro.Rnw
 - knitr_intro.tex
 - knitr_pdfjpeg
 - SASIntro.sas
 - sleep.csv
 - test.html
 - test.rtf
 - test2.html

- My Folders
- sasuser.v94

Tasks and Utilities

- Snippets
- Libraries
- File Shortcuts

SAS Studio

CODE LOG RESULTS

```
33 data sleep_2;
34   set sleep(where = (group eq '2'));
35   run;
36
37 data sleep_short;
38   set sleep;
39   drop group_with_no_meaning;
40   run;
41
42 data sleep_short;
43   set sleep_short;
44   keep id group extra;
45   run;
46
47 proc contents data=sleep;
48   run;
49
50 proc print data=sleep(where = (group = "1"));
51   var group extra;
52   run;
53
54
55 ods html file = '/folders/myshortcuts/WiMa_Praktikum/lectures/illustration/test.html';
56
57 proc means data = sleep;
58   var extra;
59   by group;
60   run;
61
62 ods html close;
63
64 ods trace on;
65 proc univariate data = sleep;
66   var extra;
67   run;
68 ods trace off;
69
70
71
72 ods output Univariate.extra.BasicMeasures = basic;
73 proc univariate noprint data = sleep;
74   var extra;
75   run;
76 ods output close;
77
78 ods rtf file = '/folders/myshortcuts/WiMa_Praktikum/lectures/illustration/test.rtf';
79 proc print data = basic;
80   run;
81 ods rtf close;
82
83
```

Line 80, Column 5

Messages: 24 User: sasdemo

READ DATA

PROC IMPORT

```
proc import datafile='/folders/myshortcuts/WiMa_Praktikum/lectures/illustration/sleep.csv'  
  out=sleep  
  dbms=csv  
  replace;  
run;
```

- `datafile` specifies the path to the data set
- `out`: name of the data set in SAS
- `dbms`: file type
- `replace` replace the `sleep` data set if `proc import` is rerun
- The `run;` command signals to SAS that the previous commands can be executed

PROC IMPORT

Total rows: 20 Total columns: 3

	extra	group	ID	
1	0.7	1	1	
2	-1.6	1	2	
3	-0.2	1	3	
4	-1.2	1	4	
5	-0.1	1	5	
6	3.4	1	6	
7	3.7	1	7	
8	0.8	1	8	
9	0	1	9	
10	2	1	10	
11	1.9	2	1	
12	0.8	2	2	
13	1.1	2	3	
14	0.1	2	4	
15	-0.1	2	5	
16	4.4	2	6	
17	5.5	2	7	
18	1.6	2	8	
19	4.6	2	9	
20	3.4	2	10	

IMPORT DATA IN A DATA STEP

```
data sleep2;  
  infile '/folders/myshortcuts/WiMa_Praktikum/lectures/illustration/sleep.csv'  
  dsd  
  delimiter = ','  
  firstobs = 2;  
input extra group id;
```

- `infile` specifies where the file is
- `dsd` is an option to handle missing values, e.g., 30 , , 40 , 50
- `delimiter` specifies how the data are separated, in this case with a ,
- `firstobs` tells SAS to start reading the data at the 2nd line
- The `input` statement gives the variable names

THE DATA STEP

INTRODUCTION

Data steps are made up of programming statements, e.g.,

- Assignment
- conditional operations
- subsetting

The Data step always begin with the keyword `data` followed by the name you want to give to the resulting data set

- Optionally the `set` statement permits to process an existing data set

OPERATORS

Arithmetic operators

*	multiplication	+	addition	/	division
-	subtraction	**	exponentiation		

Comparison operators

= or eq	equal to	^= or ne	not equal to
> or gt	greater than	>= or ge	greater than or equal to
< or lt	less than	<= or le	less than or equal to

Boolean Operators

& or and and | or or or ^ or not negations

ASSIGNMENT

Assignment statement assign values to new or existing variables. These values may be

- A constant
- Another variable
- The results of mathematical expressions

```
data sleep;  
  set sleep;  
  a_variable = 2;  
  extra_1 = extra ** 2;  
  extra_2 = log(extra_1);  
  extra_3 = exp(extra);  
run;
```

CONDITIONAL OPERATIONS

```
data sleep;  
  set sleep;  
  if extra < 0 and group eq 2 then group_with_no_meaning = 'A';  
  else group_with_no_meaning = 'B';  
run;
```

Total rows: 20 Total columns: 7

 Rows 1-20

extra	group	ID	group_with_no_meaning	extra_1	extra_2	extra_3
0.7	1	1	B	0.49	-0.713349888	2.0137527075
-1.6	1	2	B	2.56	0.9400072585	0.201896518
-0.2	1	3	B	0.04	-3.218875825	0.8187307531
-1.2	1	4	B	1.44	0.3646431136	0.3011942119
-0.1	1	5	B	0.01	-4.605170186	0.904837418
3.4	1	6	B	11.56	2.4475508632	29.964100047
3.7	1	7	B	13.69	2.6166656393	40.44730436
0.8	1	8	B	0.64	-0.446287103	2.2255409285
0	1	9	B	0	.	1
2	1	10	B	4	1.3862943611	7.3890560989
1.9	2	1	B	3.61	1.2837077723	6.6858944423
0.8	2	2	B	0.64	-0.446287103	2.2255409285
1.1	2	3	B	1.21	0.1906203596	3.0041660239
0.1	2	4	B	0.01	-4.605170186	1.1051709181
-0.1	2	5	A	0.01	-4.605170186	0.904837418
4.4	2	6	B	19.36	2.9632090818	81.450868665
5.5	2	7	B	30.25	3.4094961845	244.69193226
1.6	2	8	B	2.56	0.9400072585	4.9530324244
4.6	2	9	B	21.16	3.052112607	99.484315642
3.4	2	10	B	11.56	2.4475508632	29.964100047

SUBSETTING

Subsetting if

```
data sleep_1;  
  set sleep;  
  if group eq 1 then delete;  
run;
```

where statement

```
data sleep_2;  
  set sleep(where = (group eq '2'));  
run;
```

drop or keep variables

```
data sleep_short;  
  set sleep;  
  drop group_with_no_meaning;  
run;
```

```
data sleep_short;  
  set sleep_short;  
  keep id group extra;  
run;
```

SAS PROCEDURES

SAS PROCEDURES

SAS procedures begin with the keyword `proc` followed by the name of the procedure and the name of the data you want to use in the procedure.

```
proc contents data=sleep;  
run;
```

The CONTENTS Procedure

Data Set Name	WORK.SLEEP	Observations	20
Member Type	DATA	Variables	7
Engine	V9	Indexes	0
Created	04/11/2016 11:48:01	Observation Length	40
Last Modified	04/11/2016 11:48:01	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	SOLARIS_X86_64, LINUX_X86_64, ALPHA_TRU64, LINUX_IA64		
Encoding	utf-8 Unicode (UTF-8)		

Engine/Host Dependent Information	
Data Set Page Size	65536
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	1632
Obs in First Data Page	20
Number of Data Set Repairs	0
Filename	/tmp/SAS_work80B600001639_localhost.localdomain/SAS_work2C1700001639_localhost.localdomain/sleep.sas7bdat
Release Created	9.0401M3
Host Created	Linux
Inode Number	143322
Access Permission	rw-rw-r--
Owner Name	sasdemo
File Size	128KB
File Size (bytes)	131072

Alphabetic List of Variables and Attributes					
#	Variable	Type	Len	Format	Informat
3	ID	Char	4	\$4.	\$4.
1	extra	Num	8	BEST12.	BEST32.
5	extra_1	Num	8		
6	extra_2	Num	8		
7	extra_3	Num	8		
2	group	Char	3	\$3.	\$3.
4	group_with_no_meaning	Char	1		

SAS PROCEDURES

```
proc print data=sleep(where = (group = "1"));  
  var group extra;  
run;
```

Obs	group	extra
1	1	0.7
2	1	-1.6
3	1	-0.2
4	1	-1.2
5	1	-0.1
6	1	3.4
7	1	3.7
8	1	0.8
9	1	0
10	1	2

OUTPUT DELIVERY SYSTEM (ODS)

WHAT IS ODS

To provide more flexibility in producing outputs, SAS introduced the ODS. Output can be produced in the following format

- SAS data set (OUTPUT)
- Normal listing (LISTING)
- Postscript (PRINTER)
- PDF (PDF)
- HTML (HTML)
- RTF (RTF)

Procedures produce ODS objects (find their names using the help pages or
ODS TRACE ON

AN EXAMPLE

```
ods html file = 'test.html';  
  
proc means data = sleep;  
  var extra;  
  by group;  
  run;  
  
ods html close;
```


ONLY OUTPUT A PART OF THE RESULTS

First search what you want with `ods trace on`

```
ods trace on;  
proc univariate data = sleep;  
    var extra;  
run;  
ods trace off;
```

then output the part you want

```
ods output Univariate.extra.BasicMeasures = basic;  
proc univariate data = sleep;  
    var extra;  
run;
```

and print to a file

```
ods rtf file = '/folders/myshortcuts/WiMa_Praktikum/lectures/illustration/test.rtf';  
proc print data = basic;  
run;  
ods rtf close;
```