SAS – short introduction

Fall 2015

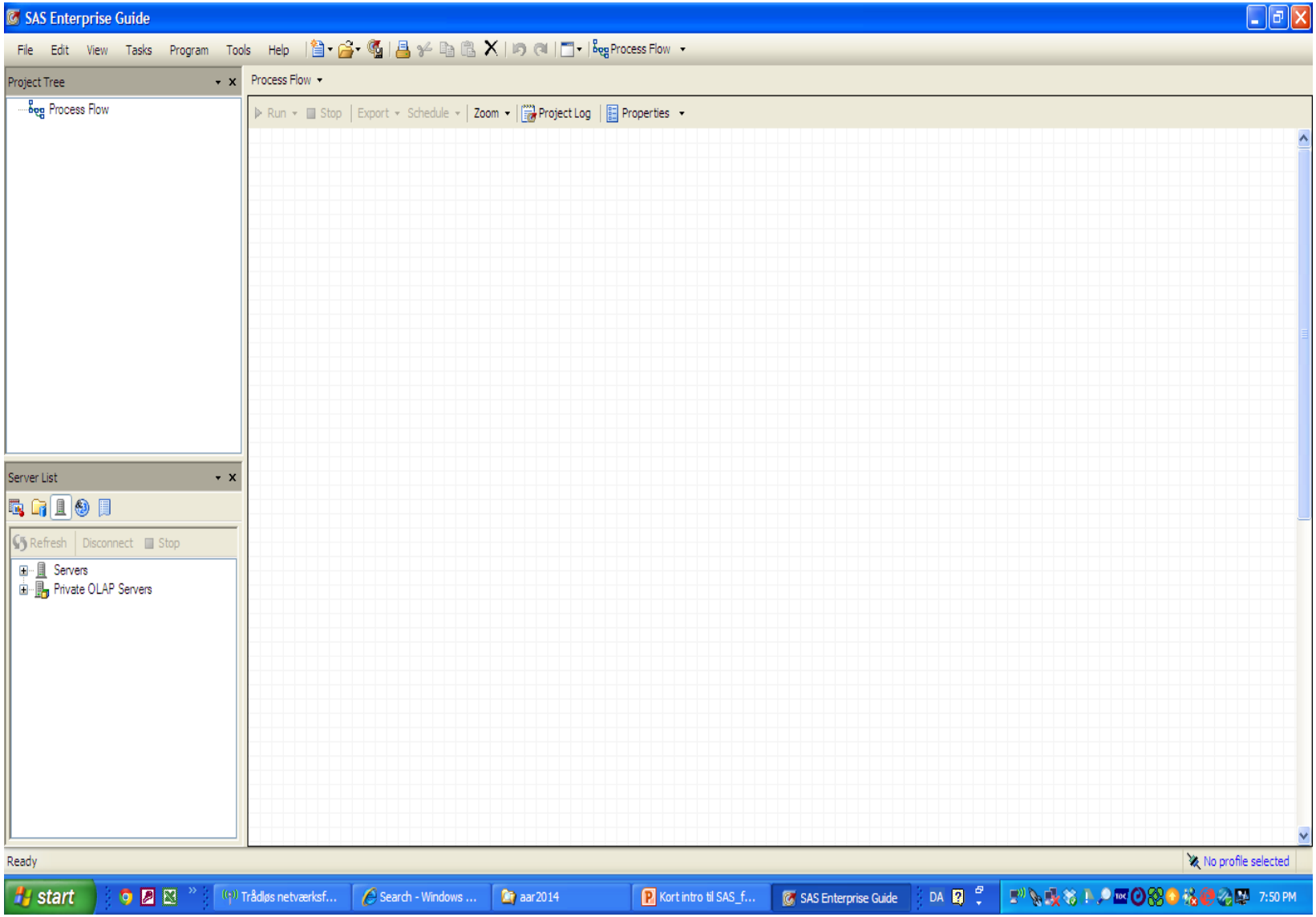
Janne Petersen
Judith L Jacobsen
Lene Theil Skovgaard

Why SAS?

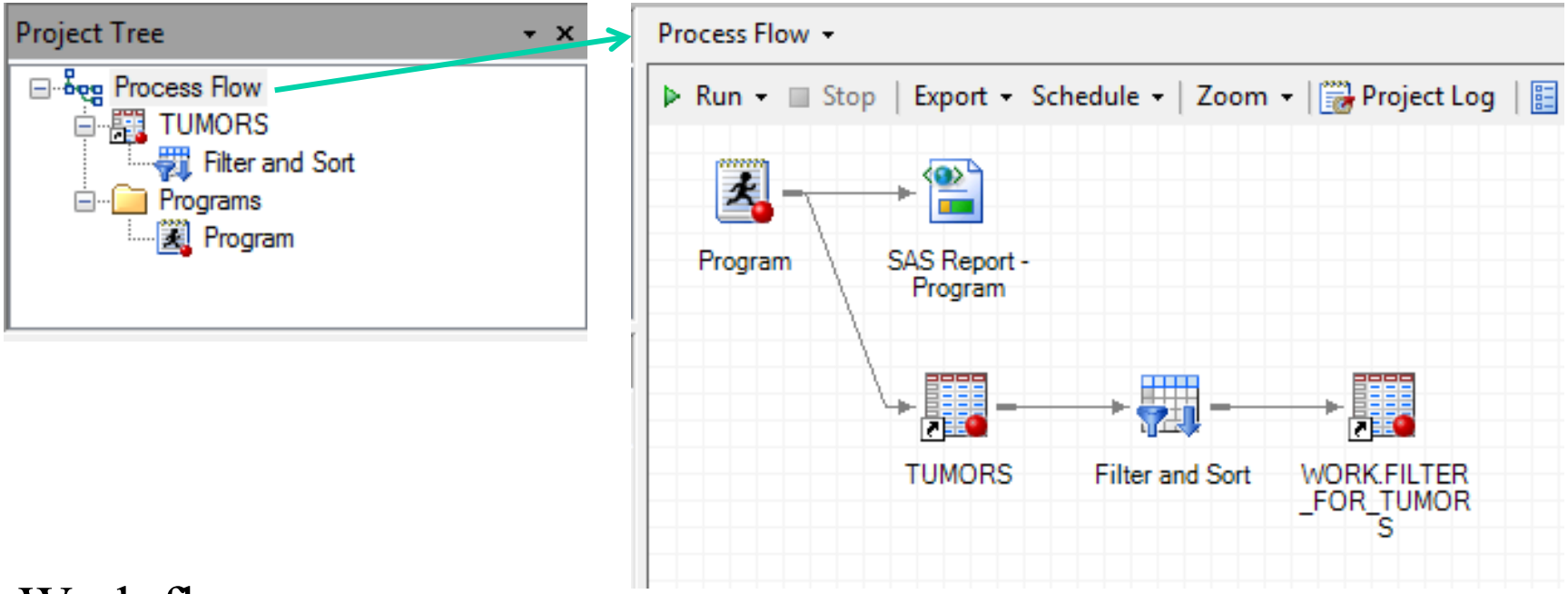
- ”Anything goes”
- Free license for all PhD students
  - university and hospitals
- Real programming---with aid from Enterprise Guide

Reasonable alternatives: R, STATA or SPSS

Enterprise Guide



Your ”office desk”



Work flow
Programs
Datasets
Actions

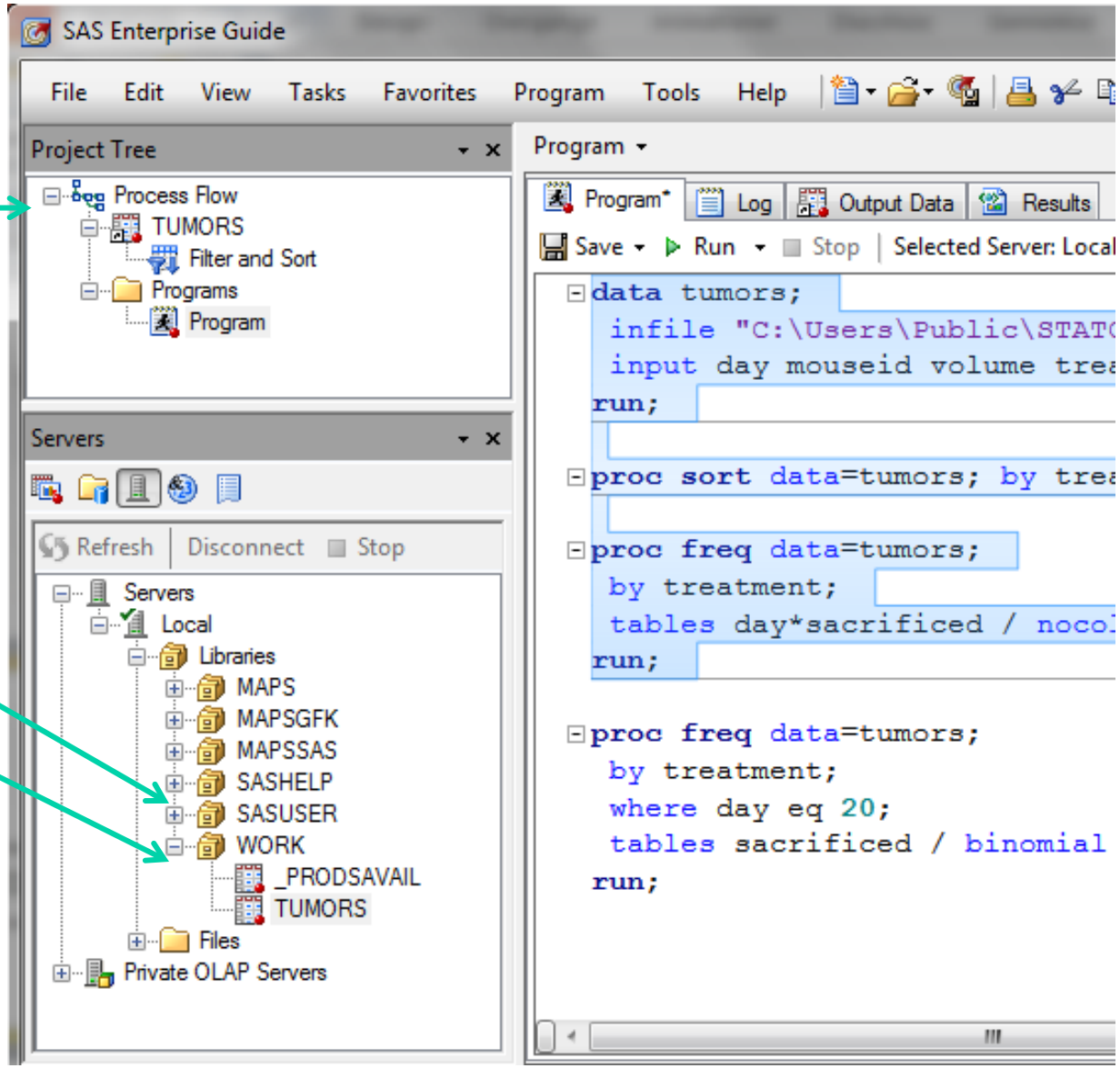
Structure of folders etc.

Process flow
Programs

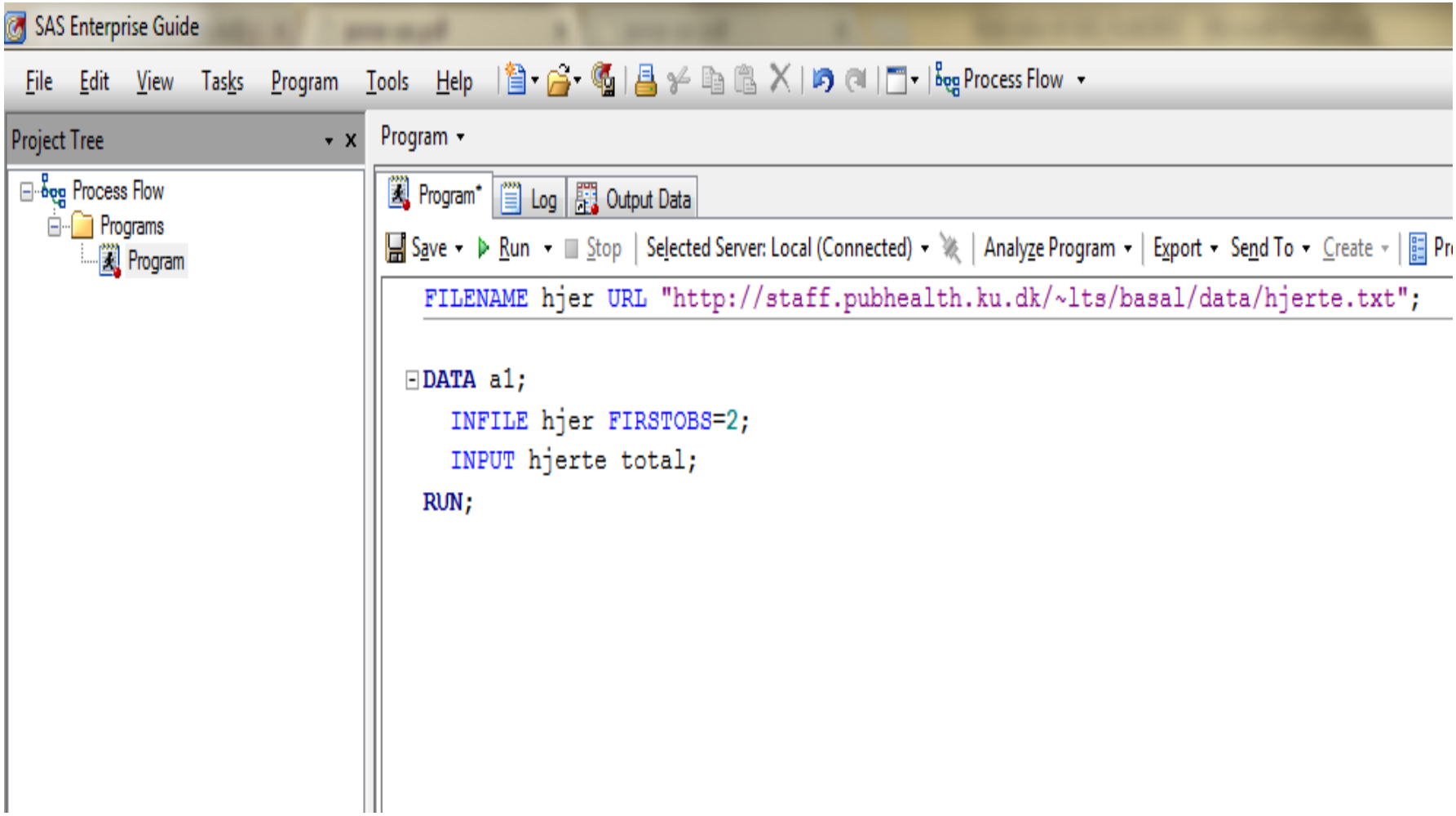
Files and folders

Save a dataset permanently in the SASUSER library

Work folders are temporary

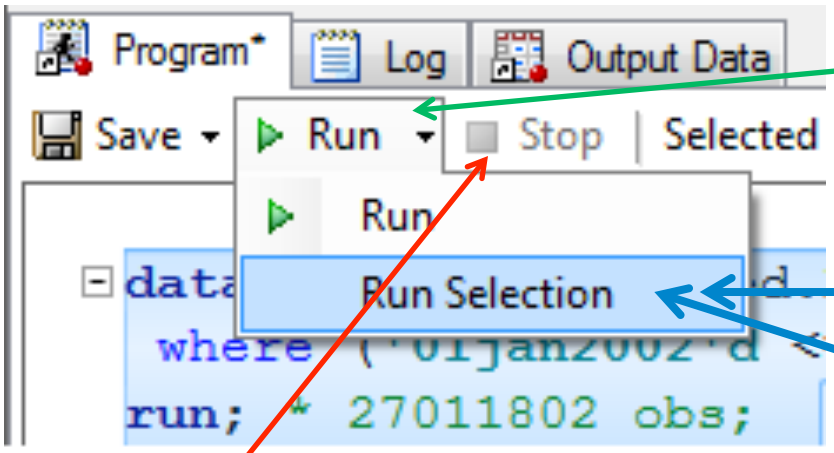


The program editor



Here we write the programs that SAS will execute.

To run a program



Run the entire program

Drop down menu OR

F8 will run only the highlighted part.

Stop: aborts the execution

Red while ”running” and otherwise grey

The log

Program* | Log | Output Data

Export | Send To | Create | Log Summary | Project Log | Properties

```
29          GOPTIONS ACCESSIBLE;
30          FILENAME hjer URL "http://staff.pubhealth.ku.dk/~lts/basal/data/hjerte.txt";
31
32          DATA a1;
33            INFILE hjer FIRSTOBS=2;
34            INPUT hjerte total;
35          RUN;

NOTE: The infile HJER is:
      Filename=http://staff.pubhealth.ku.dk/~lts/basal/data/hjerte.txt,
      Local Host Name=HVHLA1768,
      Local Host IP addr=fe80::80c8:7821:52d9:47e4%14,
      Service Hostname Name=pc059.sund.ku.dk,
      Service IP addr=192.38.117.59,
      Service Name=httpd,Service Portno=80,
      Lrecl=32767,Recfm=Variable

NOTE: 21 records were read from the infile HJER.
      The minimum record length was 8.
      The maximum record length was 8.
NOTE: The data set WORK.A1 has 21 observations and 2 variables.
NOTE: DATA statement used (Total process time):
      real time           0.19 seconds
      cpu time            0.01 seconds
```

Check that data were read succesfully

The log

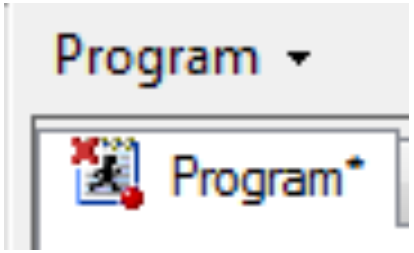
Notes (green) Information about data steps and analyses SAS has performed. E.g. number of observationer read from a datafile.

Warnings (turquoise) Errors that SAS has fixed ---check that this was done correctly!

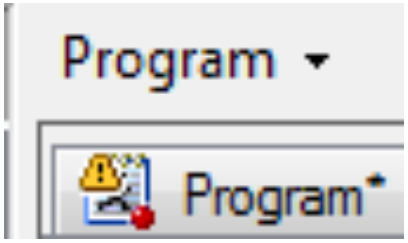
Error (red) Syntax error, that SAS couldn't fix. Hence the program was not executed.

Indication of errors

- The red x means that errors were found in the program while attempting execution.



- The yellow triangle means that warnings were printed during execution



The output data

Program

Program* | Log | Output Data | Results

Filter and Sort | Query Builder | Data | Describe | Graph | Analyze

	day	mouseid	volume	treatment	sacrificed
1	1	51	27.2	contr	0
2	4	51	38	contr	0
3	6	51	78.7	contr	0
4	8	51	83.2	contr	0
5	11				0
6	13				0
7	15				0
8	18				0
9	20				0
10	22				0
11	25				0
12	27				0
13	29	51	589.5	contr	0
14	32	51	992.2	contr	0
15	34	51	.	contr	1
16	36	51	.	contr	1
17	39	51	.	contr	1
18	1	52	178	contr	0

brthdtn
1946-01-01
1945-01-01
1951-01-01
1943-01-01
1964-01-01
1948-01-01
1948-01-01
1954-01-01
1955-01-01
1957-01-01
1940-01-01
1950-01-01
1961-01-01
1954-01-01
1946-01-01
1939-01-01
1955-01-01
1948-01-01

Icons correspond to different types of data:
Text string
Date
Numerical variable

Variables

Numerical variables are always numbers.

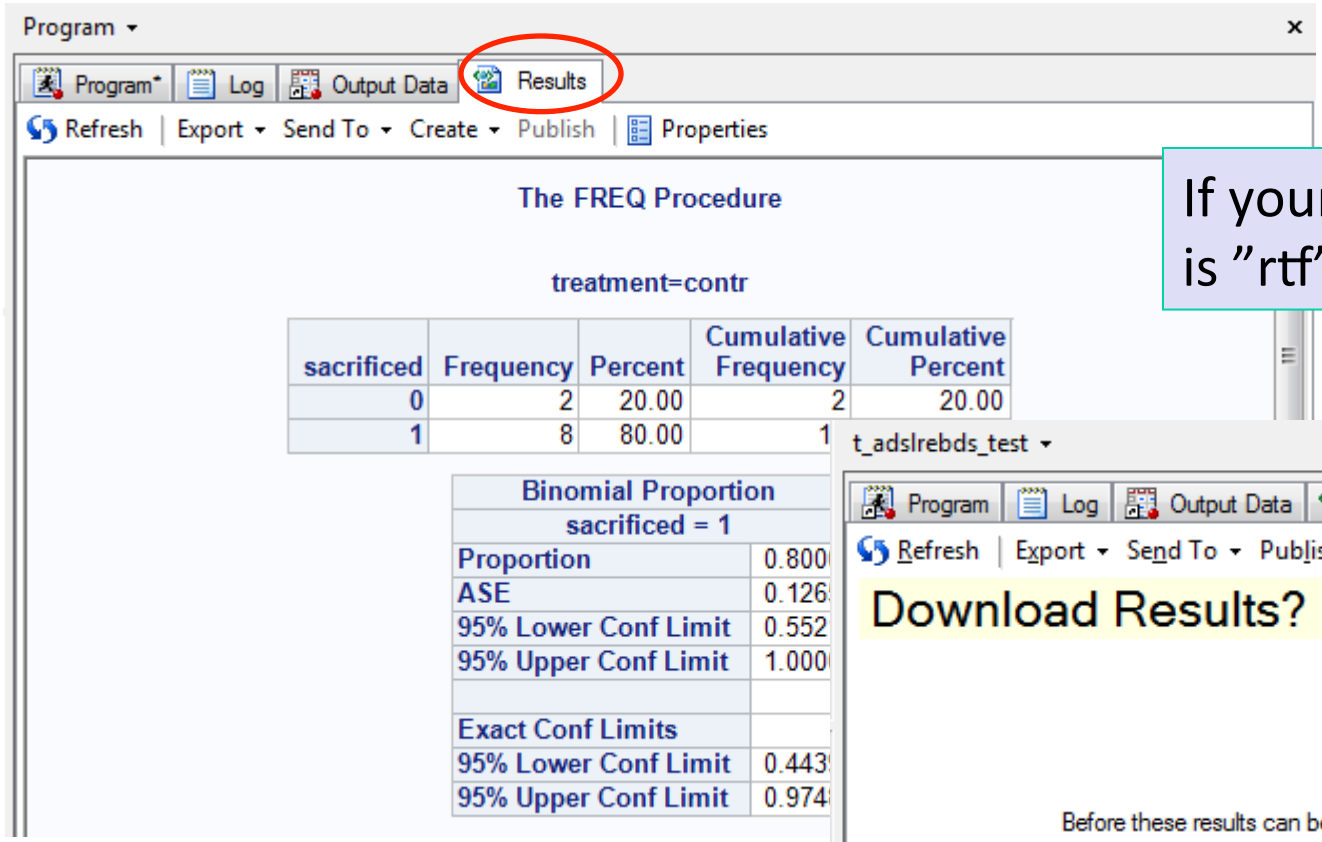
E.g. Age with values 45, 37 …

Averages etc can be computed when a variable is numerical

Text string variables usually contain text but may hide essentially numerical information.

E.g. Month with values ”Jan”, ”Feb” ...

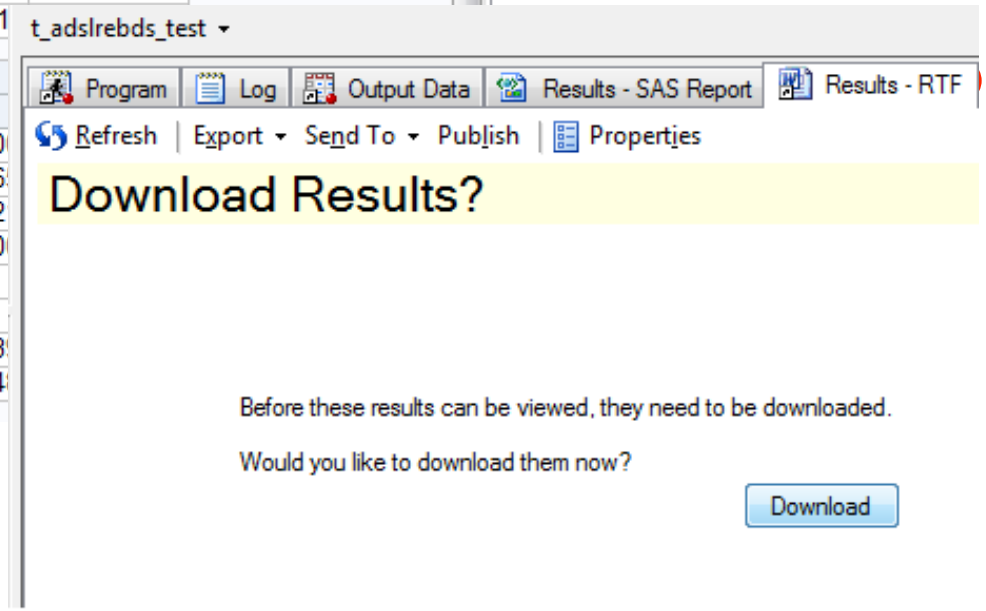
Results



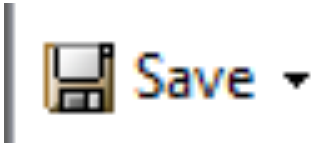
sacrificed	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	2	20.00	2	20.00
1	8	80.00	1	

Binomial Proportion	
sacrificed = 1	
Proportion	0.800
ASE	0.126
95% Lower Conf Limit	0.552
95% Upper Conf Limit	1.000
Exact Conf Limits	
95% Lower Conf Limit	0.443
95% Upper Conf Limit	0.974

If your output destination is ”rtf” you find it here.



Saving your work

- Shift Ctrl S (or: 

saves the program (or the log, or…)
- Ctrl S

  saves the entire project

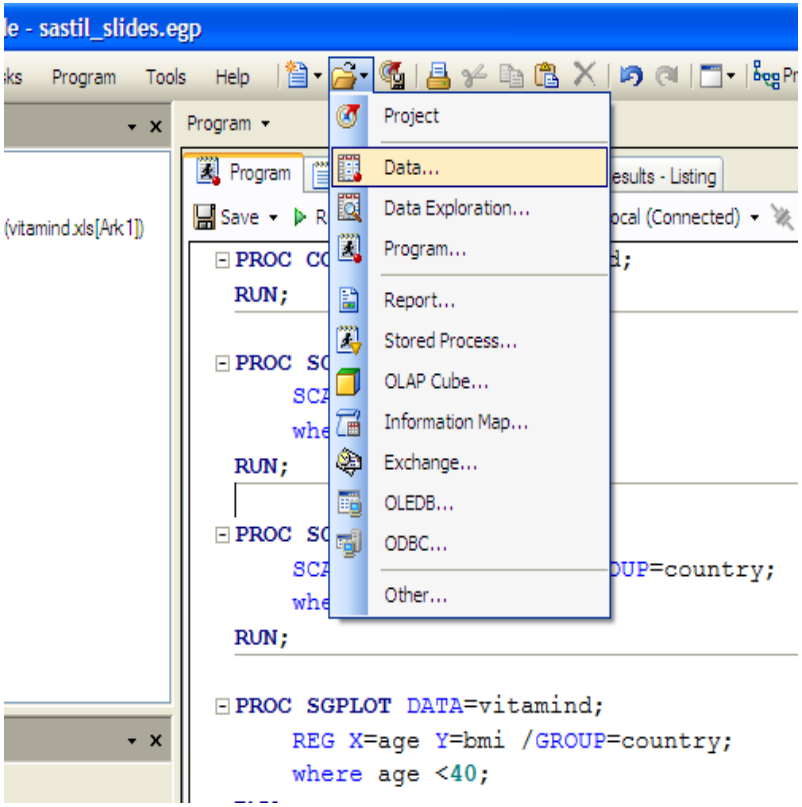
  -work flow,

  -(temporary) datasets,

  -output, etc. – i.e. EVERYTHING.

Importing data: Method 1

1. Download the data file from the course webpage.

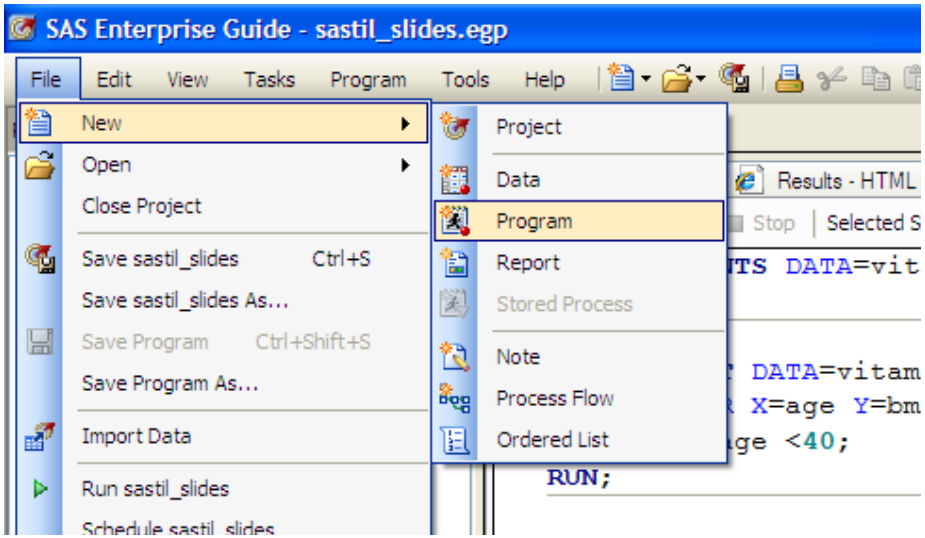


2. Choose Open -> Data to import SAS datasets (.sas7bdat), text files, Excel files etc.

Code for a program that imports the dataset is automatically generated

Importing data: Method 2

Write a new program which says:



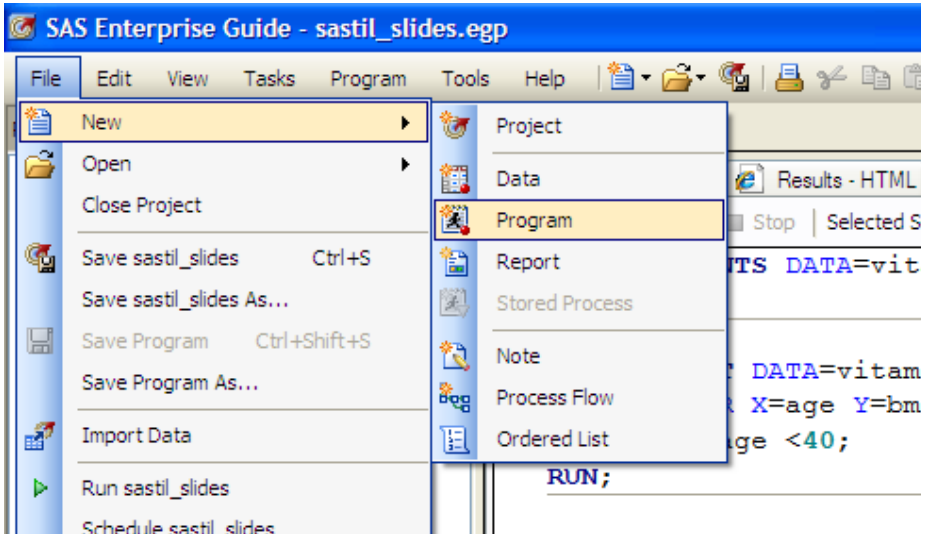

```
FILENAME hearts URL "http://staff.pubhealth.ku.dk/
~lts/basal/data/hjerte.txt";

DATA a1;
  INFILE hearts FIRSTOBS=2;
  INPUT heart total;
RUN;
```

This only works if WIFI does!!!

Name all variables in the datafile and in the CORRECT ORDER

Importing data: Method 3

Write a new program which says:




```
DATA a1;
  INFILE "C:\courses\statistics\hearts.txt" FIRSTOBS=2;
  INPUT heart total;
RUN;
```

Name all the variables in the datafile in the correct order.

This works if you have downloaded the data and stored it in this location (no WIFI needed).

SAS programming

SAS programs consist of two types of steps

DATA (data step):

1. To import a dataset
2. To add new or delete old variables
3. To store the result in a new dataset

PROC (procedure step):

1. To produce summaries of the data; tables, plots and results of statistical analyses

Data step

```
DATA SASUSER.new;
  SET WORK.old;
RUN;
```

Copy the dataset ”old” from work library and store it in the identical dataset ”new” in the sasuser library.

Don’t forget to put **;** at the end of each line.

- WORK is temporary
- SASUSER is permanent

Useful comments

Comments /* like this */ (shortcut : CTRL *)
are skipped by SAS during execution (i.e. no error)

Example:

```
DATA new;
   SET WORK.old; /*work could be dropped*/
RUN;
```

Adding new variables

```
DATA new;
  SET old;
  BMI=weight/(height**2);
RUN;
```

The dataset ”old” should contain the variables weight and height. The datasæt ”new” contains an additional variable: BMI

ATT: Missing values

```
DATA new;
  SET old;
 BMI=weight/(height**2);
 IF .z<BMI<24 THEN BMI2gr=1;
 IF BMI>=24 THEN BMI2gr=2;
RUN;
```

The new variable BMI2gr categorises BMI into two groups (above or below 24). Use ".z<" to ensure that missing values remain missing (SAS reads missing values as very large negative values)

Operators

=	EQ	Equal to
^=	NE	Not equal to
<	LT	Less than
>	GT	Greater than
<=	LE	Less than or equal to
>=	GE	Greater than or equal to
	IN	Is in (a set of values)
&	AND	And
|	OR	Or
^	NOT	Negation (opporsite)

Calculations

*	Multiplication
/	Division
+	Addition
-	Subtraction
**	Raise to power a**b = a^b
Exp(variable)	The exponential function
Log(variable)	The natural logarithm
Log2(variable)	Base 2 logarithm
Log10(variable)	Base 10 logarithm
ABS(variable)	Absolute value
ROUND(variable)	Rounded value

Procedure steps

SAS contains a vast number of procedure that has been developed over many years and by many teams of programmers.

Similar procedures may have different syntax!

But all procedure steps begin with PROC.

Overall procedure syntax

```
PROC procname DATA=dataname;
        (specific part of program)
RUN;
```

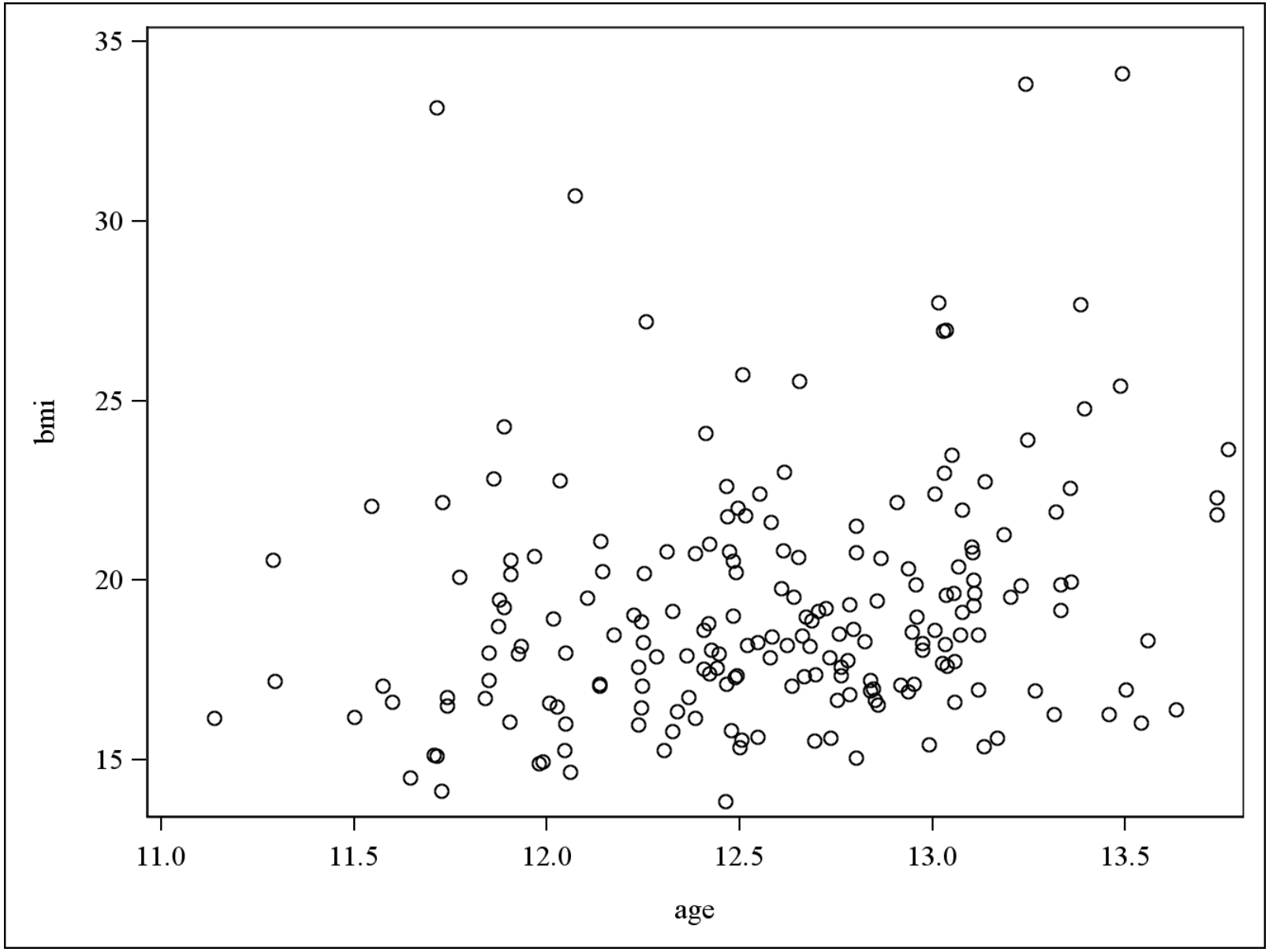
Always start by naming the relevant procedure and the data to be analysed.

Scatter plots

```
PROC SGPLOT DATA=vitamind;
  SCATTER X=age Y=bmi;
RUN;
```

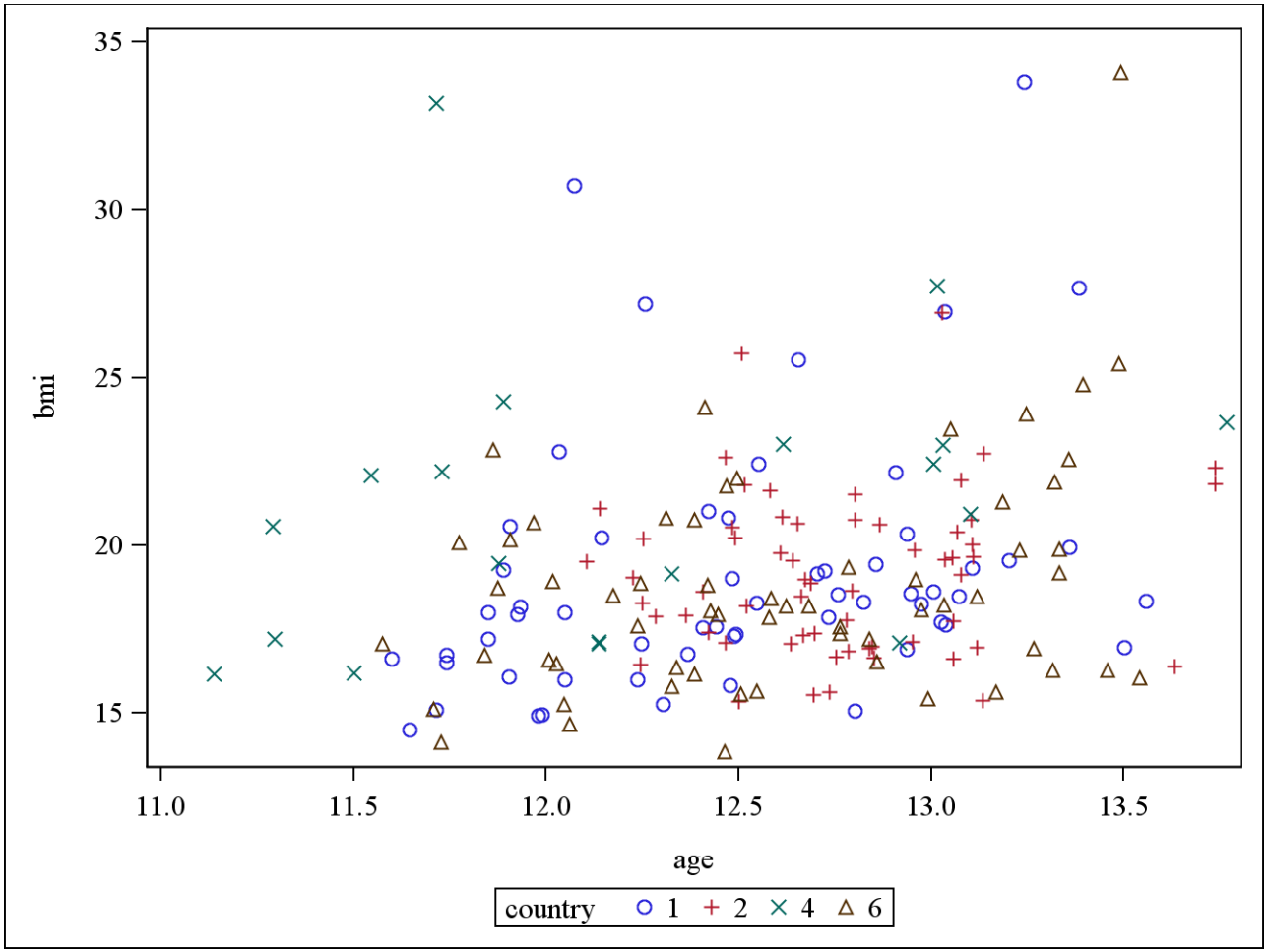
Plots bmi against age.

Remember to put the predictor on the X-axis and the outcome on the Y-axis.


35
30
25
20
15
bmi
11.0
11.5
12.0
12.5
13.0
13.5
age

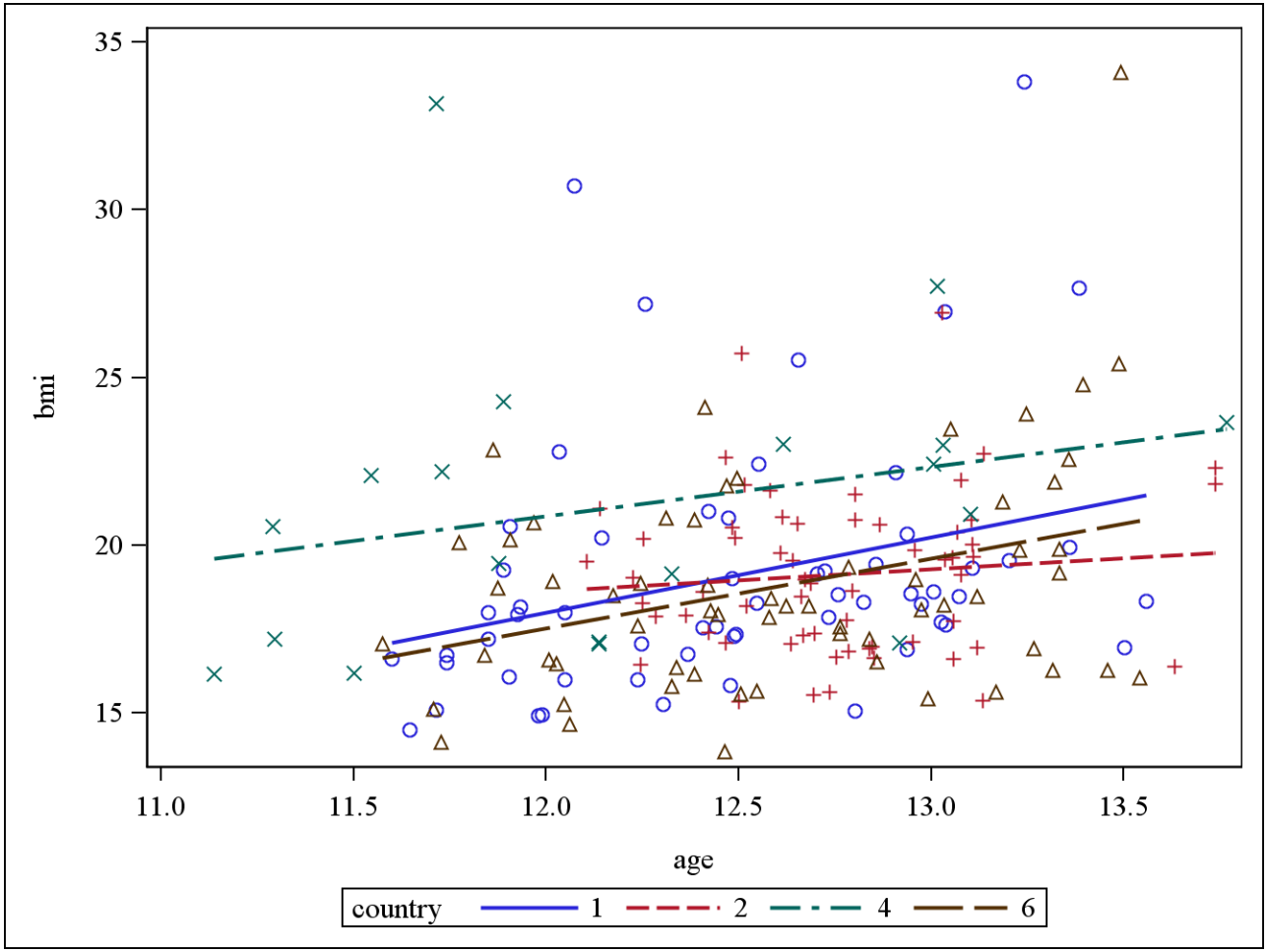
Scatter plot with groupings

```
PROC SGPLOT DATA=vitamind;
  SCATTER X=age Y=bmi/GROUP=country;
RUN;
```



Scatter plot with regression

```
PROC SGPLOT DATA=vitamind;
  REG X=age Y=bmi / GROUP=country;
RUN;
```



SAS help

- Google is often very helpful
- Enterprise Guide editor suggestions
- Press F1 for help
- http://support.sas.com/documentation/onlinedoc/base/

SAS online courses

- http://www.ats.ucla.edu/stat/seminars/

Movies are recommended

DIY pages

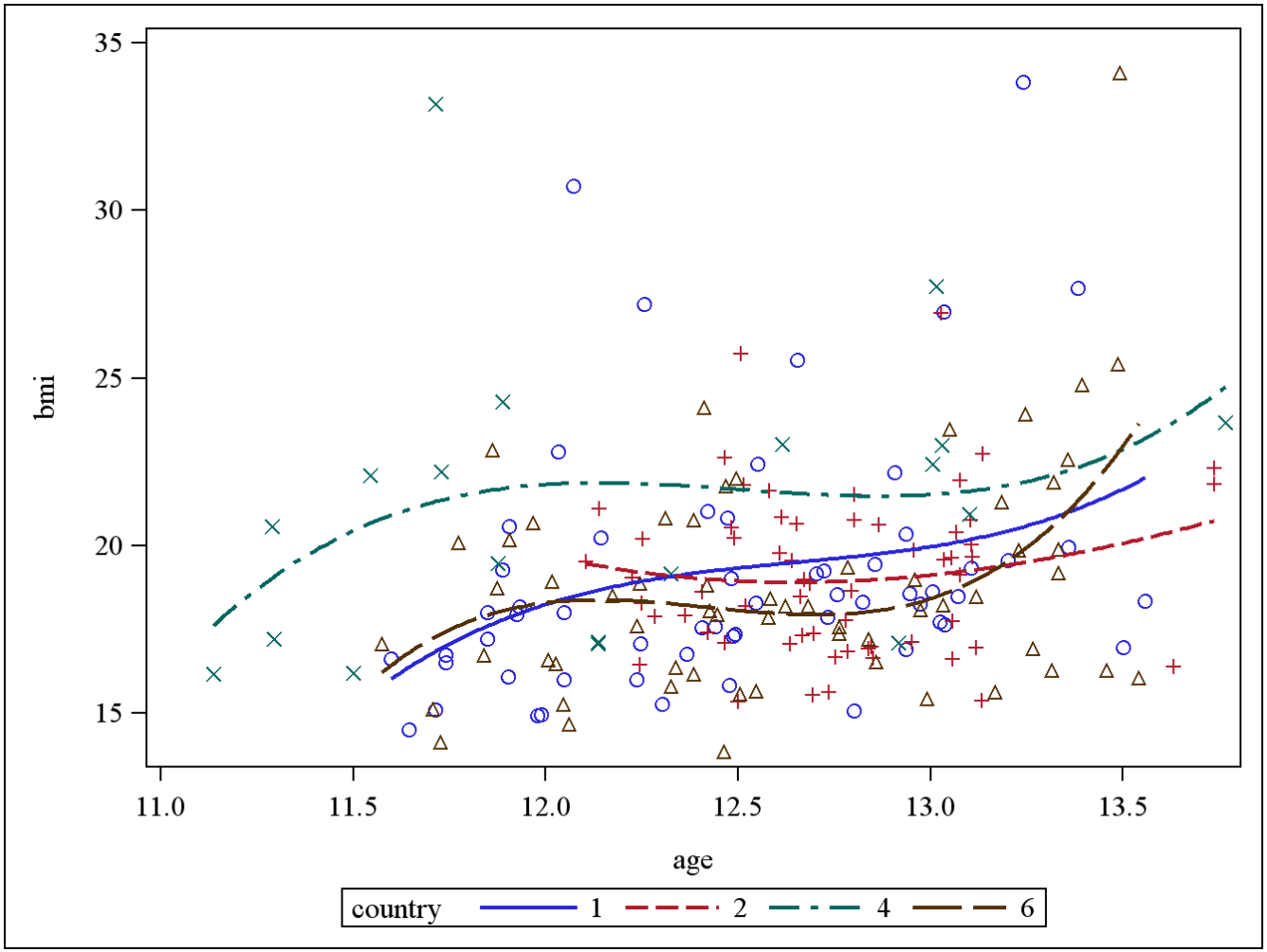
What remains of these slides is DIY.

- To get you started with SAS
- To help you solve the exercise problems.

The data set used in the following slides is here: "http://staff.pubhealth.ku.dk/~lts/basal/data/vitamind.txt";

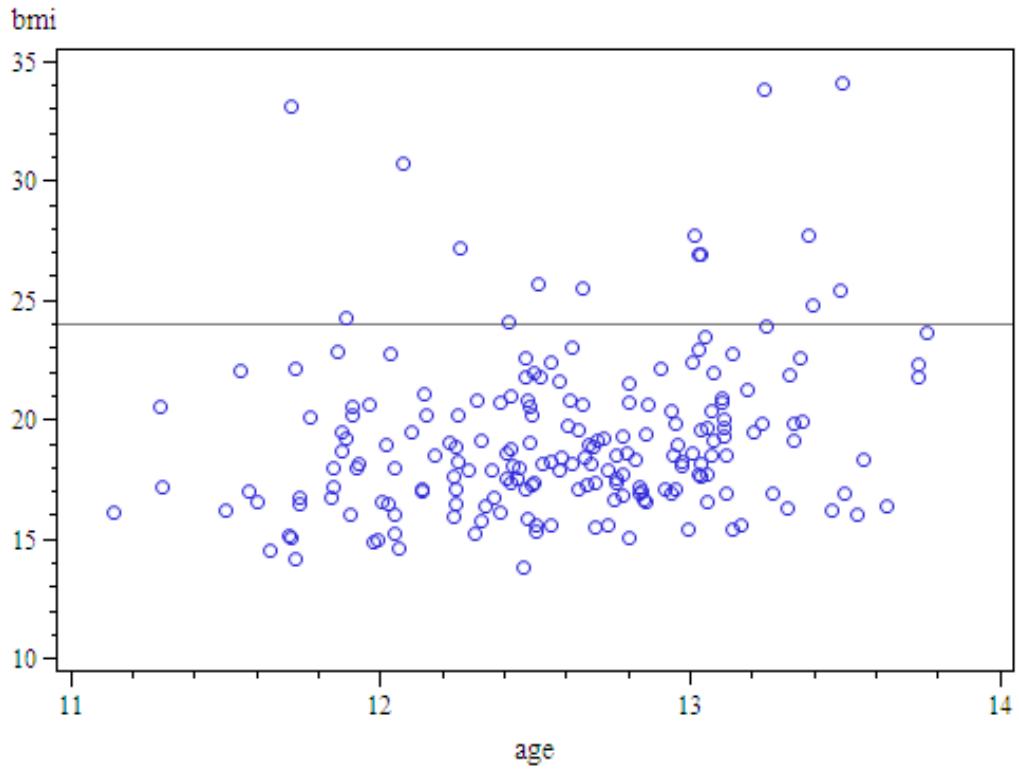
Scatter plot with curves

```
PROC SGPLOT DATA=vitamind;
  REG X=age Y=bmi/GROUP=country DEGREE=3;
RUN;
```



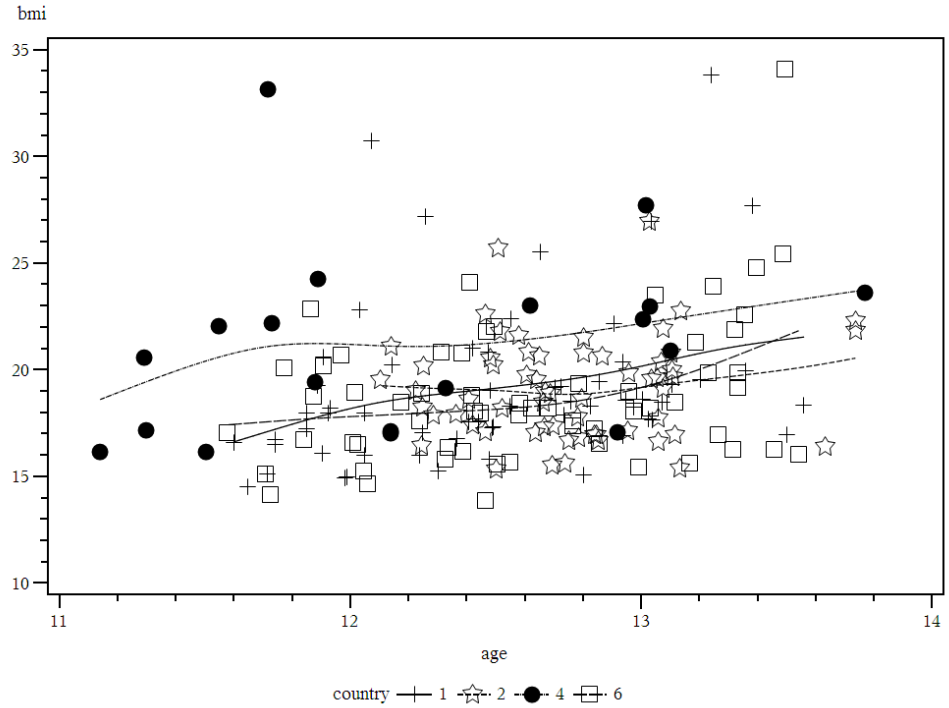
Add a reference line

```
PROC SGPLOT DATA=vitamind;
  SCATTER X=bmi Y=age;
REFLINE=24;
RUN;
```



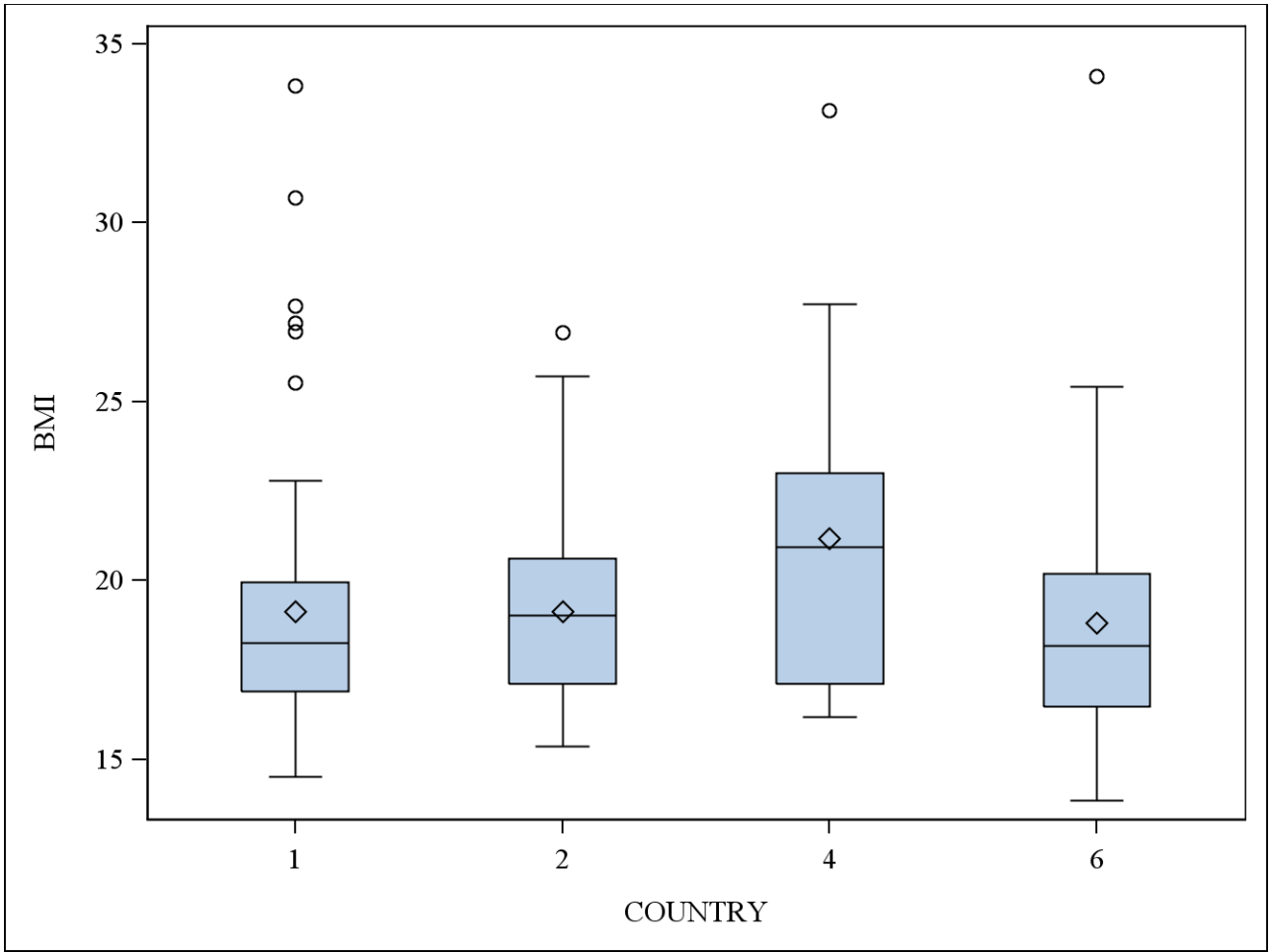
Changing the plot appearance

- Enhanced graphics editor.
- Various options to PROC SGPLOT (check with online ressources)



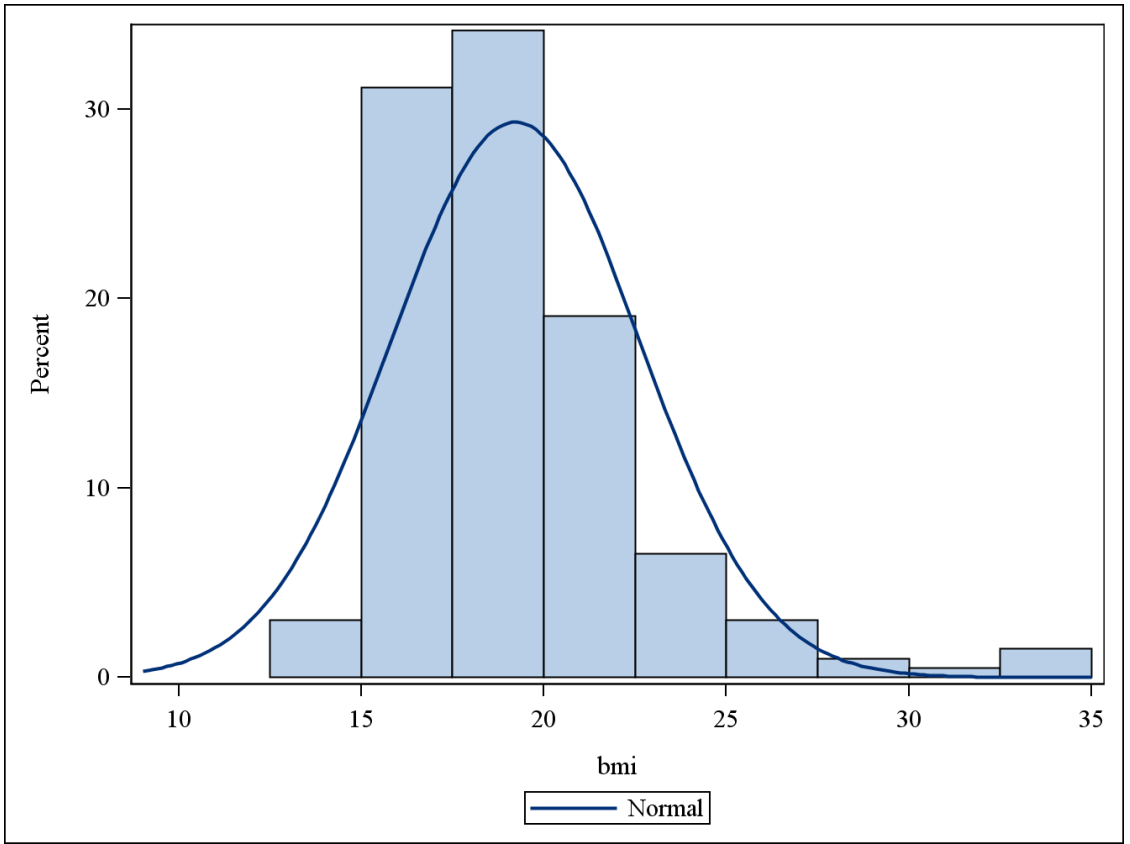
Box plot

```
PROC SGPLOT DATA=vitamind;
  VBOX bmi/CATEGORY=country;
RUN;
```



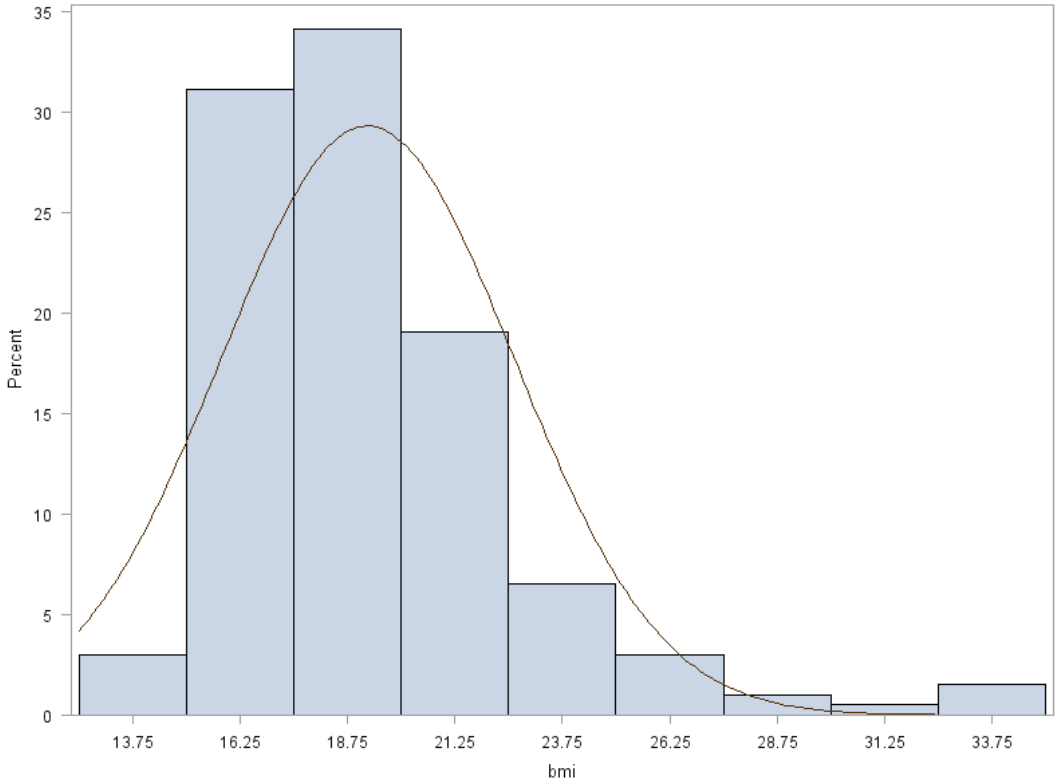
Histogram

```
PROC SGPLOT DATA=vitamind;
  HISTOGRAM bmi;
  DENSITY bmi;
RUN;
```



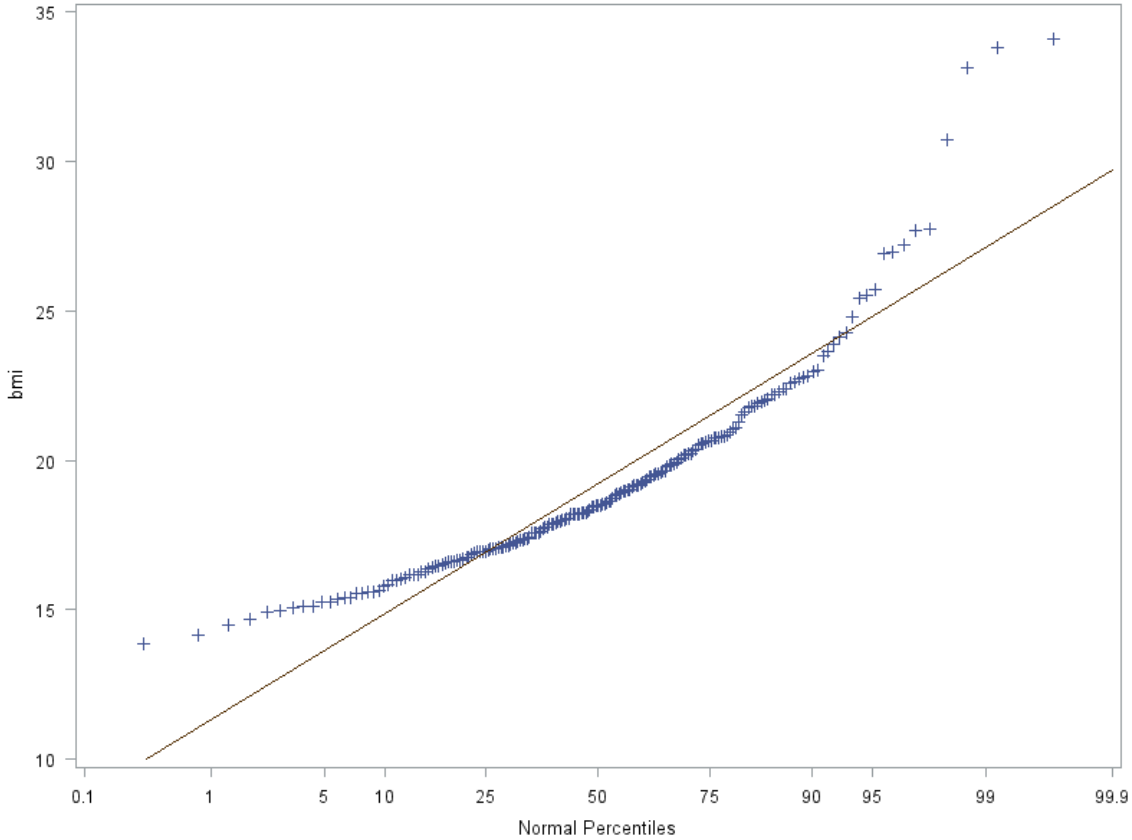
Histogram – alternative

```
PROC UNIVARIATE DATA=VITAMIND;
  VAR bmi;
  HISTOGRAM bmi/NORMAL;
RUN;
```



Probability Plot

```
PROC UNIVARIATE DATA=VITAMIND;
  VAR bmi;
  PROBPLOT bmi/NORMAL(MU=est SIGMA=est);
RUN;
```



Quantiles

```
PROC UNIVARIATE DATA=VITAMIND;
  VAR bmi;
  OUTPUT OUT=p_res pctlpre=P pctlpts=2.5,97.5;
RUN;
```

The 2.5 and the 97.5 percentiles for BMI are stored in a new dataset named ”p_res”.

Descriptive statistics

```
PROC MEANS DATA=vitamind N NMISS MEAN
MEDIAN P25 P75 MAXDEC=2;
   VAR bmi age;
RUN;
```

Variable	N	Miss	Mean	Median	25th Pctl	75th Pctl
bmi	412	0	23.60	23.03	18.56	27.64
age	412	0	43.21	69.47	12.64	71.84

Other than N NMISS MEAN MEDIAN P25 P75 MAXDEC you can compute e.g. Stddev etc.

Frequency tables

```
PROC FREQ DATA=VITAMIND;
  TABLE country;
RUN;
```

country	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	112	27.18	112	27.18
2	114	27.67	226	54.85
4	60	14.56	286	69.42
6	126	30.58	412	100.00

Cross tabulation

```
PROC FREQ DATA=VITAMIND;
  TABLE country*category/NOPERCENT NOCOL;
RUN;
```

Table of country by category			
country	category		
Frequency Row Pct	1	2	Total
1	59 52.68	53 47.32	112
2	60 52.63	54 47.37	114
4	19 31.67	41 68.33	60
6	61 48.41	65 51.59	126
Total	199	213	412

Output formats

- Click Tools
- Choose Options
- Click Results
- Choose whatever format you would like to get your results in, e.g. .RTF

Alternative: Export the default output from the output window.

Missing values

Text strings: represented as empty string

Numerical variable: represented as . or .x

ATT . is given a value of ”minus infinity”.

So ”IF .z<var <8 THEN…” should be used whenever there are missing values of ”var” while ”IF var <8 THEN…” should not.

Shortcuts

To see a list of shortcuts choose:

- Help
- SAS Enterprise Guide Help
- Keyboard Shoutcuts

F8	Run the program
Ctrl i	Prettyfy my program
F1	Help
Ctrl Shift H	Search online for marked word
Ctrl *	/**/ Make a new comment

Autocompletion

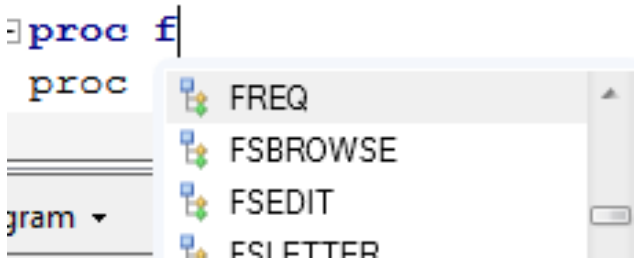
The shortcut “CTRL + Shift + V” turns on autocompletion.

```
data hes_charlson;
 merge hes_icd_hypo(drop=poi
 by icd ;
 if ja1
run; * 2
```

- cond_name
- cond_no
- ICD
- new_wgt
- old_wgt

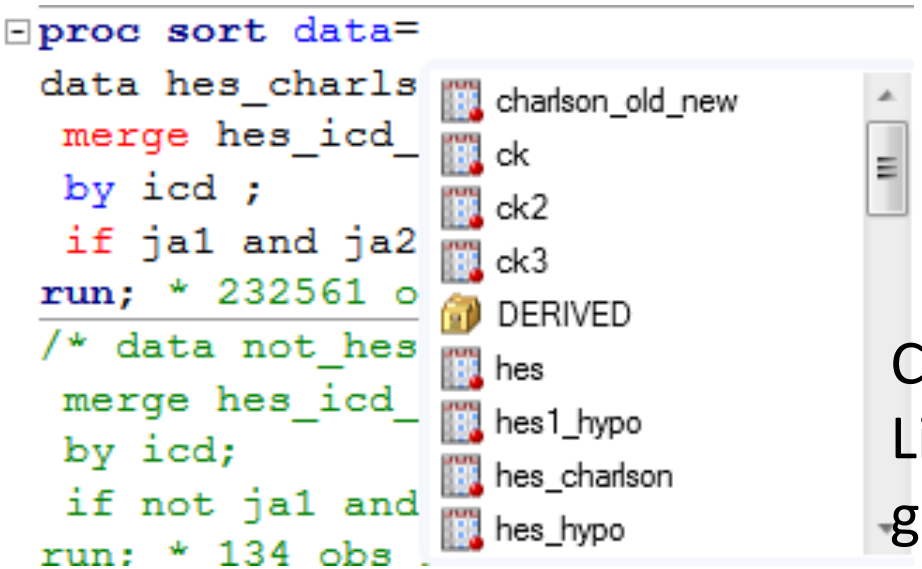
List of variables in the relevant dataset

Suggested code



- After Proc

Lists all procedurer starting with e.g. ”f”, ”fr”, etc



- After data

Lists all active datasets

CTRL + D
Lists all dataset that has been generated in a DATA step.

Syntax dictionary

```
proc sort data=hes_icd_hypo; by icd; run;
```

Keyword: PROC

Context: [GLOBAL STATEMENT] PROC statement

Syntax: PROC procedure-name <options>;

Begins a PROC step. The PROC step consists of a group of SAS statements that call and execute a procedure, usually with a SAS data set as input.

Search: The Product Documentation, Samples & SAS Notes, Papers

Hover the mouse over ”proc”

- Then you get an explanation of the syntax

Procedure Syntax

```
proc sort data=
data
 merg
 by i
 if j
run;
```

Keyword: SORT

Context: [PROCEDURE DEFINITION] PROC SORT

Syntax: PROC SORT <collating-sequence-option> <other option(s)>;
BY <DESCENDING> variable-1 <...<DESCENDING> variable-n>;

The SORT procedure orders SAS data set observations by the values of one or more character or numeric variables. The SORT procedure either replaces the original data set or creates a new data set. PROC SORT produces only an output data set.

Search: The Product Documentation, Samples & SAS Notes, Papers

Hover the mouse over "sort"

- Then you get an explanation of the syntax

Splitting the screen

Option 1: **Split the screen to see different sections of the same program/output etc.**

- Right click at the top of the window frame (or just click the split screen icon)
- Choose ‘Split’
  - ‘Stacked’, ‘Side by Side’ or both

Option 2: **Split screen to see different windows simultaneously**

- Choose View in the main menu
- Click ‘Workspace layout’
- Choose ‘Stacked’ or ‘side by side’.
- Every part of the screen has a drop down menu in the top left corner where you can choose which window to look at (e.g. Log AND Program).

Different sections of the program

Top and bottom can be wieved simultaneously, so you don't have to scroll.

Program

Program* | Log | Output Data | Results

Save | Run | Stop | Selected Server: Local (Connected) | Analyze Program

```
data tumors;
  infile "C:\Users\Public\STATCON\Undervisning\Biostat\E
  input day mouseid volume treatment $ sacrificed;
run;

proc sort data=tumors; by treatment mouseid day; run;

proc freq data=tumors;
  by treatment;
  tables day*sacrificed / nocol nopercent;
run;
```

```
run;

proc freq data=tumors;
  by treatment;
  where day eq 20;
  tables sacrificed / binomial(level='1');
run;
```

Different windows stacked