Use of SAS
December 2010

Introduction

Who are we?

Karl Bang Christensen and Esben Budtz-Jørgensen,

from the Department of Biostatistics, University of Copenhagen.

Format

- 6 lessons per day for three days
- Lectures/demonstrations and exercises
- 80% participation required
- Sign participation lists morning *and* afternoon
- Morning: 9.15-12, Afternoon: 13-15.45
- Coffee break ?

Login etc..

- you should have Username/Passwd
- Recommended: Work 2 and 2 at the machines

Plan

- Theoretical background (courses)?
- not much statistics here
- Focus on the SAS language
- Slide copies
- Reference: The Little SAS Book, (does not cover graphics)

Monday

1. Introduction
2. SAS overview
3. SAS datasets
4. Introduction to the SAS language
5. Transformation and selection
6. Simple univariate procedures
7. SAS procedures for t-tests

Tuesday

1. SAS procedures for:
   (a) simple regression analysis
   (b) graphics
   (c) residual plots
2. More about the SAS language:
   (a) combination of data sets
   (b) dates
   (c) formats

Wednesday

1. multiple regression - Proc GLM
2. analysis of categorical data - Proc Freq
3. logistic regression - Proc Logistic
4. reading data into SAS
5. Evaluation

1. Introduction to SAS programming

Use of SAS
December 2010

Basic concept of SAS programming

- Write a SAS program
  - A program is like a recipe: a series of instructions to be executed in a specified sequence
- Let SAS execute and interpret your program and do some statistical calculations
- SAS responds by giving results in some format

– We need to know the syntax and rules for the SAS language.

Typical use of SAS for statistical analysis

- You have data in some format
- Create SAS data (day 3)
- Get the data into SAS (libname statement)
- Look at the data using SAS
- Perhaps transform or select part of the data, i.e making the data ready for statistical analysis
- Use the appropriate SAS statistical procedure, called a ”proc”, e.g. `proc logistic` for logistic regression
- Get your results out
- Check that SAS did what you asked for (SAS does some ”logging” of your instructions)
- Interpret the SAS output

Basic structure of the SAS system

- SAS programing
  - data step
  - proc step
- Base SAS (50 different proc's)
  - Data manipulation: so-called "data-step", `proc sort`
  - Showing data: `proc contents`, `proc print`, `proc format`
  - Basic statistics: `proc means`, `proc univariate`, `proc freq`
- Special modules
  - SAS/STAT: `proc ttest`, `proc glm`, `proc logistic`, ... (more than 65 proc's)
  - SAS/GRAPH: `proc gplot` (23 proc's)
  - other modules: SAS/ASSIST, QC, ETS, FSP, IML, ...

SAS-data sets

Observations × variables :

sex	age	weight	name
1	8	25	John
2	5	17	Anna
2	13	48	Maria
⋮	⋮	⋮	⋮

- Variable names: sex age weight name (OBS: not æøå)
- Variable types: Numeric or Character

SAS Analyst

- Add-on to SAS ("application")
- Menu/form-oriented
- Writes and runs programs for you
- Similar to SPSS

BUT:

- Does not cover everything
- Tedious to use in the long run
- Analysis will be difficult to reproduce

We do not use this

SAS Display Manager

Framework for program development and data handling

- Editor window: Writing SAS programs
- Log window: ”Logging”, i.e. what happen when and did it work?
- Output window: Results from proc’s
- SAS Explorer: Navigating the SAS system as you have seen
- Results: Structured output, designed for helping the user navigating in large amount of output

SAS-Explorer and SAS libraries

Similar to Windows Explorer, where you can access SAS-data stored in SAS libraries.

- SAS libraries contains:
  - SAS-data sets
  - SAS catalogs (user defined formats, options, setup info, etc.)
- Double-click Libraries in SAS-Explorer. SAS is born with some libraries of which two are important to know about:
  - WORK
  - SASUSER

Getting data into SAS: the libname statement

On your own drive in the library `p:\sas\data\other` you will find the SAS data set `fitness.sas7bdat`.

To get this into SAS, we trun `p:\sas\data\other` into a SAS library. This is done with the a libname statement. Go to the Editor window. Type in the fragment below and click on "the running man" in toolbar or press function key F3.

```
libname new 'p:\sas\data\other';
```

We give the library `p:\sas\data\other` the nickname 'new'. The name 'new' is something you decide. BUT you have to give the precise path of the library.

From now on we can refer to files in library using the name 'new'.

Now use SAS-Explorer to find the fitness data.

Viewtable

The data set is opened in a window called VIEWTABLE

- Note entries under **View**: Forms/Table mode and Column Names/Labels.
- Data can be in "Browse" mode or "Edit" mode. In Browse-mode, you can only navigate the data set, not change it, add new variables, etc.
- Default is Browse mode
- You switch modes with **Edit** $\longrightarrow$ **Mode** $\longrightarrow$ **Edit**
- We recommend NEVER to change data in viewtable. It is much better to write a program doing the changes.
- NB: Do not try to run programs that modify a data set which is open in Viewtable: Now, close Viewtable using a single-click in upper right corner.

Exercise: Reading in some ”nice” data

In this exercise you will be working with a data file in SAS format called `bissau.sas7bdat`. Data comes from rural Guinea-Bissau, West-Africa: 5273 children visited when being less than 7 months of age and followed for approximately six months. Registration of vaccination status, weight, etc at visit and deaths registered during follow-up.

- Using WINDOWS explorer, find the SAS-data set `bissau.sas7bdat` on your personal drive in the directory `p:\sas\data\africa`.
- Link this directory to a SAS library called `afrika` using a `libname` statement.
- Take a look at the data using VIEWTABLE.

More about the Editor window

- Here you write programs and ”submit” them to SAS
- Standard free text editing
- SAS is case-INsentitive (SEX, sex, Sex all refer to the same)
- Load and save editor contents (called `.sas` files, e.g. `henrik.sas`)
- Line numbers, Colours, indentation, and separators
- Type in the fragment below and click on “the running man” in toolbar or press function key F3:

```
proc print data=new.fitness;
run;
```

PROC PRINT

Obs	age	weight	runtime	rstpulse	runpulse	maxpulse	oxygen	group
1	57	73.37	12.63	58	174	176	39.407	2
2	54	79.38	11.17	62	156	165	46.080	2
3	52	76.32	9.63	48	164	166	45.441	2
4	50	70.87	8.92	48	146	155	54.625	2
5	51	67.25	11.08	48	172	172	45.118	2
6	54	91.63	12.88	44	168	172	39.203	2
7	51	73.71	10.47	59	186	188	45.790	2
8	57	59.08	9.93	49	148	155	50.545	2
9	49	76.32	9.40	56	186	188	48.673	2
10	48	61.24	11.50	52	170	176	47.920	2
11	52	82.78	10.50	53	170	172	47.467	2
12	44	73.03	10.13	45	168	168	50.541	1
13	45	87.66	14.03	56	186	192	37.388	1
14	45	66.45	11.12	51	176	176	44.754	1
15	47	79.15	10.60	47	162	164	47.273	1
⋮							⋮	
26	38	89.02	9.22	55	178	180	49.874	0
27	47	77.45	11.63	58	176	176	44.811	0
28	40	75.98	11.95	70	176	180	45.681	0
29	43	81.19	10.85	64	162	170	49.091	0
30	44	81.42	13.08	63	174	176	39.442	0
31	38	81.87	8.63	48	170	186	60.055	0

Now try the procedure CONTENTS which will display the so-called header for the data set:

```
proc contents data=new.fitness;
run;
```

PROC CONTENTS

The CONTENTS Procedure

Data Set Name	NEW.FITNESS	Observations	31
Member Type	DATA	Variables	8
Engine	V9	Indexes	0
Created	12. januar 2006 torsdag 23:28:07	Observation Length	64
Last Modified	12. januar 2006 torsdag 23:28:07	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label	Exercise/fitness study data set		
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Label
1	age	Num	8	Age in years
8	group	Num	8	Experimental group
6	maxpulse	Num	8	Maximum heart rate
7	oxygen	Num	8	Oxygen consumption
4	rstpulse	Num	8	Heart rate while resting
5	runpulse	Num	8	Heart rate while running
3	runtime	Num	8	Min. to run 1.5 miles
2	weight	Num	8	Weight in kg

Log window

In the Editor, add the option `position`, but you should misspell it as `poTition`

```
proc contents data=new.fitness poTition;
run;
```

Well, something happend, inspite of the error. Try to look in the log window: Press F6 or use your mouse. SAS does some guessing - very helpful. The option `position` will add a list where the variables are listed in order of their "position" in the data set. Now try to write `politi` instead. Now we get an error - look in the log.

INFORMATION IN THE LOG WINDOW IS IMPORTANT TO CONSULT EVEN IF YOU BELIEVE EVERYTHING IS OKAY...

Results window

Results window got folders added

Print: The SAS system

Contents: The SAS system

Double-click the nodes inside these to see its contents

More about SAS libraries

- Structure containing datasets (and SAS catalogs)
- Actually just a folder on the computer disk
- See Properties in the Explorer for precise location
- The library WORK is a *temporary* library, i.e., it disappears (with everything inside it) when SAS is shut down. Used for intermediate results.
- SASUSER is permanent. Datasets here are still available the next time you open SAS.
- More than one library can refer to the same folder on the disk.

More about SAS-data sets

- Specified like `new.fitness`
- First part (before the dot) is the SAS library
- Second part is the name of the data set
- If first part is omitted, WORK is assumed
- Seen from Windows, data are in files with extension `sas7bdat`
- data sets have two logical parts, the header (a descriptive part) and the actual data
  - PROC CONTENTS respectively PROC PRINT

More about the DATA step

Submit the following data-step

```
data fitness;
  set new.fitness;
run;
```

The first line: creates a new data set. It is called fitness and is placed in the WORK library. Sofar it is empty.

The second line: copies the fitness data from the new-library into data set in WORK.

Go to the WORK library and find this version of the fitness data.

Saving a SAS program, Output window, Log window

To save what you have been producing in the Editor, you act as in e.g. Word:

File $\longrightarrow$ Save as...

You save the Output and Log window in the same manner.

File extensions are:

`*.sas` Program files

`*.log` Log files

`*.lst` Output files

Keys

Shortcut keys for navigating the SAS system, press F9 to see. Important ones are:

- F5: Editor
- F6: Log
- F7: Output
- F3: ”Submit”, i.e. run the SAS program in the Editor

Close down the Keys window using either your mouse or the standard Windows ”Ctrl+F4”. Press F5 to go to the Editor.

NB: If you by accident closed the Editor you can get another one using the menu **View** $\longrightarrow$ **Enhanced Editor** (the View menu also have a ”program editor” which is an older less featured version of the editor).

Advantages/disadvantages in SAS

+ Handles large amounts of data
+ Flexible data handling
+ Cover many (many) statistical models/methods
+ Many platforms
+ De facto standard, also large user-groups
– A bit old-fashioned
– Difficult to learn
– Unflexible in advanced programming
– Hard to make good-looking graphics (?)

The help system

- Via the small book icon on the toolbar or F1 – but does not work here
- Usually, you'll need to use the index
- You generally need to know roughly what you are looking for. It is not a textbook.
- SAS OnlineDoc which contains material that used to be found in large manuals (you can still buy those):

`http://support.sas.com/onlinedoc/913/docMainpage.jsp`

- "The Little SAS Book"

`http://support.sas.com/publishing/bbu/companion_site/56649.html`

How to get organized

- Interactive runs are convenient, but also dangerous
- Remember to save and keep program code.
- Collect the fragments into coherent `.sas` files that can be run from scratch.
- Test your programs in a freshly started SAS session.

Exercise: Bissau data

- How many observations and variables are in the data set? Use e.g. `proc contents`.
- Make a copy of data `afrika.bissau` into `work.bissau`. Use the 'set'-statement.
- Make a SAS program that makes a `proc print` of the Bissau data. Save the program somewhere on your personal drive.
- Restart SAS (closing and starting SAS).
- Open your SAS program and submit it. Did it work? If not, why not?

SAS programming

- A program is a "recipe": A series of instructions to be executed in a specified sequence
- Notice: SAS is not a spreadsheet. Output is output, and does not change automatically if data are changed
- Some rules and conventions are necessary for SAS to be able to interpret its instructions
- Separators (semicolon, /)
- Parentheses and quote symbols must be matched

A simple SAS program

```
data new;                                |
  set original;                          |
  age=1997-birthyr;                      |  Data Step
  bmi=weight/(height*height);            |
run;                                     |


proc print data=new;                     |
  var id age bmi;                        |
run;                                     |
                                         |  Proc Steps
proc means data=new;                     |
  var age bmi;                           |
run;                                     |
```

DATA steps and PROC steps

- Roughly speaking, SAS programs consist of two kinds of *steps* (= blocks of instructions):
- DATA steps define datasets. E.g. by reading raw data, computing transformed variables, selecting cases, etc.
- PROC steps contain standard procedures that operate *on* datasets. You can't, e.g., transform variables in a PROC step.
- Normal arrangement of a SAS program is to put DATA steps at the beginning, but they can occur intermixed
- There are a a few SAS *statements* in addition to the DATA and PROC steps. They typically set up definitions for later use: LIBNAME, OPTIONS, AXIS, and SYMBOL statements are the most common ones

Basic things about the SAS language

- Almost everything starts with a keyword and ends with semicolon (exception being that there is no keyword before computations in a DATA step)
- **Statements** are pieces of code separated by semicolon

```
OPTIONS ls=80;
PROC GPLOT data=sasuser.fitness;
   PLOT maxpulse * age;
RUN;
PROC GLM data=sasuser.fitness;
   MODEL maxpulse = age / solution;
RUN;QUIT;
```

- Keywords: OPTIONS PROC PLOT MODEL RUN QUIT
- Some statements belong together in blocks (**steps**).

Things to notice

```
OPTIONS ls=80;
PROC GPLOT data=sasuser.fitness;
   PLOT maxpulse * age;
RUN;
PROC GLM data=sasuser.fitness;
   MODEL maxpulse = age / solution;
RUN;QUIT;
```

- The slash symbol (`/`) is often used to introduce options for a statement
- Separators like semicolons and slashes are necessary to avoid ambiguity: `solution` is not a variable name, `run` is not an option.
- SAS detects the end of a step when there is a new DATA or PROC statement (RUN is not always needed).

Formatting of code

- SAS generally doesn't care about whitespace and line breaks

```
data
        work.cohort;
        set course.males98;
run;
```

- is the same as

```
data work.cohort; set course.males98; run;
```

- Good practice is to have at most one statement per line.

Indentation

- Enhances readability considerably. (You *will* have to read your own old code!)
- DATA and PROC steps are entered starting at the left edge. Likewise `OPTIONS` statements and `RUN` and `QUIT`.
- Any subordinate statements are indented by 2-4 blanks
- In statements which do not fit on one line, subsequent lines are also indented.
- This creates visual groups, so that you can easily see where one thing ends and the next begins.

Example of good indentation

```
data new;
  set original;
  age=1997-birthyr;
  bmi=weight/(height*height);
run;


proc print data=new;
  var id age bmi;
run;

proc means data=new;
  var age bmi;
run;
```

Ingredients of a DATA step

```
data new;
  set original;
  age=1997-birthyr;
  bmi=weight/(height*height);
run;
```

- Specification line (name of new data set)
- Data source (here: name of old SAS data set)
- Computation
- Assignment

Variables

- Columns of a dataset
- Can be numerical (usual case)
- – or character (text strings)
- Values of a character variable are given in quotes: `'male'` or `"male"`
- A dot (`.`) denotes a missing value for a numerical variable and is the lowest number in SAS.
- Calculations involving a missing will result in a missing (most of the times)
- A "" or " denotes a missing value for a character variable.

Names of variables

- SAS is case-insensitive (`SEX`, `sex`, `Sex` all refer to the same variable)
- Names can be up to 32 characters long (older SAS: max 8)
- Names can consist of (english) letters, numbers and underscore (_)
  – but can *not* start with a number

Comments

Two ways of making comments in SAS programs:

```
/* Comments */

 * Comments ;
```

Example:

```
/* Here I make a new data
with new variables age and bmi*/
data new;
  set original;
  age=1997-birthyr;
  bmi=weight/(height*height);
run;

*Here I print age and bmi;
proc print data=new;
```

```
  var id age bmi;
run;

*Here I calculate means;
proc means data=new;
  var age bmi;
run;
```

2. Simple procedures

Use of SAS
December 2010

Contents

- transformation and selection
- descriptive procedures
- sorting data

Example O'Neill et.al. (1983):

Lung function for 25 patients with cystic fibrosis.

Table 12.11 Data for 25 patients with cystic fibrosis (O'Neill *et al.*, 1983)

Sub	Age	Sex	Height	Weight	BMP	FEV_1	RV	FRC	TLC	PEmax
1	7	0	109	13.1	68	32	258	183	137	95
2	7	1	112	12.9	65	19	449	245	134	85
3	8	0	124	14.1	64	22	441	268	147	100
4	8	1	125	16.2	67	41	234	146	124	85
5	8	0	127	21.5	93	52	202	131	104	95
6	9	0	130	17.5	68	44	308	155	118	80
7	11	1	139	30.7	89	28	305	179	119	65
8	12	1	150	28.4	69	18	369	198	103	110
9	12	0	146	25.1	67	24	312	194	128	70
10	13	1	155	31.5	68	23	413	225	136	95
11	13	0	156	39.9	89	39	206	142	95	110
12	14	1	153	42.1	90	26	253	191	121	90
13	14	0	160	45.6	93	45	174	139	108	100
14	15	1	158	51.2	93	45	158	124	90	80
15	16	1	160	35.9	66	31	302	133	101	134
16	17	1	153	34.8	70	29	204	118	120	134
17	17	0	174	44.7	70	49	187	104	103	165
18	17	1	176	60.1	92	29	188	129	130	120
19	17	0	171	42.6	69	38	172	130	103	130
20	19	1	156	37.2	72	21	216	119	81	85
21	19	0	174	54.6	86	37	184	118	101	85
22	20	0	178	64.0	86	34	225	148	135	160
23	23	0	180	73.8	97	57	171	108	98	165
24	23	0	175	51.1	71	33	224	131	113	95
25	23	0	179	71.5	95	52	225	127	101	195

Data has been read into the sas-dataset:

`p:\sas\data\other\pemax.sas7bdat`

Definition of new variables

We want to study body mass index, `bmi`.

```
libname mysas 'p:\sas\data\other';

data temp;      /*temporary data set*/
 set mysas.pemax;
 bmi=weight/(height/100)**2;
run;

proc print data=temp;
run;
```

Obs	age	sex	height	weight	fev1	pemax	bmi
1	7	1	109	13.1	32	95	11.0260
2	7	2	112	12.9	19	85	10.2838
3	8	1	124	14.1	22	100	9.1701
.	.	.	.	.	.	.	.

Transformations

- **Arithmetics**
  - The usual operators: + - * /
  - Raising to a power: **, e.g.. `x**2`
  - Square root: `sqrt(x)`
  - Logarithms: `log(x), log10(x), log2(x)`
- Relations:

  ```
  =  <  >  <= >= <> (unequal)
  eq lt gt le ge ne (alternative notation)
  ```

- Logical operators:

  ```
  and   or   not
  ```

Other types of variable definitions

```
data mysas.pemax;
  set mysas.pemax; /*original file changed*/

if sex=1 then csex='m';
if sex=2 then csex='f';

if 0<bmi<12 then fat=1;
if 12<=bmi<18 then fat=2;
if 18<bmi then fat=3;
run;

proc print data=mysas.pemax;
  var csex age bmi fat;
run;
```

Obs	csex	age	bmi	fat
1	f	7	11.0260	1
2	m	7	10.2838	1
.	.	.	.	.
.	.	.	.	.
14	m	15	20.5095	3

If I do not want to change 'mysas.pemax', I can make a new data set:

```
data temp;
  set mysas.pemax;

if sex=1 then csex='m';
if sex=2 then csex='f';

if 0<bmi<12 then fat=1;
if 12<=bmi<18 then fat=2;
if 18<bmi then fat=3;
run;

proc print data=temp;
  var csex age bmi fat;
run;
```

Obs	csex	age	bmi	fat
1	f	7	11.0260	1
2	m	7	10.2838	1
.	.	.	.	.
.	.	.	.	.
14	m	15	20.5095	3

Measures of location, centre

- Average

$$\bar{x} = \frac{1}{n}(x_1 + \cdots + x_n)$$

  - can be interpreted as the centre of gravity

  - is heavily influenced by outlying observations

- Median
  - the observation in 'the middle'
  - is not influenced by outlying observations (**robust**)

Measures of variation

- Variance, standard deviation

$$s^2 = \frac{1}{n-1}\Sigma(x_i - \bar{x})^2$$

$$\textbf{standard deviation} = \sqrt{\textbf{variance}}$$

- Quantiles
  - median: 50% quantile
  - quartiles: 25%, 50% and 75% quantiles

Calculation of summary statistics in SAS

```
proc means data=temp;
run;
```

```
The MEANS Procedure

Variable    N           Mean        Std Dev        Minimum        Maximum
-------------------------------------------------------------------------
age        25     14.4800000      5.0589854      7.0000000     23.0000000
sex        25      1.4400000      0.5066228      1.0000000      2.0000000
fev1       25     34.7200000     11.1971723     18.0000000     57.0000000
pemax      25    109.1200000     33.4369058     65.0000000    195.0000000
bmi        25     15.3422331      3.8633242      9.1701353     22.7777778
-------------------------------------------------------------------------
```

These are **default**,

others may be chosen as **options**

Options in PROC MEANS

```
/*Some of the keywords available with PROC MEANS:
      N - number of observations
      MEAN - mean value
      MIN - minimum value
      MAX - maximum value
      SUM - total of values
      NMISS - number of missing values
      MAXDEC=n - set maximum number of decimal places */

statistic-keyword(s) specifies which statistics to compute
and the order to display them in the output.
The available keywords in the PROC statement are

 Descriptive statistic keywords
 CLM RANGE CSS SKEWNESS|SKEW CV STDDEV|STD KURTOSIS|KURT STDERR
 LCLM SUM MAX SUMWGT MEAN UCLM MIN USS N VAR NMISS

Quantile statistic keywords
    MEDIAN|P50 Q3|P75 P1 P90 P5 P95 P10 P99 Q1|P25 QRANGE
```

If we want to see the medians:

```
proc means data=temp median;
  var age bmi fev1;
run;
```

```
The MEANS Procedure

Variable          Median
------------------------
age           14.0000000
bmi           14.8660771
fev1          33.0000000
------------------------
```

Oops: Now, we got **only** the median!

```
proc means data=temp N mean median;
  var age bmi fev1;
run;
```

```
The MEANS Procedure

Variable     N            Mean          Median
---------------------------------------------
age         25      14.4800000      14.0000000
bmi         25      15.3422331      14.8660771
fev1        25      34.7200000      33.0000000
---------------------------------------------
```

Alternative procedure: UNIVARIATE

```
proc univariate data=mysas.pemax;
  var bmi;
run;
```

a lot of output...

```
The UNIVARIATE Procedure
Variable:  bmi

                          Moments

N                          25    Sum Weights               25
Mean                15.3422331    Sum Observations    383.555827
Std Deviation       3.86332415    Variance            14.9252735
Skewness            0.27214922    Kurtosis            -0.7599282
Uncorrected SS      6242.80947    Corrected SS        358.206564
Coeff Variation     25.1809768    Std Error Mean      0.77266483

             Basic Statistical Measures

    Location                    Variability

Mean     15.34223     Std Deviation            3.86332
Median   14.86608     Variance                14.92527
Mode        .         Range                   13.60764
                      Interquartile Range      5.36231
```

Tests for Location: Mu0=0

Test	-Statistic-		-----p Value------	
Student's t	t	19.85626	Pr > |t|	<.0001
Sign	M	12.5	Pr >= |M|	<.0001
Signed Rank	S	162.5	Pr >= |S|	<.0001

Quantiles (Definition 5)

Quantile	Estimate
100% Max	22.77778
99%	22.77778
95%	22.31516
90%	20.50953
75% Q3	17.98454
50% Median	14.86608
25% Q1	12.62222
10%	10.35503
5%	10.28380
1%	9.17014
0% Min	9.17014

Extreme Observations

------Lowest------		-----Highest-----	
Value	Obs	Value	Obs
9.17014	3	19.4021	18
10.28380	2	20.1995	22
10.35503	6	20.5095	14
10.36800	4	22.3152	25
11.02601	1	22.7778	23

Categorical variables: PROC FREQ

Can be used to illustrate distributions of categorical variables, e.g.

```
proc freq data=temp;
  tables csex sex fat;
run;
```

The FREQ Procedure

csex	Frequency	Percent	Cumulative Frequency	Cumulative Percent
f	11	44.00	11	44.00
m	14	56.00	25	100.00

sex	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	14	56.00	14	56.00
2	11	44.00	25	100.00

fat	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	6	24.00	6	24.00
2	13	52.00	19	76.00
3	6	24.00	25	100.00

Filtering data - selecting subsets

- In `DATA`-step
  - regarding observations: `WHERE, IF, DELETE`
  - regarding variables: `DROP, KEEP`
- In procedures
  - Regarding observations: `WHERE`
  - Regarding variables: `VAR`-statement
    (depending on procedure)

How to use WHERE

If we only want to look at the girls:

```
data temp1;                  /* temporary data set */
 set mysas.pemax;
 where csex='f';
run;

proc print data=temp1;
  var csex age bmi;
run;

Obs     csex     age      bmi

  1      f         7    10.2838
  2      f         8    10.3680
  3      f        11    15.8894
  4      f        12    12.6222
  ..............
```

Alternative ways of writing,

using **IF** or **DELETE**

Look out, if data contains missing values.

```
data temp2;
 set mysas.pemax;
 if csex='f';
run;

----------------------------------------

data temp3;
 set mysas.pemax;
 if csex ne 'm';
run;

----------------------------------------

data temp4;
 set mysas.pemax;
 if sex='m' then delete;
run;
```

temp2 includes only girls,

temp3, temp4 includes grils *and* children with information on 'csex'.

Filterings may be combined:

```
data temp5;
 set mysas.pemax;
 where csex='f' and age>12;
run;

proc print data=temp;
  var csex age bmi;
run;
```

Obs	csex	age	bmi
1	f	13	13.1113
2	f	14	17.9845
3	f	15	20.5095
4	f	16	14.0234
5	f	17	14.8661
6	f	17	19.4021
7	f	19	15.2860

Note: Observations with age missing are included with:

```
data pemax6;
set mysas.pemax;
where csex='female' and age<12;
run;
```

How to use DROP and KEEP

Now that we have `bmi`,
we may not need `height` and `weight`:

The most efficient is to simply delete them:

```
data temp7;
 set mysas.pemax;
 drop height weight;
run;
```

You can also specify which variables you want to keep:

```
data temp8;
 set mysas.pemax;
 keep bmi csex;
run;
```

Using WHERE in procedures

If we for a specific procedure only want to look at the girls (but continue to work with all data):

```
proc print data=mysas.pemax;
  where csex='f';
  var csex age bmi;
run;
```

```
Obs     csex     age      bmi

  1    female      7    10.2838
  2    female      8    10.3680
  3    female     11    15.8894
  4    female     12    12.6222
  ..............
```

Sorting of data

- Often used because other procedures require sorted data
- Example

```
proc sort data=account;
   by town descending debt;
run;
```

- data are sorted by `town` (small values first).
- for each value of `twon` data are sorted by `debt` (high values first - because of descending statement).

BY statement

- Used in many procedures (MEANS, REG, GLM, ...)
- Runs analyses within groups
- Requires that data are sorted
- Remember to get rid of missings, or they will form a separate group
- Example

```
proc sort data=new.fitness;
  by group;
run;
proc means data=new.fitness;
  var maxpulse;
  by group;
run;
```

Output

```
------------------------------------ Experimental group=0 ------------------------------------

                                   The MEANS Procedure

                        Analysis Variable : maxpulse Maximum heart rate

          N            Mean         Std Dev         Minimum         Maximum
         10     177.5000000       6.1689185     168.0000000     186.0000000


------------------------------------ Experimental group=1 ------------------------------------

                        Analysis Variable : maxpulse Maximum heart rate

          N            Mean         Std Dev         Minimum         Maximum
         10     172.7000000       9.1899704     164.0000000     192.0000000


------------------------------------ Experimental group=2 ------------------------------------

                        Analysis Variable : maxpulse Maximum heart rate

          N            Mean         Std Dev         Minimum         Maximum
         11     171.3636364      10.9660634     155.0000000     188.0000000
```

The Juul data set

Anders Juul et al., Dep. GR, Rigshosp.

Serum IGF-I (Insulin-like Growth Factor) reference data set

Age	N	Source
0–5	44	Circumcision, hernia operation
5–20	833	4 schools in the Copenhagen area
20+	153	Hospital staff

AGE	age
MENARCHE	1. mentrual period occurred (1/2, 2 for yes)
SEXNR	1 for boys, 2 for girls
SIGF1	Serum IGF-I
TANNER	Puberty stage a.m. Tanner (1–5)
TESTVOL	Testicular volume

Exercise: Simple procedures

In your p-drive, you can find the file `'\juul2.sas7bdat'` containing the Juul data with variables

- Age (years)
- Height (cm)
- Menarche (No/Yes: 1/2)
- Sexnr (M/F: 1/2)
- Serum IGF1, growth hormone (μg/ml)
- Tanner stage (1–5)
- Testis volume (ml)
- Weight (kg)

1. Read the data it into SAS using a libname statement.
2. Calculate means and standard deviations
3. Use PROC FREQ to determine the distribution of the categorical variables
4. Make a new variable giving the BMI for each person
5. Determine the BMI distribution in each of the Tanner stages (e.g. using a BY-statement).

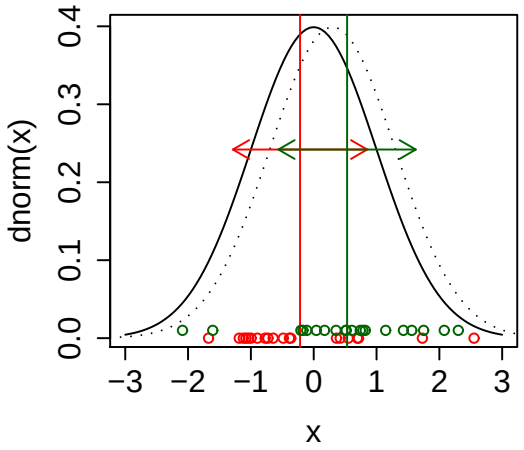
Use of SAS
December, 2010

3. t-test and One-way ANOVA

Comparing two samples

Two groups,

$x_{11}, \ldots, x_{1n_1}$ $\quad$ $x_{21}, \ldots, x_{2n_2}$



$N(\mu_1, \sigma_1^2)$

$N(\mu_2, \sigma_2^2)$

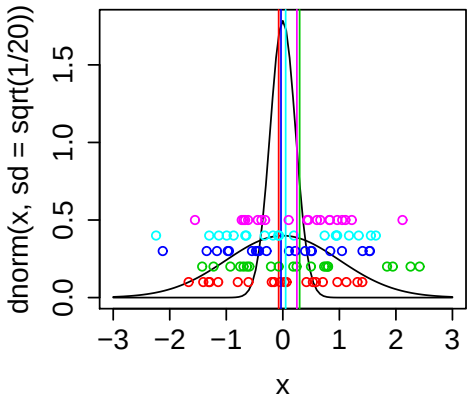
$(\bar{x}_1, s_1^2)$

$(\bar{x}_2, s_2^2)$

Significant difference between $\bar{x}_1$ and $\bar{x}_2$?

Null hypothesis $H_0 : \mu_1 = \mu_2$

Two-sample t-test



$$\text{SEM} = s/\sqrt{n}$$

Standard error of mean

$$\text{SEDM} = \sqrt{\text{SEM}_1^2 + \text{SEM}_2^2}$$

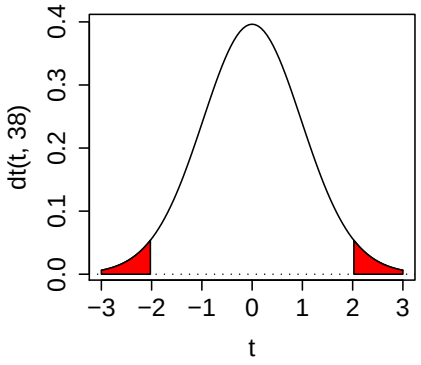
Standard error of difference of means

$$\boxed{t = \frac{\bar{x}_2 - \bar{x}_1}{\text{SEDM}}}$$

test-statistic t measures disagreement between data and H_0

The p-value

$$\boxed{t = \frac{\bar{x}_2 - \bar{x}_1}{\text{SEDM}}}$$



t: measures disagreement between data and H_0

If H_0 is true: distribution of t is symmetric around 0

p: the prob. of having observed a more extreme t-value

if $p < 5\%$: H_0 is rejected

Two tests: same or different variances?

Assume $\sigma_1^2 = \sigma_2^2$ before testing $\mu_1 = \mu_2$?

Same variance:

- Natural under null hypothesis (same distributions)
- Nice theory.

Separate variances:

- Looks specifically for difference in means
- Approximative theory.

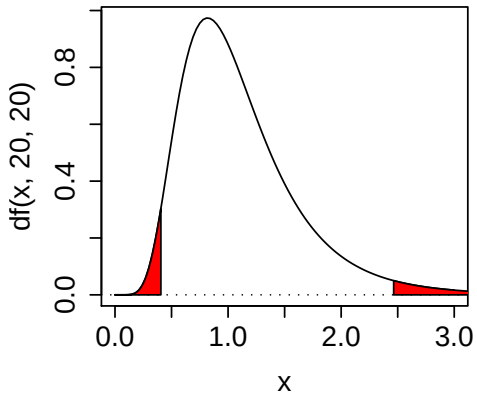
Test for same variance

Test statistic

$$F = s_1^2 / s_2^2$$

F-distribution with (f_1, f_2) degrees of freedom

$f_1 = n_1 - 1 \qquad f_2 = n_2 - 1$



Note: 2-sided test

t-test in SAS

Example from Anders Juul

IDEA: Compare men and women with respect to $\sqrt{\mathsf{SIGF1}}$ for 20–30 year olds.

1. Open dataset: SET statement;
2. Compute SSIGF1 = SQRT(SIGF1)
3. Start PROC TTEST
4. Use WHERE to select subgroup
5. Specify dependent variable
6. — and classification
7. Do not forget RUN

Code for the t-test

```
data juul;
   set sasuser.juul;
   ssigf1=sqrt(sigf1);
run;
proc ttest data=juul;
   where age > 20 and age < 30;
   var ssigf1;
   class sexnr;
run;
```

Output

The TTEST Procedure

Statistics

Variable	sexnr	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err
ssigf1	1	23	15.789	16.517	17.245	1.3021	1.6836	2.3829	0.3511
ssigf1	2	18	15.798	16.824	17.85	1.5481	2.063	3.0928	0.4863
ssigf1	Diff (1-2)		-1.49	-0.307	0.8763	1.5225	1.8586	2.3864	0.5849

T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
ssigf1	Pooled	Equal	39	-0.52	0.6030
ssigf1	Satterthwaite	Unequal	32.5	-0.51	0.6125

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
ssigf1	Folded F	17	22	1.50	0.3666

Exercise: T-test

Consider again the Juul data with variables

- Age (years)
- Height (cm)
- Menarche (No/Yes: 1/2)
- Sex (M/F: 1/2)
- Serum IGF1, growth hormone (μg/ml)
- Tanner stage (1–5)
- Testis volume (ml)
- Weight (kg)

Here the main aim is to compare the IGF1-level in boys and girls *above the age of 5 years*.

1. For each Tanner-stage, test if the IGF1-level is the same in boys and girls. The distribution of IGF1 seems to be skew, but $\sqrt{SIGF1}$ can be assumed to follow a normal distribution.

One-way ANOVA

Comparing more than two groups

$\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_k \qquad s_1, s_2, \ldots, s_k$

Joint test for any differences between the groups.

Why not just pairwise t-tests?

MASS SIGNIFICANCE
LOSS OF OVERVIEW

The fewer tests, the better.

Notation and Models

x_{ij} observation no. j in group no. i

(i.e., x_{35} the 5th observation in group 3)

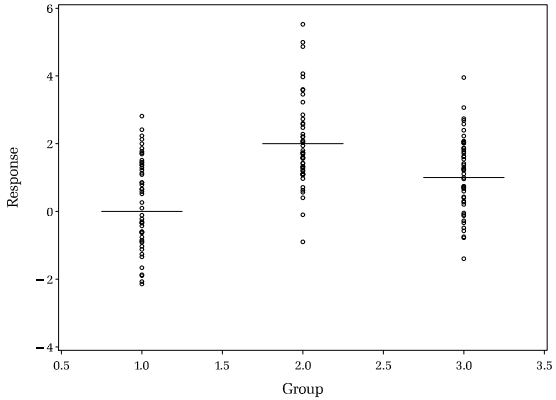
The model

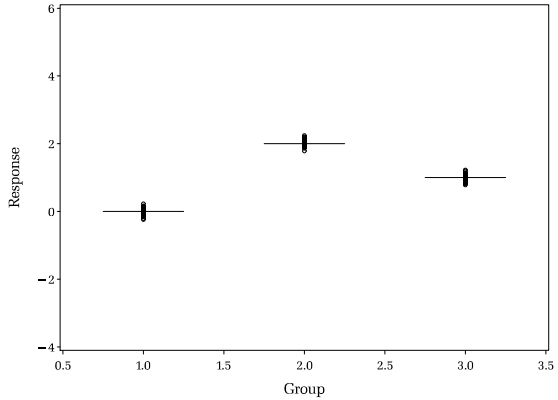
$$X_{ij} = \mu_i + \epsilon_{ij} \qquad \epsilon_{ij} \sim N(0, \sigma^2)$$

The hypothesis of no differences between groups

$$H_0 : \mu_1 = \mu_2 = ... = \mu_k$$

Variation within and between groups





Main idea:

If the variation between group means is large compared to the variation within groups, it is a sign that the hypothesis is wrong.

Sums of squares

Variation (**W**ithin) groups: $\text{SSD}_W = \sum_i \sum_j (x_{ij} - \bar{x}_i)^2$

$\bar{x}_i$ mean for group i

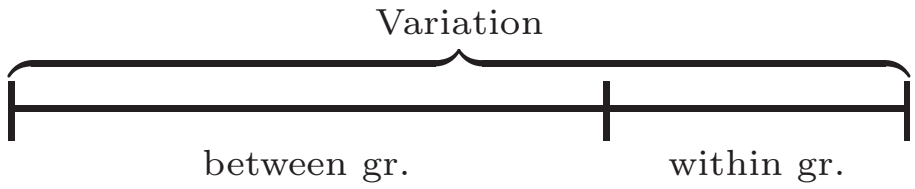
Variation (**B**etween) groups: $\text{SSD}_B = \sum_i \sum_j (\bar{x}_i - \bar{x}_\cdot)^2$

$\bar{x}_\cdot$ total (grand) mean

Can be mathematically proven that

$$\text{SSD}_B + \text{SSD}_W = \text{SSD}_{\text{total}} = \sum_i \sum_j (x_{ij} - \bar{x}_\cdot)^2$$

The model (grouping) *explains* part of the variation

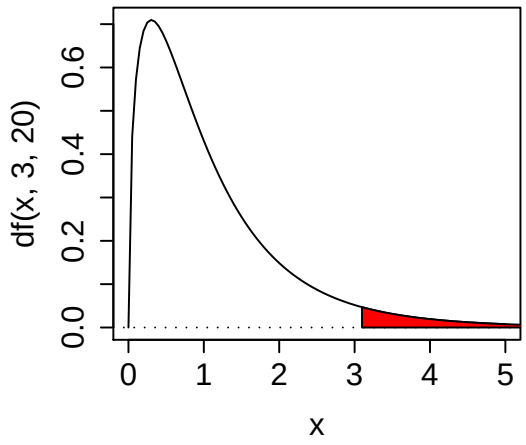


F-test for identical group means

We reject the hypothesis if the variation between groups is large compared to the variation within groups. Consider

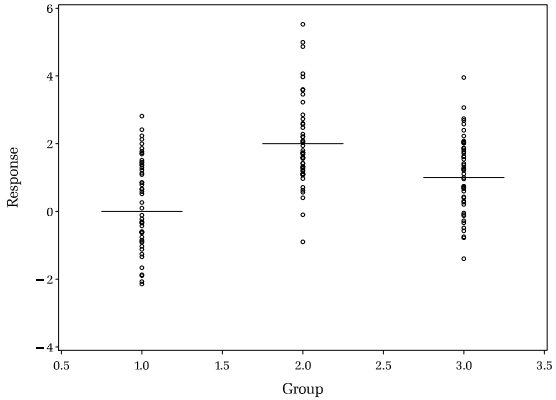
$$F = [\mathrm{SSD}_B/(k-1)]/[\mathrm{SSD}_W/(N-k)]$$

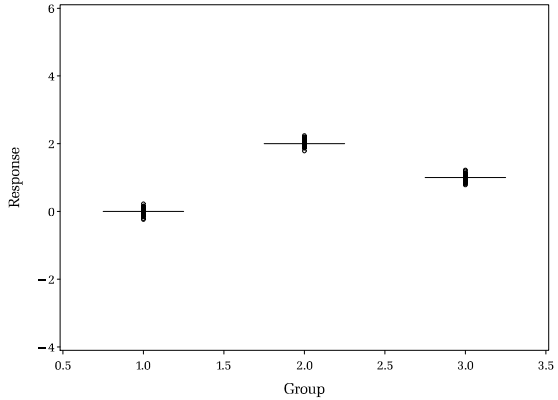
If group differences are coincidental then F follows an F-distribution:



If F too large: Reject the hypothesis that the groups are identical.

Testing for identical group means





high variation within grp.

F is small

H_0 is *not* rejected

small variation within grp.

F is large

H_0 is rejected

One-way ANOVA in SAS

IDEA: Compare boys in different Tanner stage with respect to their $\sqrt{\mathsf{SIGF1}}$

1. This time generate a new data set, `juulboys`
2. Select: $\mathsf{SEXNR} = 1$, $\mathsf{AGE} < 20$
3. Use GLM
4. MODEL statement: What is described by what?
5. Remember to say that `tanner` is a grouping (CLASS)

Code for ANOVA

```
data juulboys;
   set sasuser.juul;
   ssigf1 = sqrt(sigf1);
   if sexnr = 1 and 0 < age < 20;
run;
proc glm data=juulboys;
   class tanner;
   model ssigf1 = tanner / solution;
run;
```

(PROC ANOVA can also be used)

Output

The GLM Procedure

Class Level Information

Class	Levels	Values
tanner	5	1 2 3 4 5

Number of observations 546

NOTE: Due to missing values, only 400 observations can be used in this analysis.

The GLM Procedure

Dependent Variable: ssigf1

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	6054.50950	1513.62738	147.14	<.0001
Error	395	4063.35801	10.28698		
Corrected Total	399	10117.86751			

R-Square	Coeff Var	Root MSE	ssigf1 Mean
0.598398	18.35978	3.207333	17.46934

Source	DF	Type I SS	Mean Square	F Value	Pr > F
tanner	4	6054.509502	1513.627376	147.14	<.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
tanner	4	6054.509502	1513.627376	147.14	<.0001

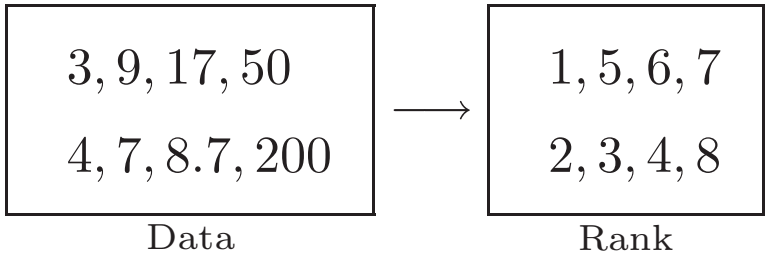
Parameter		Estimate		Standard Error	t Value	Pr > |t|
Intercept		21.49657843	B	0.29908531	71.87	<.0001
tanner	1	-7.93544551	B	0.37819314	-20.98	<.0001
tanner	2	-3.24333496	B	0.60013505	-5.40	<.0001
tanner	3	-0.19150204	B	0.73260639	-0.26	0.7939
tanner	4	1.26129188	B	0.64103059	1.97	0.0498
tanner	5	0.00000000	B	.	.	.

Nonparametric tests

Mann-Whitney test (alias Wilcoxon)

Kruskal-Wallis test

Idea: "t-test" or "ANOVA" on *ranks*



Distribution is (in principle) known under null hypothesis. Does not depend on data following a Normal distribution.

Other "scores" than ranks can also be used

Nonparametric tests in SAS

- ```
  proc npar1way wilcoxon data=sasuser.juul;
     where sexnr = 1 and 0 < age < 20;
     var sigf1;
     class tanner;
  run;
  ```
- (Mann-Whitney is obtained if there are only two groups to compare)
- It is the `wilcoxon` option that select rank scores and thus the Kruskal-Wallis/Mann-Whitney test, see manual for alternatives.
  ```

Output

The NPAR1WAY Procedure

Wilcoxon Scores (Rank Sums) for Variable sigf1
Classified by Variable tanner

tanner	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
1	192	20758.00	38496.00	1155.20917	108.114583
2	38	8222.00	7619.00	677.99178	216.368421
3	23	6569.50	4611.50	538.28582	285.630435
4	32	10387.00	6416.00	627.30283	324.593750
5	115	34263.50	23057.50	1046.52522	297.943478

Average scores were used for ties.

Kruskal-Wallis Test

Chi-Square	254.3465
DF	4
Pr > Chi-Square	<.0001

4. Regression and graphics

Use of SAS
December 2010

Contents

- Correlation
- Simple linear regression
- Scatter plots
- Histogram, Box plot, Probability plot
- Residual plots

Example: Obesity and blood pressure

```
proc print data=sasuser.bp;
  var sex obese bp;
run;
```

Obs	sex	obese	bp
1	male	1.31	130
2	male	1.31	148
3	male	1.19	146
.	.	.	.
.	.	.	.
101	female	1.64	136
102	female	1.73	208

Correlation

Is obesity related to blood pressure?

`proc corr` in SAS:

- Default is the *parametric correlation*, based on the bivariate normal distribution
  also denoted as the **Pearson correlation**
- The **Spearman correlation** is the most commonly used *nonparamtric* rank correlation
- The **Kendall correlation** is an alternative rank correlation

Correlation measures the strength of the (linear) association between two variables

The correlation coefficient is calculated as:

$$r = r_{\mathrm{xy}} = \frac{S_{\mathrm{xy}}}{\sqrt{S_{\mathrm{xx}} S_{\mathrm{yy}}}} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 \sum_{i=1}^{n} (y_i - \bar{y})^2}}$$

- takes on values between -1 and 1
- 0 corresponds to independence
- +1 and -1 correspond to perfect linearity
  positive resp. negative

Correlations in SAS

```
proc corr pearson spearman;
    var bp obese;
run;
```

```
Pearson Correlation Coefficients, N = 102
       Prob > |r| under H0: Rho=0

                bp         obese

bp          1.00000       0.32614
                           0.0008

obese       0.32614       1.00000
             0.0008

Spearman Correlation Coefficients, N = 102
       Prob > |r| under H0: Rho=0

                bp         obese

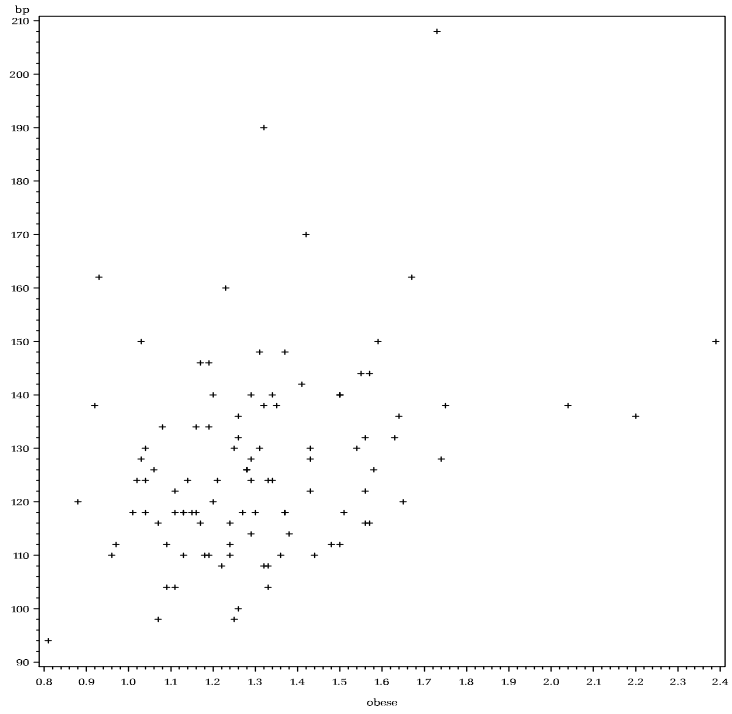
bp          1.00000       0.30363
                           0.0019

obese       0.30363       1.00000
             0.0019
```

Scatter plot

In raw form:

```
proc gplot;
  plot bp*obese;
run;
```



This plot can be improved a lot....

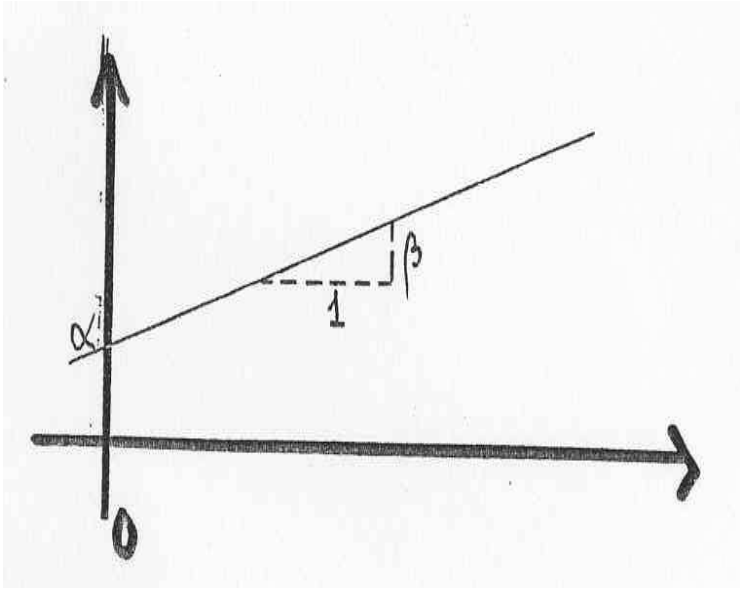
Linear regression

- **Y**: Response variable, outcome variable, dependent variable (here `bp`)
- **X**: Explanatory variable, independent variable, covariate (here `obese`)

Data: Bivariate observations of X and Y for a series of individuals or 'units':

$$(x_i, y_i), i = 1, \cdots, n$$

The equation for a straight line: $Y = \alpha + \beta X$



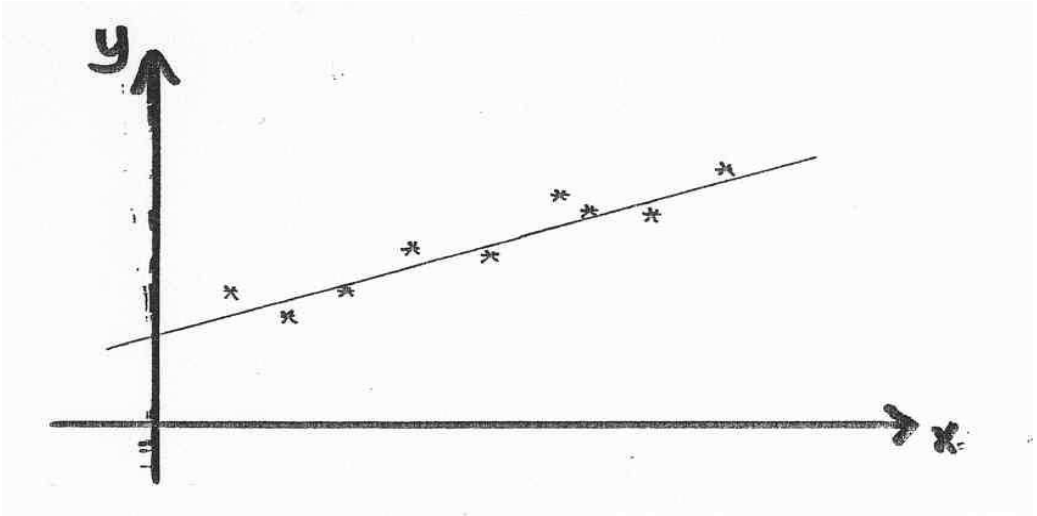
Interpretation:

- α: intercept, (intersection with Y-axis)
  The blood pressure for an individual with obesity 0.
  Often an illegal extrapolation.

- β: slope, regression coefficient
  The difference in blood pressure for two individuals with a difference in obesity of 1
  Often the parameter of interest.

The simple linear regression model

$$Y_i = \alpha + \beta X_i + \varepsilon_i, \quad \varepsilon_i \sim N(0, \sigma^2) \text{ indep.}$$



Estimation is performed using **least squares method**:

Determine α and β, to minimize

$$\sum_{i=1}^{n} (y_i - (\alpha + \beta x_i))^2 = \sum_{i=1}^{n} \varepsilon_i^2$$

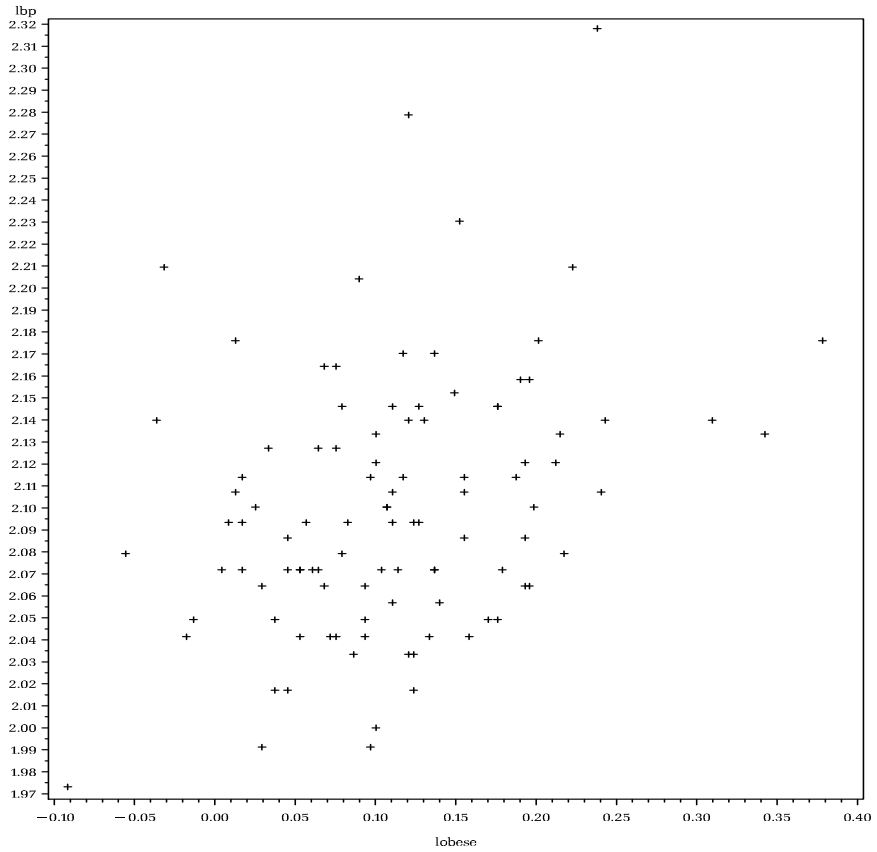
Assumptions in linear regression

- Linearity in the mean value
- Independence between *error terms* ε_i
- Normally distributed error terms, $\varepsilon_i \sim N(0, \sigma^2)$
- Variance homogeneity, i.e identical variances for all ε_i's

The last two assumptions are not quite fullfilled here, so we try a logarithmic transformation:

```
data sasuser.bp;
set sasuser.bp;
lbp=log10(bp);
lobese=log10(obese);
run;
```

After the logarithmic transformation:



Regression in SAS

```
proc reg data=sasuser.bp;
  model lbp=lobese;
run;
```

```
Dependent Variable: lbp
Analysis of Variance
                          Sum of          Mean
Source           DF      Squares        Square      F Value       Prob>F

Model             1      0.03809       0.03809       12.398       0.0006
Error           100      0.30727       0.00307
C Total         101      0.34536

    Root MSE       0.05543     R-square       0.1103
    Dep Mean       2.09983     Adj R-sq       0.1014
    C.V.           2.63983

Parameter Estimates

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

intercept  1      2.073139    0.00935900       221.513        0.0001
lobese     1      0.241193    0.06850116         3.521        0.0006
```

Interpretation

The estimated relation is
$\log_{10}(\text{bp}) = 2.073 + 0.241 \times \log_{10}(\text{obese})$

Interpretation: When $\log_{10}$-`obese` increases with one unit, $\log_{10}$-`lbp` will increase with 0.241 units

This can be *backtransformed* to the original scale:

- $\log_{10}(X_2) - \log_{10}(X_1) = 1 \Rightarrow$
  $X_2/X_1 = 10^1 = 10$

  Thus, a one unit increase in `lobese` corresponds to a 10-fold increase in `obese`

- $\log_{10}(Y_2) - \log_{10}(Y_1) = 0.241 \Rightarrow$
  $Y_2/Y_1 = 10^{0.241} = 1.74$

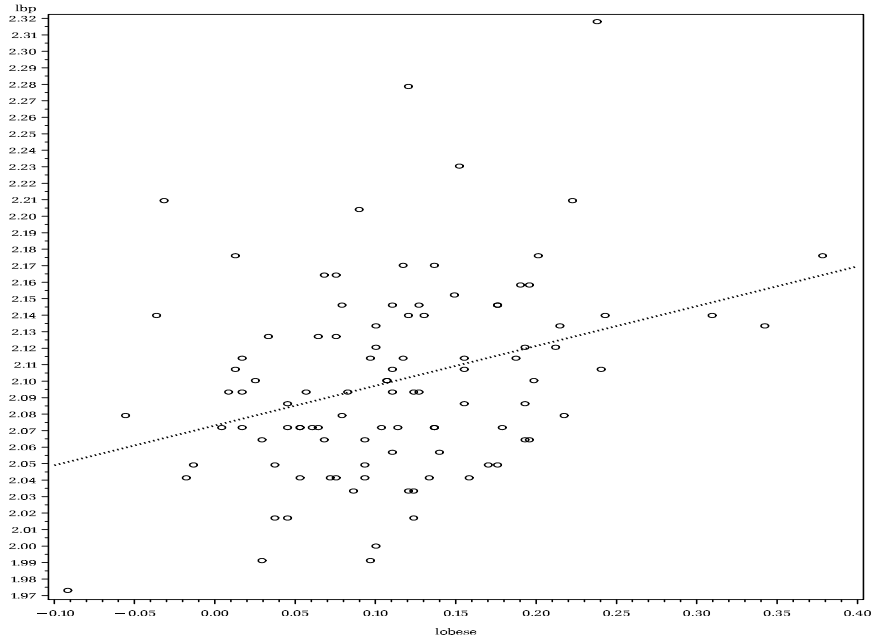
  Thus, an increase of 0.241 in `lbp` corresponds to a 74% incrase in `bp`

Conclusion: a 10-fold increase in `obese` results in a 74% incrase in `bp`

Add a regression line to the plot

```
proc gplot data=sasuser.bp;
  plot lbp*lobese;
symbol1 v=circle i=rl l=33;
run;
```

`i=rl` gives the regression line - `l=33` dotted linie



The variance around the regression line

σ^2 is estimated as

$$s^2 = \frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{\alpha} - \hat{\beta}x_i)^2$$

can be found in the output as
mean square error, here 0.00307

Standard error around the regression line
(sometimes - somewhat misleadingly - denoted '*residual standard error*')
here **root mean square error**

$$s = \sqrt{s^2} = 0.05543$$

This is also on a logarithmic scale, but in general it has the **same units** as the original outcome variable and is therefore **easier to interpret** than the variance.

Confidence limits

- confidence limits by hand:

$$\begin{aligned} & \hat{\beta} \pm t_{97.5\%}(n-2) \times se(\hat{\beta}) \\ = \quad & 0.241 \pm 1.984 \times 0.0685 = (0.105, 0.377) \end{aligned}$$

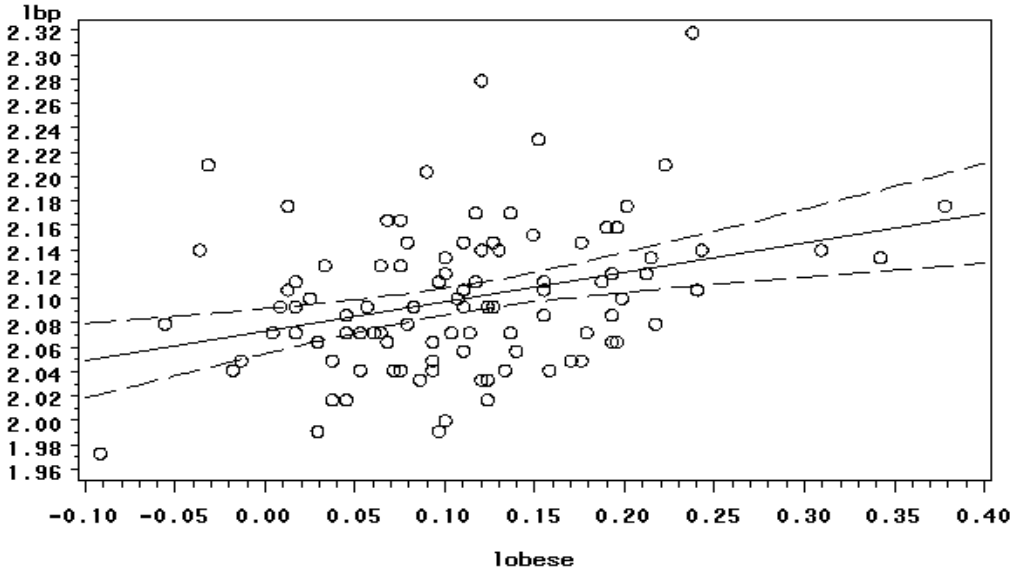
use the option `clb` in SAS:

```
proc reg data=sasuser.bp;
  model lbp=lobese / clb;
run;
```

Confidence interval for regression line

```
proc gplot;
  plot lbp*lobese;
symbol1 v=circle i=rlclm95 l=1;
run;
```

`i=irclm95` gives confidence limits



Exercise: Regression and graphics I

Consider again the Juul data. In this exercise we want to study the effect of age on the SIGF1-level.

1. Get the data into SAS using a libname statement.
2. Create a new data set containing only prepubertal children (Tanner stage 1 and age > 5).
3. Use PROC GPLOT to plot the relationship between $\sqrt{\text{SIGF1}}$ and age for prepubertal children.
4. Use PROC REG to do a regression analysis of $\sqrt{\text{SIGF1}}$ vs. age for prepubertal children.

Scatter plots in SAS: PROC GPLOT

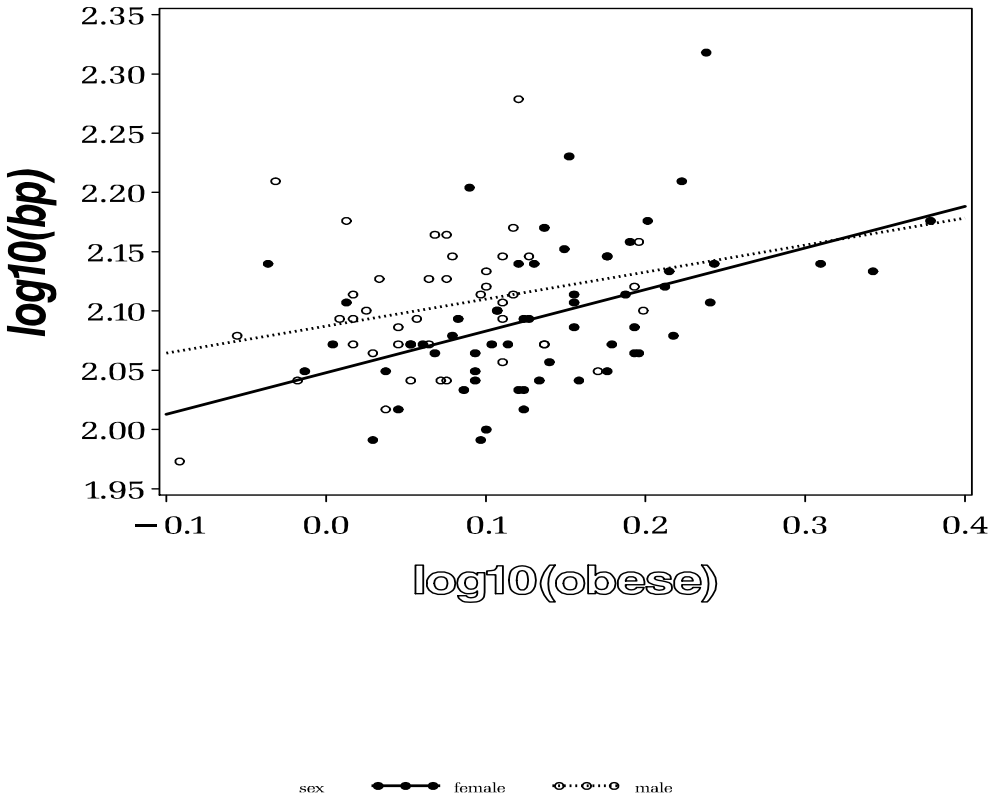
In the raw form:

```
proc gplot;
  plot bp*obese;
run;
```

more code can give nicer output:

```
proc gplot data=sasuser.bp;
  plot lbp*lobese=sex
  / haxis=axis1 vaxis=axis2 frame;
axis1 value=(H=2)
      minor=NONE
      label=(H=3 F=swissbe 'log10(obese)');
axis2 order=1.95 to 2.35 by 0.05
      length=12 cm
      value=(H=2)
      minor=NONE
      label=(A=90 R=0 H=3 F=swissbi 'log10(bp)');
symbol1 v=dot i=rl c=BLACK l=1 w=2;
symbol2 v=circle i=rl c=BLACK l=33 w=2;
title1 F=cscript h=3 'obesity and blood pressure';
run;
```

we have included statements on: 'axis', 'symbol' and 'title'


log10(bp)
2.35
2.30
2.25
2.20
2.15
2.10
2.05
2.00
1.95
−0.1
0.0
0.1
0.2
0.3
0.4
log10(obese)
sex
female
male

Symbol statements

One symbol statement for each group
(each value of `sex`)

Options:

- `v=circle`: plotting symbol: circle/dot/star
- `h=2`: the size of the plotting symbol
  (default 1)
- `i=none`: interpolation method:
  `none/join/rl/rlcli95/rlclm95`
- `c=black`: colour of points: `black/red/blue`
- `l=1`: line type,
  1:solid, 2-46: different dashings

Plotting symbols

`'v=  '` in `symbol` statements

VALUE=	Plot Symbol	VALUE=		Plot Symbol	VALUE=		Plot Symbol
PLUS	+	_	(underscore)	□	+	(plus)	⊕
X	×	"	(double quote)	♤	>	(greater than)	♂
STAR	✳	#	(pound sign)	♡	.	(period)	♃
SQUARE	□	$	(dollar sign)	♢	<	(less than)	♄
DIAMOND	◇	%	(percent)	♧	,	(comma)	☉
TRIANGLE	△	&	(ampersand)	☘	/	(slash)	♆
HASH	#	'	(single quote)	⚜	?	(question mark)	♇
Y	Y	=	(equals)	☆	(	(left parenthesis)	☾
Z	Z	-	(hyphen)	⊙	)	(right parenthesis)	☌
PAW	⁘	@	(at)	☿	:	(colon)	✳
POINT	·	*	(asterisk)	♀			
DOT	●						
CIRCLE	○						

Note: The special symbols in this table are listed in default order.

Note: Only the values in column one are specified by name. The values in columns two and three are specified only by character. The names of the characters are included for clarity only.

Interpolation methods

`'i=  '` (or `'interpol=  '`) in `symbol` statements

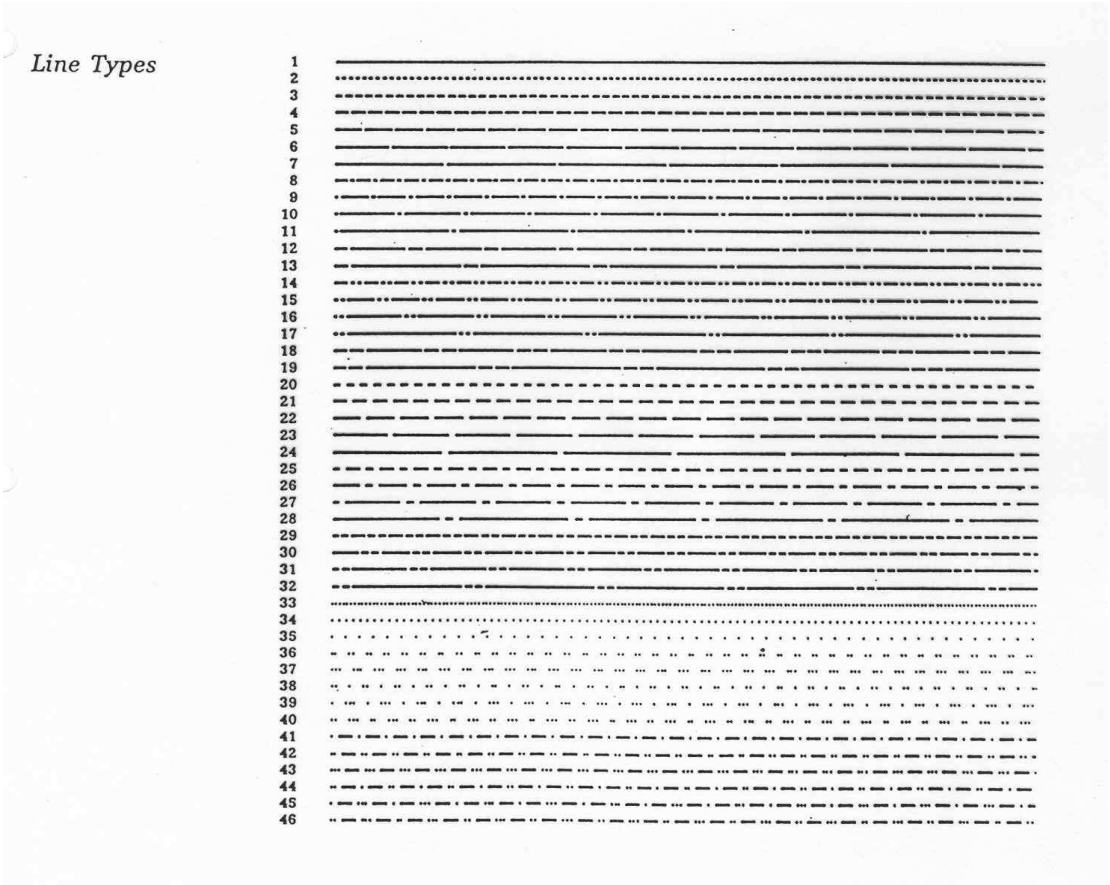
Table 19.2 *Selected Interpolation Methods*

If your data have . . .	Then you might choose one of these methods . . .	And specify INTERPOL=
one Y value for each X value	join	JOIN
	fitting a regression line	R<L | C | Q><*options*>
	needle	NEEDLE
	spline	SPLINE<*options*>
	spline with Lagrange interpolation	L<1 | 3 | 5><*options*>
	spline with user-defined smoothing	SM<0 . . . 99><*options*>
one or more Y values for each X value	fitting a regression line	R<L | C | Q><*options*>
	spline	SPLINE<*options*>
	spline with Lagrange interpolation	L<1 | 3 | 5><*options*>
	spline with user-defined smoothing	SM<0 . . . 99><*options*>
several Y values for each X value	box plots	BOX<*options*>
	high-low or high-low-close	HILO<C><*options*>
	standard deviation	STD<1 | 2 | 3><*options*>

Note: If you do not specify an interpolation method, the GPLOT procedure simply marks the data points with the plot symbol. This is equivalent to specifying INTERPOL=NONE.

Line types

`'l=  '` in `symbol` statements



AXIS specifications

- `haxis=axis1 vaxis=axis2`: horizontal axis is axis1 and vertical axis is axis2. axis1 and axis2 must be specidfied.
- `length=12cm`: the length of the axis
- `value=(h=2)`: the size of the digits on the axis
- `minor=9`: number of tickmarks between the numbers, may be set to `none`
- `label=(A=90 R=0 h=2 'text')` specifies the axis text, the size of this,
  and its direction
  - `A=90`: The whole text has to be rotated 90 degrees counterclockwise, so that it fits the Y axis
  - `R=0` this may make *the letters* slant
- `order=(0 to 10 by 1)` specifies the desired numbers on the axis

Fonts in SAS

`'f=  '` in `symbol`, `axis` or `title`

Type Style	Font Name	Type Sample	Uniform Font
Brush	BRUSH	*A B C a b c 1 2 3*	
Century			
Bold	CENTB	**A B C a b c 1 2 3**	CENTBU
Bold Empty	CENTBE	A B C a b c 1 2 3	
Bold Italic	CENTBI	***A B C a b c 1 2 3***	CENTBIU
Bold Italic Empty	CENTBIE	*A B C a b c 1 2 3*	
Expanded	CENTX	**A B C a b c 1 2 3**	CENTXU
Expanded Empty	CENTXE	A B C a b c 1 2 3	
Expanded Italic	CENTXI	***A B C a b c 1 2 3***	CENTXIU
Expanded Italic Empty	CENTXIE	*A B C a b c 1 2 3*	
German	GERMAN	𝔄 𝔅 ℭ a b c 1 2 3	GERMANU
German Italic	GITALIC	𝔄 𝔅 ℭ a b c 1 2 3	GITALICU
Hershey			
Sans Serif	SIMPLEX	A B C a b c 1 2 3	SIMPLEXU
Sans Serif Bold	DUPLEX	A B C a b c 1 2 3	DUPLEXU
Serif	COMPLEX	A B C a b c 1 2 3	COMPLEXU
Serif Bold	TRIPLEX	A B C a b c 1 2 3	TRIPLEXU
Serif Bold Italic	TITALIC	*A B C a b c 1 2 3*	TITALICU
Serif Italic	ITALIC	*A B C a b c 1 2 3*	ITALICU
Old English	OLDENG	𝔄 𝔅 ℭ a b c 1 2 3	OLDENGU
Script	SCRIPT	*A B C a b c 1 2 3*	
Cscript	CSCRIPT	*A B C a b c 1 2 3*	
Simulate	SIMULATE	A B C a b c 1 2 3	SIMULATE
Swiss	SWISS	**A B C a b c 1 2 3**	SWISSU
Empty	SWISSE	A B C a b c 1 2 3	
Bold	SWISSB	**A B C a b c 1 2 3**	SWISSBU
Bold Empty	SWISSBE	A B C a b c 1 2 3	
Bold Italic	SWISSBI	***A B C a b c 1 2 3***	SWISSBIU

(continued)

Histograms in SAS

```
proc univariate data=sasuser.bp;
  var lbp;
  class sex;
  histogram / cfill=gray
              endpoints=1.9 to 2.3 by 0.1 normal;
  inset mean std skewness / header='descriptive';
run;
```

`histogram`: gives a histogram for variable `lbp`

`cfill=gray`: bars are gray

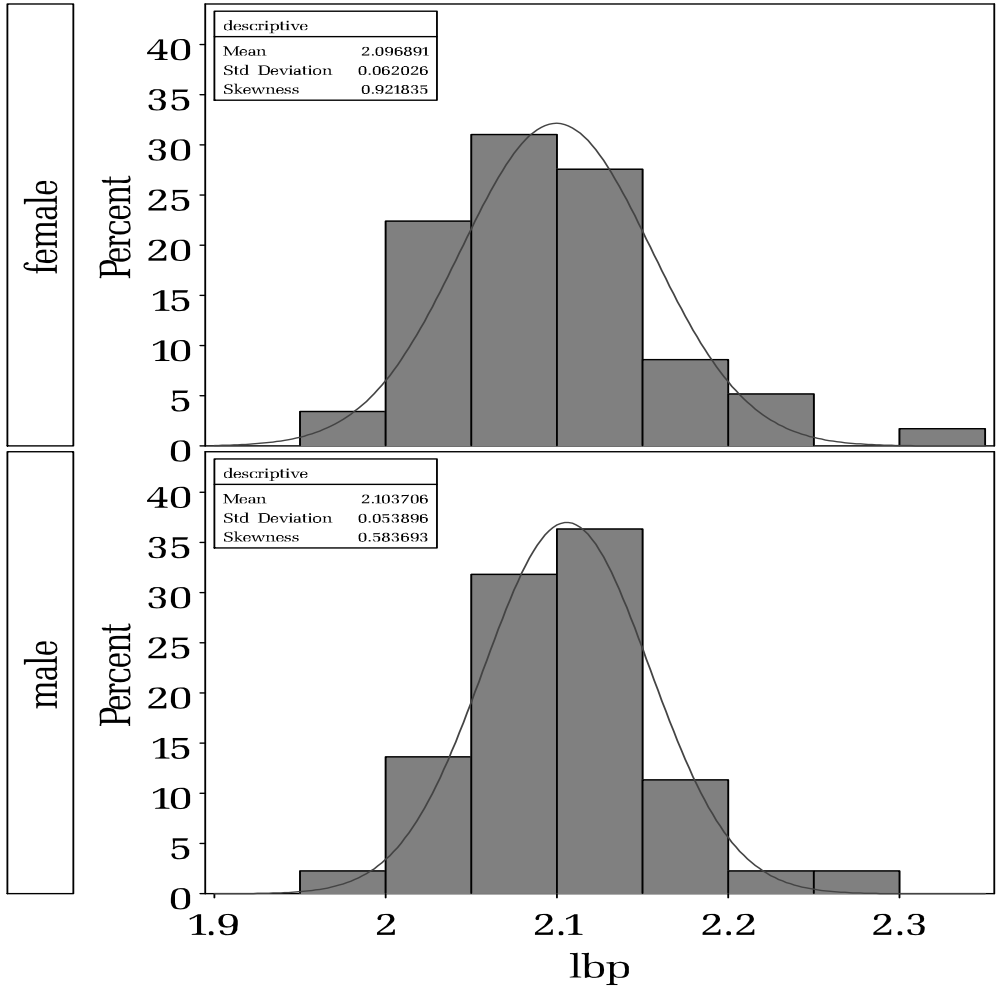
`class sex`: histogram for both values of `sex`

`endpoints=1.9 to 2.3 by 0.1`: numbers on x-axis

`normal`: the best fitting normal curve is included

`inset mean std skewness / header='descriptive'`: header is included

PROC UNIVARIATE with HISTOGRAM



Probability plots

```
proc univariate data=sasuser.bp;
  var lbp;
  class sex;
  probplot / height=3 normal(mu=EST sigma=EST l=33);
run;
```

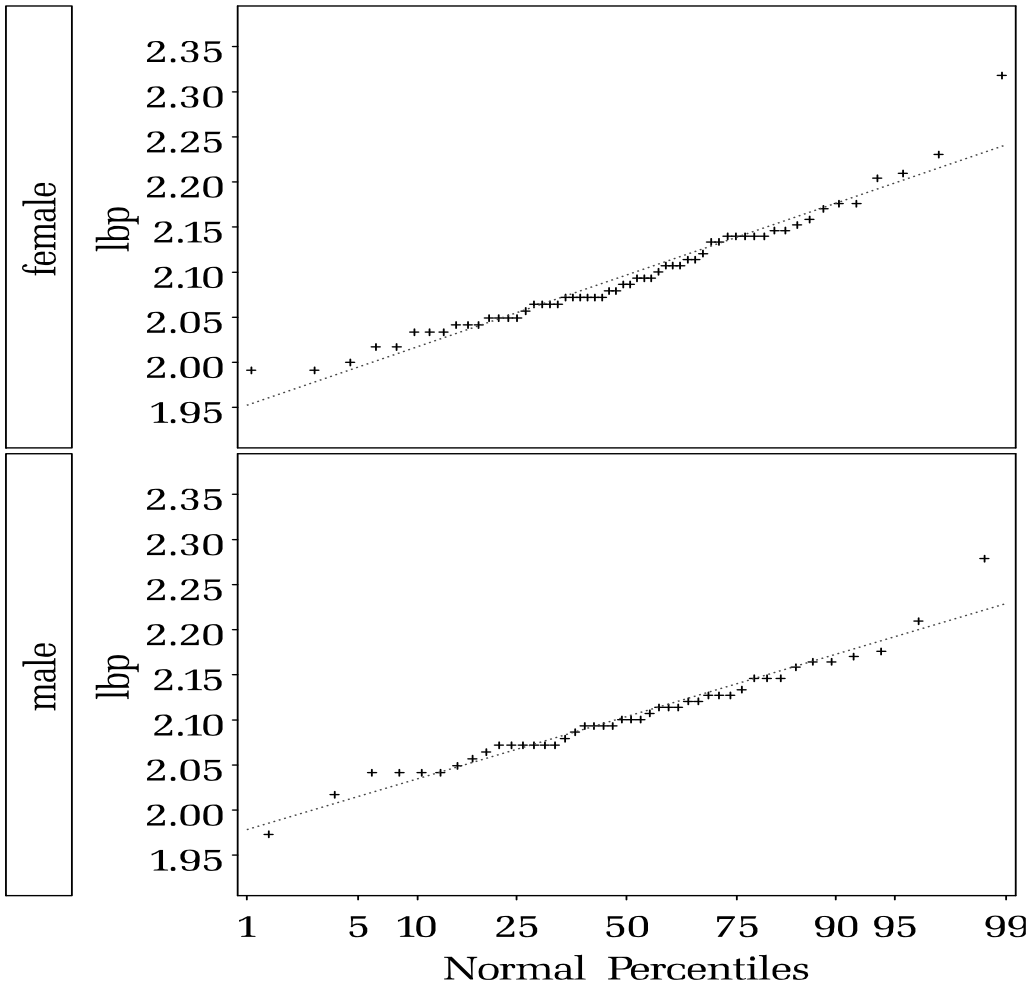
`probplot`: gives a probability plot for variable `lbp`

`class sex`: plot for both values of `sex`

`height=3`: size of the text

`normal(mu=EST sigma=EST l=33)` shows the line of the best fitting normal distribution. The line is dotted (`l=33`).

PROC UNIVARIATE with PROBPLOT



Box plots

```
proc boxplot  data=sasuser.bp;
  plot lbp*sex / height=3 boxstyle=schematic;
run;
```

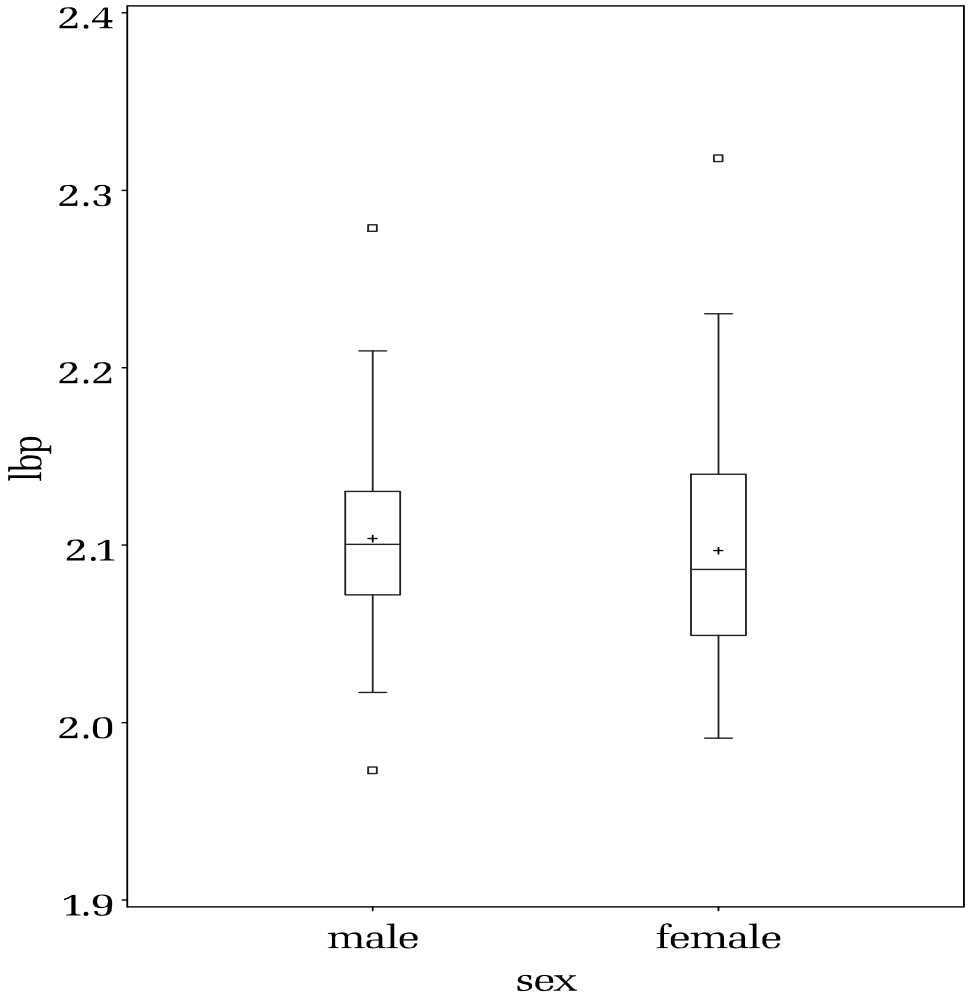
use `proc boxplot` *not* `proc univariate`

`lbp*sex`: the distribution of `lbp` for each value of `sex`

`height=3`: size of the text

`boxstyle=schematic`: specifies the type of boxplot (there are many)

PROC BOXPLOT



How to save graphs

SAS can use the 'output delivery system' (ODS) to direct output from a SAS procedure to other places such as data sets or files

As we shall see, out put can also be saved into data set using the 'output out' command

Importantly, graphs can be saved using ODS

```
ods rtf file='p:\eksempel.rtf';
proc gplot;
---;
run; quit;
ods rtf close;
```

this generates a file in 'rich text format' (RTF), which can be read by Word.

Model control in regression analysis

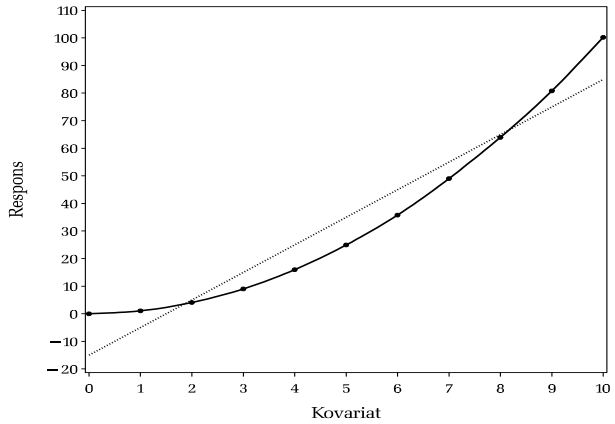
Residuals: $\hat{\varepsilon}_i = y_i - \hat{y}_i = y_i - (\hat{\alpha} + \hat{\beta} x_i)$

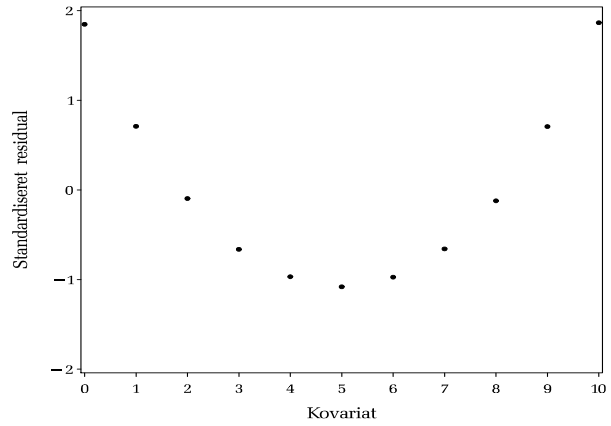
Residuals are plotted against:

- the explanatory variable x_i
  – to check linearity
- the fitted values $\hat{y}_i$
  – to check variance homogeneity
  (and normality)
- '*normal scores*' i.e. probability plot
  – to check normality

The first two types ought to give an impression of pure scatter, while the probability plot ought to show a straight line.

Residual-plots and linearity





Various types of residuals

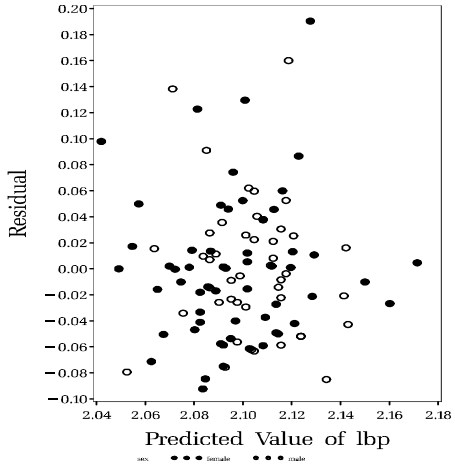
- Ordinary residuals = model deviations: $\hat{\varepsilon}_i = y_i - \hat{y}_i$
  in SAS denoted `r`
- **Normalised** residuals, also denoted **standardised** residuals or Student residuals
  in SAS denoted `student`

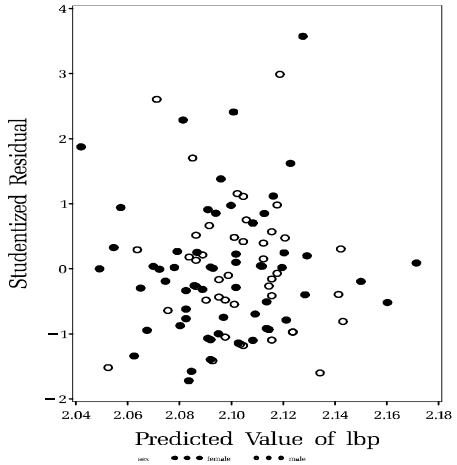
In the procedure **REG** we can calculate **and save** all these quantities for later use, typically graphics:

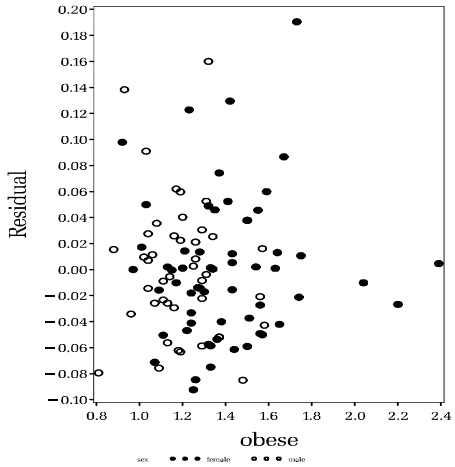
```
proc reg data=sasuser.bp;
  model lbp=lobese;
output out=res p=yhat r=resid student=student;
run;
```

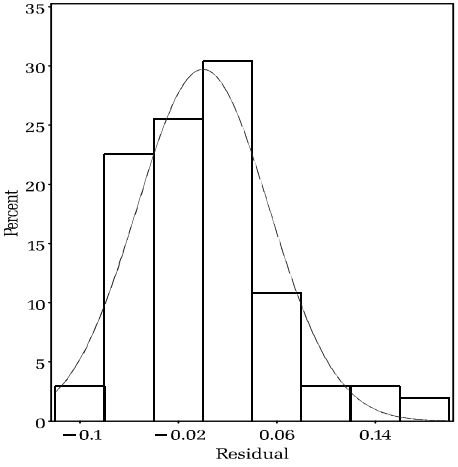
Here, we get a new data set `res` (`work.res`) containing 3 new variables (`yhat`, `resid`, `student`), which we may then use, e.g. to make a residual plot:

```
proc gplot data=res;
plot student*yhat;
run;
```


Residual
Predicted Value of lbp
sex
female
male


Studentized Residual
Predicted Value of lbp
sex
female
male


Residual
obese
sex
female
male


Percent
Residual

Exercise: Regression and graphics II

Consider again the regression analysis of $\sqrt{\text{SIGF-I}}$ vs. age for prepubertal children (Tanner stage 1 and age > 5).

1. Modify your code from exercise I to calculate residuals and expected values based on the regression model (use an OUTPUT statement).
2. Make residual plots as well as scatter plots of data with estimated regression lines. Use different symbols for the two genders.

5. More about the SAS language

Use of SAS
December 2010

Lists of variables

- Sometimes you need to refer to many variables at once
- E.g., if you have repeated measurements or just many similar variables

```
proc freq;
  tables spm1-spm392;
run;
```

- similar names `x1-x20`
- All character variables: `_CHARACTER_`
- All numerical variables: `_NUMERIC_`
- All variables: `_ALL_`

Labels

```
libname juul 'p:\sas\data\juul';
proc freq data=juul.juul2;
  table tanner;
run;
data hope;
  set juul.juul2;
  label tanner="Tanner stage";
run;
proc freq data=hope;
  table tanner;
run;
```

Formats

- Information about how to read or print variable
- Built-in formats (Numerical, dates, character)
- User defined formats (1=Male 2=Female)
- Pretty printouts
- Grouping in tables and analyses

Formats, cont.

- Standard formats: `10.3`, `best12.`, `E12.`, `$10.`, `date10.`, `yymmdd10.`.
- Always contain a dot (Do not forget it!)
- Can be associated permanently with variable in DATA step, or:
- Specified ad hoc with FORMAT statement in PROC steps.
- User defined formats are created and listed by PROC FORMAT

Example with proc format

```
proc format;
   value sexfor 1="male" 2="female";
run;
data a;
   input sex;
   datalines;
   1
   2
   3
   .
   ;
run;
proc print data=a;
run;
proc print data=a;
  format sex sexfor.;
run;
```

Exercise using formats

1. Get the bissau data into SAS using a libname statement
2. Use bissau data, generate a sas program creating formats for variables
   - `dead`: 1=died 2=Survived
   - `bcg`: 1=yes 2=No
   - `dtp`: 1=yes 2=No

   Help for the first one:

```
proc format;
  value deadfmt 1=Died 2=Survived;
run;
```

3. Make a `proc freq` of the three variables using your formats.

Using formats

Generate data

```
data test;
reply=1; do i=1 to 25; output; end;
reply=2; do i=1 to 21; output; end;
reply=3; do i=1 to 35; output; end;
run;
proc print data=test;
run;
```

Generate format for the variable reply

```
proc format;
value replyfor 1='Yes     '
               2='No    '
               3='Maybe  ';
run;
```

Distribution of the variable reply

```
proc freq data=test;
table reply; run;
```

Formats in PROC and DATA steps

The format used in a proc step

```
proc freq data=test;
table reply;
format reply replyfor.;
run;
```

or the format can be associated with the variable in a data step

```
data testfor;
set test;
format reply replyfor.;
run;
```

now the format will be used every time we use the data: testfor

```
proc freq data=testfor;
  table reply;
run;
```

Save data in a permanent data set

```
libname pdrev 'p:\';
data pdrev.testfor;
set testfor;
run;
```

Restart SAS and run the program

```
libname pdrev 'p:\';
proc freq data=pdrev.testfor;
table reply;
run;
```

SAS cannot find the format!

```
proc freq data=pdrev.testfor;
table reply;
format reply; *format _all_;
run;
```

`format _all_;` removes the formats. Can be very useful

A more advanced PROC FORMAT example:

```
proc format;
  value agegrpfmt 0-1="0-1" 2-4="2-4" 5-6="5-6";
run;
proc format fmtlib;run;
proc freq data=afrika.bissau;
  table agemm;
proc freq data=afrika.bissau;
  table agemm;
  format agemm agegrpfmt.;
run;
```

NB: To be able to use your created formats next time you start SAS, you can save the SAS code in a file, and run this the next time you will use the data set.

Date formats

- Actual value stored is the number of days since 1 January 1960.

```
data hope;
  input x;
datalines;
-1
0
1
;
run;
proc print data=hope;
   format x ddmmyy10.;
run;
```

- function `mdy()` month-day-year:

```
data a;
  x=mdy(1,17,2006);
run;
proc print data=a; run;
proc print data=a;  format x ddmmyyd.;run;
```

- Final `d` in the output format indicates "dash". Other possibility `c,s,n,p` (colon, slash, none, period)

Working with dates

As dates internally are stored as days since 1 Jan 1960, one can add and subtract dates and constants:

```
data prv;
  input nr dead DDMMYY10.;
  datalines;
  1 9-01-1975
  2 12-12-1956
run;

proc print data=prv;
run;

data prv;
  set prv;
  thisday=today();
  days=thisday-doed;
  years=days/365.25;
run;

proc print data=prv;
run;
```

Exercise with dates

In the SAS data set `bissau2.sas7bdat` (in the africa directory) the first 200 observations are from the original Bissau data. Variables are:

```
id        = ID of child
dob       = Date of birth
visitdate = Date of visit
agedays   = Age in days at visit
```

Please, check that the variable `agedays` was correctly calculated.

Appending data sets: SET

more cases, same variables

```
data group0;
  set sasuser.fitness;
  where group=0;
  comment1="Data1";
  keep group comment1;
run;
data group1;
  set sasuser.fitness;
  where group=1;
  comment2="Data2";
  keep group comment2;
run;
data group2;
  set sasuser.fitness;
  where group=2;
  comment3="Data3";
  keep group comment3;
run;
data all;
  set group0 group1 group2;
run;
proc print;
run;
```

Obs	group	comment1	comment2	comment3
1	0	Data1		
2	0	Data1		
3	0	Data1		
4	0	Data1		
5	0	Data1		
6	0	Data1		
7	0	Data1		
8	0	Data1		
9	0	Data1		
10	0	Data1		
11	1		Data2	
12	1		Data2	
13	1		Data2	
14	1		Data2	
15	1		Data2	
16	1		Data2	
17	1		Data2	
18	1		Data2	
19	1		Data2	
20	1		Data2	
21	2			Data3
22	2			Data3
23	2			Data3
24	2			Data3
25	2			Data3
26	2			Data3
27	2			Data3
28	2			Data3
29	2			Data3
30	2			Data3
31	2			Data3

Merging data set: MERGE

new variables, same cases. Normally there is a key, say `id`, and all data sets must be sorted by `id`

```
data name;
  input id name $6.;
datalines;
1 Henrik
2 Esben
3 Peter
;
data surname;
  input id sname $15.;
datalines;
2 Budtz-Jørgensen
1 Jensen
;
proc sort data=surname;
  by id;
data fullname;
  merge name surname;
  by id;
proc print data=fullname;
run;
```

Obs	id	name	sname
1	1	Henrik	Jensen
2	2	Esben	Budtz-Jørgensen
3	3	Peter	

Exercise: More about MERGE

In the library `'p:\sas\prg'` you will find the file `'exercise_merge.sas'`.Run the first part of the code.

```
data name;
  input fam name $6.;
datalines;
1 Henrik
1 Gustav
2 Esben
run;
data surname;
  input fam sname $15.;
datalines;
2 Budtz-Jørgensen
1 Jensen
run;

proc print data=name;
run;
```

```
proc print data=surname;
run;
```

This gives the output:

Obs	fam	name
1	1	Henrik
2	1	Gustav
3	2	Esben

Obs	fam	sname
1	2	Budtz-Jørgensen
2	1	Jensen

We want to merge the two data sets so that the names and surnames are correctly matched (Henrik and Gustav are called Jensen while Esben is called Budtz-Jørgensen). The following code gives to possible solustions. Run the code and explain the differences in the solutions.

```
*solution 1;
data fullname1;
  merge name surname;
run;
proc print data=fullname1;
run;
*solution 2;
proc sort data=surname;
  by fam;
run;
data fullname2;
  merge name surname;
  by fam;
run;
proc print data=fullname2;
run;
```

6. The general linear model

Use of SAS
December 2010

Contents

- Analysis of covariance
- Interaction
- Multiple regression
- The general linear model

Example on lung capacity

32 patients for heart/lung transplantation

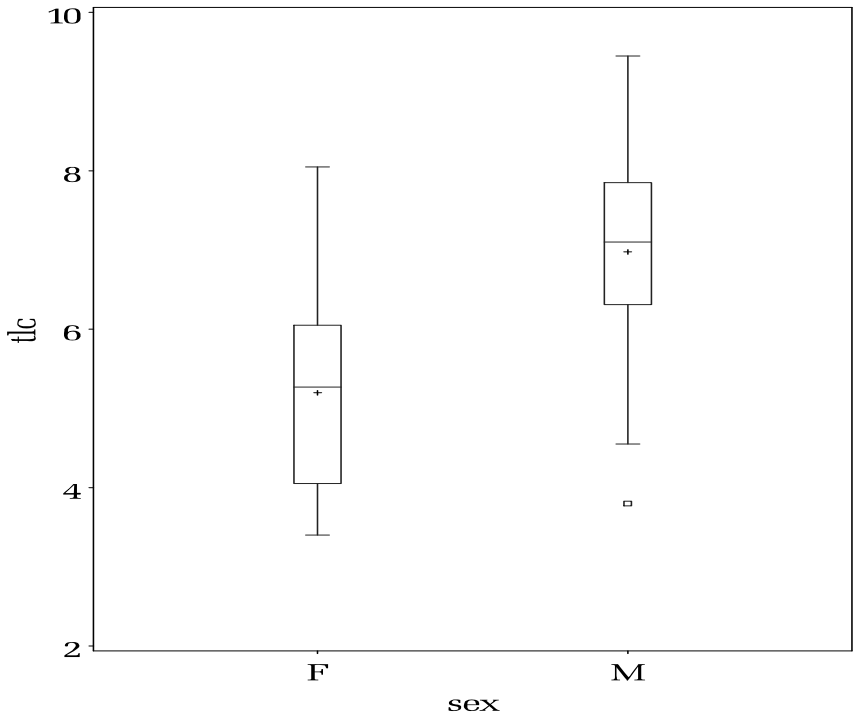
`TLC` (Total Lung Capacity) is determined from whole-body plethysmography

Are men and women different with respect to total lung capacity?

OBS	SEX	AGE	HEIGHT	TLC
1	F	35	149	3.40
2	F	11	138	3.41
3	M	12	148	3.80
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
29	F	20	162	8.05
30	M	25	180	8.10
31	M	22	173	8.70
32	M	25	171	9.45

Box plots for comparison of gender groups

```
proc boxplot data=tlc;
  plot tlc*sex / height=3 boxstyle=schematic;
run;
```



Marginal comparisons

```
proc ttest data=tlc;
  class sex;
  var tlc height;
run;
```

The TTEST Procedure

Statistics

Variable	sex	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev
tlc	F	16	4.505	5.1981	5.8913	0.9609	1.3008
tlc	M	16	6.2106	6.9769	7.7431	1.0623	1.438
tlc	Diff (1-2)		-2.769	-1.779	-0.789	1.0957	1.3711
height	F	16	155.82	160.81	165.8	6.9203	9.3682
height	M	16	168.38	174.06	179.74	7.8755	10.661
height	Diff (1-2)		-20.5	-13.25	-6.004	8.0195	10.036

Statistics

Variable	sex	Upper CL Std Dev	Std Err	Minimum	Maximum
tlc	F	2.0133	0.3252	3.4	8.05
tlc	M	2.2256	0.3595	3.8	9.45
tlc	Diff (1-2)	1.8328	0.4848		
height	F	14.499	2.342	138	177
height	M	16.5	2.6653	148	189
height	Diff (1-2)	13.414	3.5481		

T-Tests

Variable	Method	Variances	DF	t Value	Pr > |t|
tlc	Pooled	Equal	30	-3.67	0.0009
tlc	Satterthwaite	Unequal	29.7	-3.67	0.0009
height	Pooled	Equal	30	-3.73	0.0008
height	Satterthwaite	Unequal	29.5	-3.73	0.0008

Equality of Variances

Variable	Method	Num DF	Den DF	F Value	Pr > F
tlc	Folded F	15	15	1.22	0.7028
height	Folded F	15	15	1.30	0.6228

Obvious gender difference for `tlc` as well as `height`

Confounding when comparing groups

- occurs if the distribution of an important explanatory variable differ between the groups

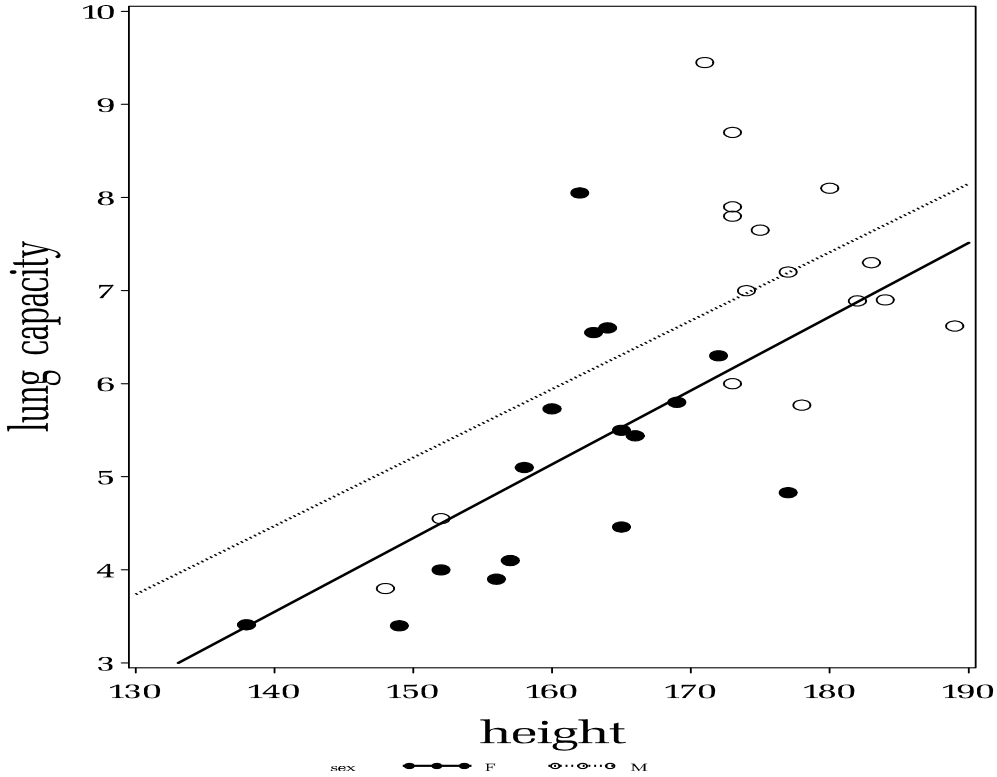
Can be avoided by performing a **regression analysis** with the relevant variables as covariates.

Example:

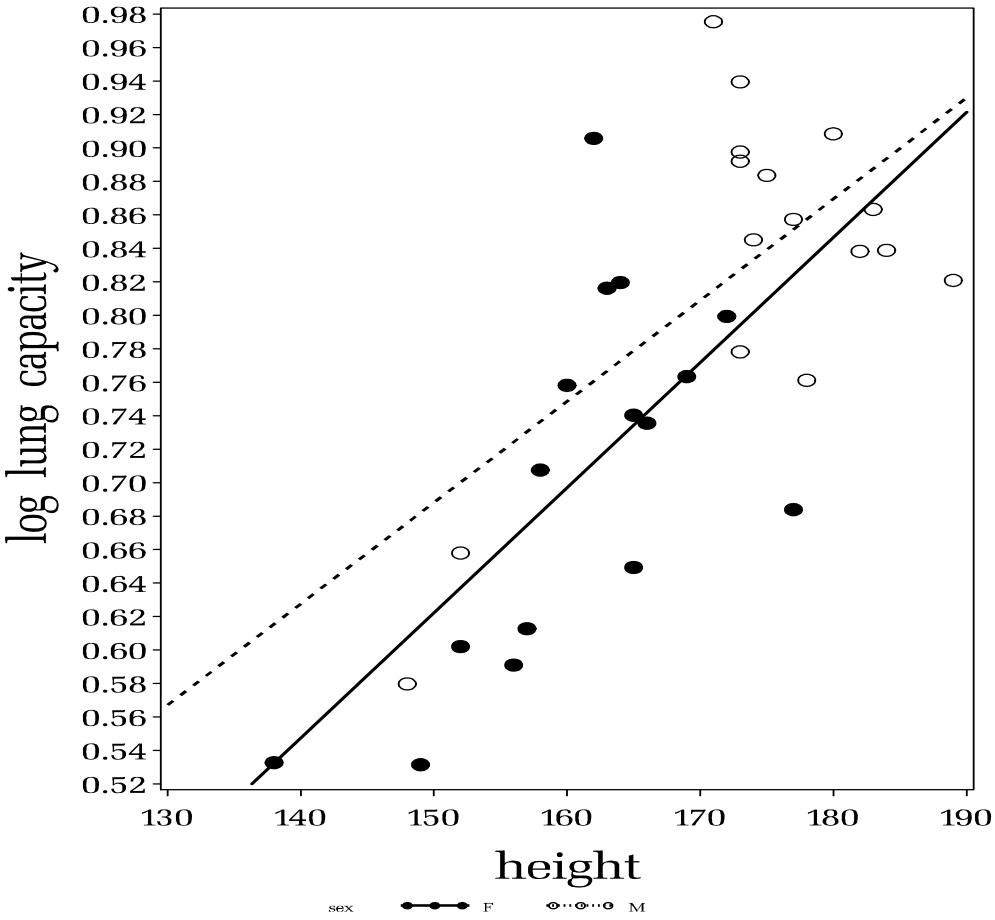
- Comparison of lung function between men and women
  — they are not of equal height

Relation between `tlc` and `height`:

```
proc gplot;
  plot tlc*height=sex;
symbol1 v=dot i=rl c=BLACK l=1 w=2 h=2;
symbol2 v=circle i=rl c=BLACK l=33 w=2 h=2;
run;
```

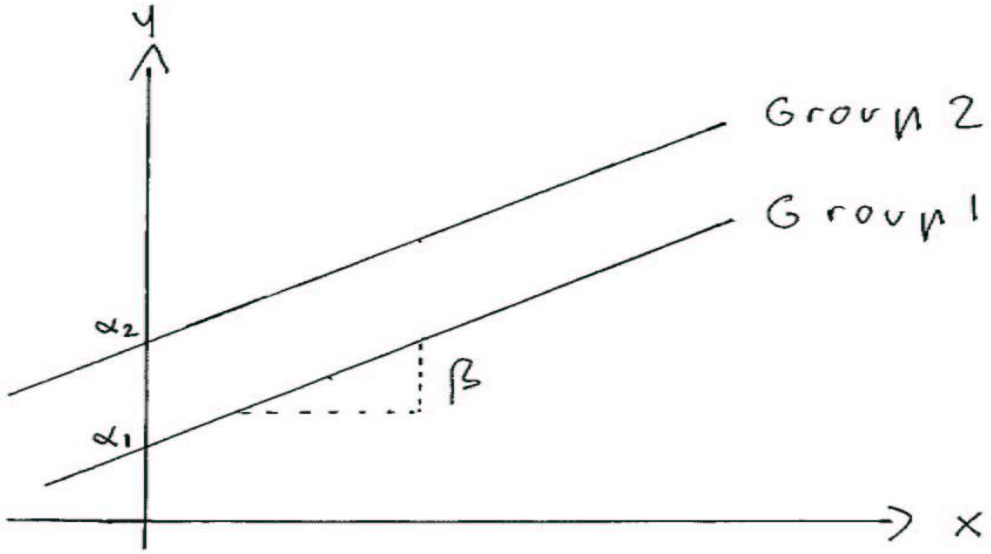


Relation between `tlc` (after transformation with base 10 logarithms) and `height`,



Analysis of covariance

Comparison of **parallel** regression lines



Model:

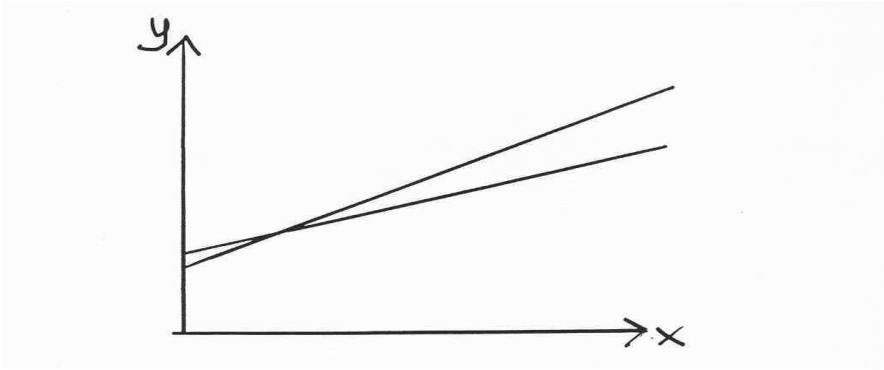
$y_{gi} = \alpha_g + \beta x_{gi} + \varepsilon_{gi} \qquad g = 1, 2; i = 1, \cdots, n_g$

Here $\alpha_2 - \alpha_1$ is the expected difference in the response between the two groups *for fixed* value of the covariate

We have adjusted for x.

But what if the lines are not at all parallel?

More **general model:** $y_{gi} = \alpha_g + \beta_g x_{gi} + \varepsilon_{gi}$



When $\beta_1 \neq \beta_2$, we say that there is **interaction** between `height` and `sex`

- The effect of height depends on gender
- The difference between men and women depends on height

In case of interaction: Do not interpret marginal effects.

Model with interaction

```
proc glm data=tlc;
  class sex;
  model ltlc=sex height sex*height / solution;
run;
```

```
The GLM Procedure
   Class Level Information
Class         Levels    Values
sex                2    F M

Number of observations    32

Dependent Variable: ltlc
                                   Sum of
Source                     DF       Squares   Mean Square  F Value  Pr > F

Model                       3    0.27230446    0.09076815    13.05  <.0001
Error                      28    0.19478293    0.00695653
Corrected Total            31    0.46708739


R-Square     Coeff Var      Root MSE     ltlc Mean

0.582984      10.85524      0.083406      0.768346
```

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sex	1	0.13626303	0.13626303	19.59	0.0001
height	1	0.13451291	0.13451291	19.34	0.0001
height*sex	1	0.00152852	0.00152852	0.22	0.6429

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sex	1	0.00210426	0.00210426	0.30	0.5867
height	1	0.13597107	0.13597107	19.55	0.0001
height*sex	1	0.00152852	0.00152852	0.22	0.6429

Parameter		Estimate	Standard Error	t Value	Pr > |t|
Intercept		-.2190181620 B	0.35221658	-0.62	0.5391
sex	F	-.2810587157 B	0.51102682	-0.55	0.5867
sex	M	0.0000000000 B	.	.	.
height		0.0060473650 B	0.00201996	2.99	0.0057
height*sex	F	0.0014344422 B	0.00306016	0.47	0.6429
height*sex	M	0.0000000000 B	.	.	.

NOTE: The X'X matrix has been found to be singular, and a generalized inverse was used to solve the normal equations. Terms whose estimates are followed by the letter 'B' are not uniquely estimable.

Where are the two lines in the output?

Linie for males (the reference group):

$$\log10(\text{Lung capacity}) = -0.219 + 0.00605 \times \text{height}$$

Linie for females:

$$\begin{aligned} \log10(\text{Lung capacity}) &= -0.219 + (-0.281) + (0.00605 + 0.00143) \times \text{height} \\ &= -0.500 + 0.00748 \times \text{height} \end{aligned}$$

Same model, new parametrisation

```
proc glm data=tlc;
class sex;
  model ltlc=sex sex*height / noint solution; run;

...
Source                     DF     Type I SS   Mean Square  F Value  Pr > F

sex                         2   19.02765491    9.51382745  1367.61  <.0001
height*sex                  2    0.13604143    0.06802071     9.78  0.0006

Source                     DF   Type III SS   Mean Square  F Value  Pr > F

sex                         2    0.01537968    0.00768984     1.11  0.3451
height*sex                  2    0.13604143    0.06802071     9.78  0.0006

                                        Standard
Parameter             Estimate             Error    t Value    Pr > |t|

sex         F     -.5000768777       0.37025922      -1.35      0.1876
sex         M     -.2190181620       0.35221658      -0.62      0.5391
height*sex  F     0.0074818072       0.00229877       3.25      0.0030
height*sex  M     0.0060473650       0.00201996       2.99      0.0057
```

Same model, 2 different parametrisations

```
proc glm data=tlc; class sex;
  model ltlc=sex height sex*height / solution;
run;
```

- One level for the reference group (`sex='M'` and `height=0`)
- A difference between genders (at `height=0`)
- An effect of `height` (slope) for the reference group
- A difference in slopes for the genders

```
proc glm data=tlc; class sex;
  model ltlc=sex sex*height / noint solution;
run;
```

- A level for each group (`sex`) (at `height=0`)
- An effect of `height` (slope) for each group (`sex`)

Here:

No indication of interaction, we omit the term

```
proc glm data=tlc;
  class sex;
  model ltlc=sex height / solution clparm;
run;
```

```
The GLM Procedure
 Dependent Variable: ltlc

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

Model                       2    0.27077594    0.13538797    20.00  <.0001
Error                      29    0.19631145    0.00676936
Corrected Total            31    0.46708739

R-Square     Coeff Var      Root MSE     ltlc Mean

0.579712      10.70821      0.082276      0.768346
```

Source	DF	Type I SS	Mean Square	F Value	Pr > F
sex	1	0.13626303	0.13626303	20.13	0.0001
height	1	0.13451291	0.13451291	19.87	0.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
sex	1	0.00968023	0.00968023	1.43	0.2415
height	1	0.13451291	0.13451291	19.87	0.0001

Parameter		Estimate	Standard Error	t Value	Pr > |t|
Intercept		-.3278068826 B	0.26135206	-1.25	0.2198
sex	F	-.0421012632 B	0.03520676	-1.20	0.2415
sex	M	0.0000000000 B	.	.	.
height		0.0066723630	0.00149683	4.46	0.0001

Parameter		95% Confidence Limits	
Intercept		-.8623318537	0.2067180884
sex	F	-.1141071749	0.0299046484
sex	M	.	.
height		0.0036110089	0.0097337172

Note: The effect of gender has disappeared!!

In **this** example we have seen

- The observed difference in lung capacity between men and women can be explained by height difference

  However, there *may* still be a gender difference (women vs. men), estimated as $-0.0421 \pm 2 \times 0.0352 = (-0.1141, 0.0299)$, corresponding to the interval (0.77, 1.07) for ratios.

  If we would rather see it as men vs. women, we invert the figures to get the confidence interval (0.93,1.30) for ratios, i.e. there may be a 30% increased lung function for men.

It **may also occur**, that

- Apparently identical groups (e.g. blood pressure for men and women) may show up differences when we correct for inhomogeneities between groups
  (e.g. obesity)

We may conclude: It is **important** to remember all relevant covariates.

General statistical tool: **Multiple regression / General linear model**

Data:

n sets of observations, made on the same 'unit':

unit	$x_1 ....x_p$	y
1	$x_{11} ....x_{1p}$	y_1
2	$x_{21} ....x_{2p}$	y_2
3	$x_{31} ....x_{3p}$	y_3
.	. . . . .	.
n	$x_{n1} ....x_{np}$	y_n

The **linear regression model** with p explanatory variables (covariates) is written:

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon$$

Interpretation of regression coefficients β

Model $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_p X_{ip} + \epsilon$ where $\epsilon \sim N(0, \sigma^2)$

Consider two subjects:

A has covariate values $(X_1, X_2, \ldots, X_p)$

B has covariate values $(X_1 + 1, X_2, \ldots, X_p)$

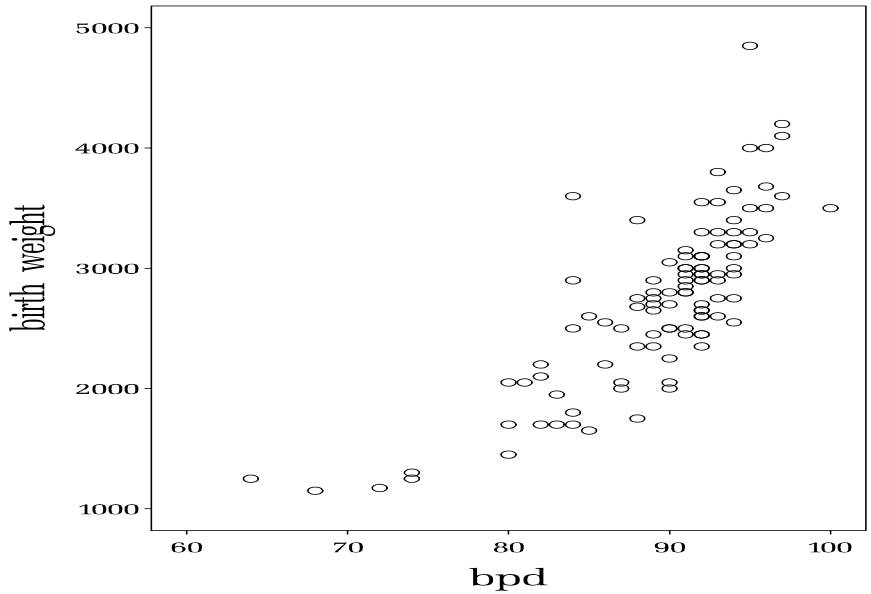
Expected difference in the response $(B - A)$

$\beta_0 + \beta_1(X_1 + 1) + \beta_2 X_2 + ... + \beta_p X_p -$
$[\beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p] = \beta_1$

β_1: is the effect of a one unit increase in X_1 *for fixed level of the other predictors*

Ultra sound scanning, immediately before birth (Secher et al.)

OBS	WEIGHT	BPD	AD
1	2350	88	92
2	2450	91	98
.	.	.	.
.	.	.	.
106	1173	72	73
107	2900	92	104



```
proc reg data=secher;
model lweight=lbpd lad / clb;
run;
```

Dependent Variable: lweight

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	14.95054	7.47527	314.93	<.0001
Error	104	2.46861	0.02374		
Corrected Total	106	17.41915			

Root MSE	0.15407	R-Square	0.8583
Dependent Mean	11.36775	Adj R-Sq	0.8556
Coeff Var	1.35530		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-8.45636	0.95457	-8.86	<.0001
lbpd	1	1.55194	0.22945	6.76	<.0001
lad	1	1.46666	0.14669	10.00	<.0001

Variable	DF	95% Confidence Limits	
Intercept	1	-10.34931	-6.56341
lbpd	1	1.09694	2.00695
lad	1	1.17577	1.75756

Interpretation of regression parameters

β_j: The effect of the j'th explanatory variable, **corrected** for the effect of the other explanatory variables –
i.e. when these are **kept fixed**

E.g: The effect of $\log_{10}(\texttt{bpd})$ corrected for the effect of $\log_{10}(\texttt{ad})$ is found to be $\hat{\beta}_1 = 1.552$

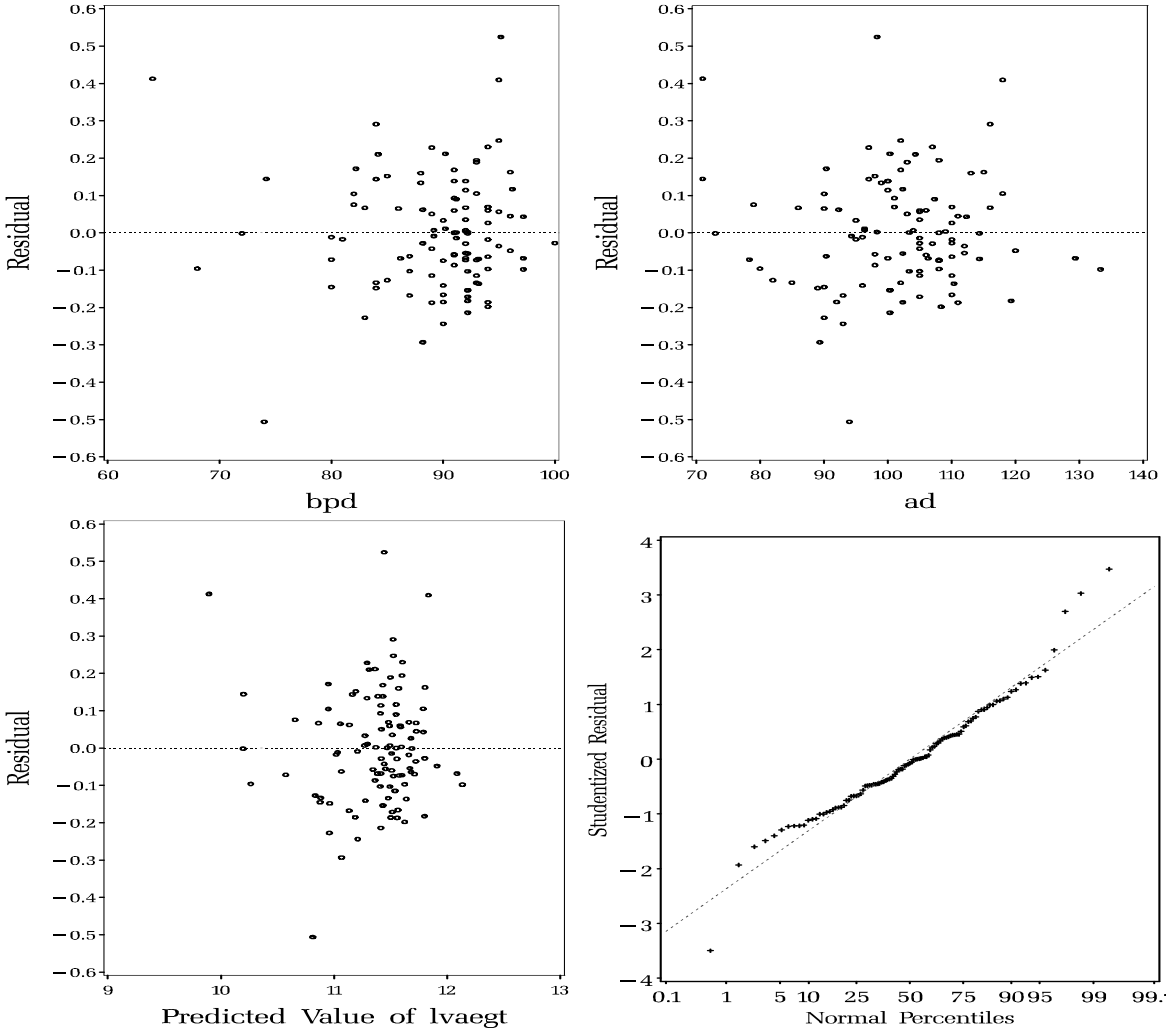
but in the marginal model **without** correction for $\log(\texttt{ad})$, we get: $\hat{\beta}_1^*$=3.332

The difference can be very **important**!

Group variables

Group variables can be directly handled in PROC GLM by choosing the group variable as a CLASS variable.

Residual plots



Ex. O'Neill et.al. (1983):
Lung function for 25 patients with cystic fibrosis.

Table 12.11 Data for 25 patients with cystic fibrosis (O'Neill *et al.*, 1983)

Sub	Age	Sex	Height	Weight	BMP	FEV_1	RV	FRC	TLC	PEmax
1	7	0	109	13.1	68	32	258	183	137	95
2	7	1	112	12.9	65	19	449	245	134	85
3	8	0	124	14.1	64	22	441	268	147	100
4	8	1	125	16.2	67	41	234	146	124	85
5	8	0	127	21.5	93	52	202	131	104	95
6	9	0	130	17.5	68	44	308	155	118	80
7	11	1	139	30.7	89	28	305	179	119	65
8	12	1	150	28.4	69	18	369	198	103	110
9	12	0	146	25.1	67	24	312	194	128	70
10	13	1	155	31.5	68	23	413	225	136	95
11	13	0	156	39.9	89	39	206	142	95	110
12	14	1	153	42.1	90	26	253	191	121	90
13	14	0	160	45.6	93	45	174	139	108	100
14	15	1	158	51.2	93	45	158	124	90	80
15	16	1	160	35.9	66	31	302	133	101	134
16	17	1	153	34.8	70	29	204	118	120	134
17	17	0	174	44.7	70	49	187	104	103	165
18	17	1	176	60.1	92	29	188	129	130	120
19	17	0	171	42.6	69	38	172	130	103	130
20	19	1	156	37.2	72	21	216	119	81	85
21	19	0	174	54.6	86	37	184	118	101	85
22	20	0	178	64.0	86	34	225	148	135	160
23	23	0	180	73.8	97	57	171	108	98	165
24	23	0	175	51.1	71	33	224	131	113	95
25	23	0	179	71.5	95	52	225	127	101	195

Which explanatory variables have a *marginal* effect on the outcome PE_{max}?

Table 12.12 Results of separately regressing PEmax on each explanatory variable

Explanatory variable	Regression coefficient	Standard error	t	P
Age	4.055	1.088	3.73	0.0011
Sex	−19.045	13.176	−1.45	0.16
Height	0.932	0.260	3.59	0.0016
Weight	1.187	0.301	3.94	0.0006
BMP	0.639	0.565	1.13	0.27
FEV_1	1.354	0.555	2.44	0.023
RV	−0.123	0.077	−1.59	0.12
FRC	−0.319	0.145	−2.20	0.038
TLC	−0.358	0.404	−0.89	0.38

Are these the variables to be included in the model?

Model with all covariates

```
proc reg data=secher;
  model pemax=age sex height weight bmp fev1 rv frc tlc;
run;
```

The REG Procedure
Dependent Variable: pemax

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	176.05821	225.89116	0.78	0.4479
age	1	-2.54196	4.80170	-0.53	0.6043
sex	1	-3.73678	15.45982	-0.24	0.8123
height	1	-0.44625	0.90335	-0.49	0.6285
weight	1	2.99282	2.00796	1.49	0.1568
bmp	1	-1.74494	1.15524	-1.51	0.1517
fev1	1	1.08070	1.08095	1.00	0.3333
rv	1	0.19697	0.19621	1.00	0.3314
frc	1	-0.30843	0.49239	-0.63	0.5405
tlc	1	0.18860	0.49974	0.38	0.7112

Model selection

- **Forward selection:** Start with no covariates. In every step, add the most significant variable

```
proc reg data=secher;
  model pemax=age sex height weight bmp fev1 rv frc tlc
              / selection=forward;
run;
```

Final model: `weight bmp fev1`

- **Backward elimination**
  Start with all covariates. At each step, omit the least significant

```
proc reg data=secher;
  model pemax=age sex height weight bmp fev1 rv frc tlc
              / selection=backward;
run;
```

Final model: `weight bmp fev1`

But:

If `weight` had been transformed with the logarithm from the start, we would have had the final model `age fev1`

Selection procedures

- `backward`
- `forward`
- ...

A 'best' method has not been identified, but backward elimination is generally recommended over forward selection.

WARNING: The output from the selected model does not take the model selection uncertainty into account. The output (regression coefficients and p-values) is identical to what would have been obtained had we fitted the final model with out doing any model selection. The importance of selected covariates is over-estimated.

Exercise: General linear models

We take another look at Juul's data.

1. Get the data into SAS using a libname statement.
2. Create a new data set including only individuals above 25 years.
3. Use PROC GPLOT to plot the relationship between age and $\sqrt{\text{SIGF-I}}$. Make separate regression lines for men and women.
4. Do a regression analysis to explore whether the slopes (age - $\sqrt{\text{SIGF-I}}$) are the same in men and women. Give an estimate for the difference in slopes, with 95% confidence interval.
5. Expand the regression model by including height. Delete in-significant covariates.

Use of SAS - December, 2010

7. Categorical data

Karl Bang Christensen* and Esben Budtz-Jørgensen

Department of Biostatistics, University of Copenhagen.

*kach@biostat.ku.dk, tel: 35327491

Analysis of categorical data

Tables (frequency tables)
Rate ratios
Odds ratios
Logistic regression

Table

	Outcome		
Exposure	Yes	No	Total
Yes	a	b	n_1
No	c	d	n_2
Total	$a+c$	$b+d$	n

Hypothesis H_0: the probability of having the outcome is the same in the two exposure groups.

The Guinea-Bissau data set

Available on the T-drive we have a SAS data set called `bissau.sas7bdat`. Data comes from rural Guinea-Bissau, West-Africa: 5273 children visited when being less than 7 months of age and followed for approximately six months.

Registration of vaccination status, weight, etc at visit and deaths registered during follow-up.

In SAS: PROC FREQ

```
/* One-way table */
proc freq data=afrika.bissau;
  tables dead;
run;

/* Two-way table */
proc freq data=afrika.bissau;
  tables bcg*dead;
run;
```

Two-way table: Risk of Dying and BCG.

bcg	dead		
Frequency / Percent / Row Pct / Col Pct	1	2	Total
1	124	3176	3300
	2.35	60.23	62.58
	3.76	96.24	
	56.11	62.87	
2	97	1876	1973
	1.84	35.58	37.42
	4.92	95.08	
	43.89	37.13	
Total	221	5052	5273
	4.19	95.81	100.00

Risk of Dying and BCG - only the information we want

```
proc freq data=afrika.bissau;
  tables bcg*dead / nocol nopercent;
run;
```

```
bcg       dead

Frequency|
Row Pct  |       1|       2|  Total
---------+--------+--------+
       1 |    124 |   3176 |   3300
         |   3.76 |  96.24 |
---------+--------+--------+
       2 |     97 |   1876 |   1973
         |   4.92 |  95.08 |
---------+--------+--------+
Total         221     5052     5273
```

The risk of dying in the two BCG groups: 3.76% with BCG and 4.92% without BCG.

We want to know if these probabilities are significantly different.

Therefore we test the null hypothesis H_0: the probability of dying is the same in the two groups.

Table

	Outcome		
Exposure	Yes	No	Total
Yes	a	b	n_1
No	c	d	n_2
Total	$a+c$	$b+d$	n

Hypothesis H_0: the probability of having the outcome is the same in the two exposure groups.

probability of under H_0 is $p = \frac{a+c}{n}$.

Chi-square test

Under H_0 expected numbers in the four cells are:

	Outcome		
Exposure	Yes	No	Total
Yes	$E(a) = p \times n_1$	$E(b) = (1-p) \times n_1$	n_1
No	$E(c) = p \times n_2$	$E(d) = (1-p) \times n_2$	n_2
Total	$a + c$	$b + d$	n

Chi-square test for testing H_0 (observed - expected):

$$X^2 = \frac{[a - E(a)]^2}{E(a)} + \frac{[b - E(b)]^2}{E(b)} + \frac{[c - E(c)]^2}{E(c)} + \frac{[d - E(d)]^2}{E(d)}$$

H_0 is rejected if p-value < 0.05 which corresponds to $X^2 > 3.84$.

Risk of Dying and BCG - expected numbers

```
proc freq data=afrika.bissau;
  tables bcg*dead / expected chisq nocol nopercent;
run;
```

Table of bcg by dead

bcg	dead		
Frequency|			
Expected |			
Row Pct |	1	2	Total
1	124	3176	3300
	138.31	3161.7	
	3.76	96.24	
2	97	1876	1973
	82.692	1890.3	
	4.92	95.08	
Total	221	5052	5273

Risk of Dying and BCG - Chi-square test

Statistics for Table of bcg by dead

Statistic	DF	Value	Prob
Chi-Square	1	4.1291	0.0422
Likelihood Ratio Chi-Square	1	4.0516	0.0441
Continuity Adj. Chi-Square	1	3.8456	0.0499
Mantel-Haenszel Chi-Square	1	4.1283	0.0422
Phi Coefficient		-0.0280	
Contingency Coefficient		0.0280	
Cramer's V		-0.0280	

The risk of dying in the two BCG groups: 3.76% and 4.92%.

We see from the Chi-square test that the probability of dying is differs significantly between the groups.

How can we quantify this?

The risk difference 4.92-3.76 = 1.16 is not always a good idea

Risk Ratio

	Outcome		
Exposure	Yes	No	Total
Yes	a	b	n_1
No	c	d	n_2
Total	$a+c$	$b+d$	n

Risk ratio:

$$\text{RR} = \frac{\text{probability of outcome among exposed}}{\text{probability of outcome among not-exposed}} = \frac{a/n_1}{c/n_2}.$$

The H_0 corresponds to RR $= 1$.

Odds

	Outcome		
Exposure	Yes	No	Total
Yes	a	b	n_1
No	c	d	n_2

Let $p = a/n_1$ be the probability of outcome among exposed. Odds can then be defined as

$$\text{odds} = \frac{p}{1-p} = \frac{a/n_1}{1-a/n_1} = \frac{a/n_1}{b/n_1} = \frac{a}{b}$$

does not contain any other information than the probability. If the probability is higher odds are higher and vice versa.

Odds ratio

	Outcome		
Exposure	Yes	No	Total
Yes	a	b	n_1
No	c	d	n_2
Total	$a+c$	$b+d$	n

Odds ratio:

$$\text{OR} = \frac{\text{odds of outcome among exposed}}{\text{odds of outcome among not-exposed}} = \frac{a/b}{c/d} = \frac{a \times d}{b \times c}$$

The H_0 corresponds to OR $= 1$.

RR and OR in PROC FREQ

```
proc freq data=afrika.bissau;
  table bcg*dead / RELRISK nocol nopercent;
run;
```

Estimates of the Relative Risk (Row1/Row2)

Type of Study	Value	95% Confidence Limits	
Case-Control (Odds Ratio)	0.7551	0.5754	0.9909
Cohort (Col1 Risk)	0.7643	0.5895	0.9910
Cohort (Col2 Risk)	1.0122	1.0000	1.0245

RR and OR in PROC FREQ II

It is important how the two variables in the `table` statement are coded. If we recode them

```
data hope;
    set afrika.bissau;
    deadny=2-dead;
run;

proc freq data=hope;
    table bcg*deadny / relrisk nocol nopercent;
run;
```

We get something else.

bcg	deadny		
Frequency|			
Row Pct	0	1	Total
1	3176	124	3300
	96.24	3.76	
2	1876	97	1973
	95.08	4.92	
Total	5052	221	5273

Statistics for Table of bcg by deadny

Estimates of the Relative Risk (Row1/Row2)

Type of Study	Value	95% Confidence Limits	
Case-Control (Odds Ratio)	1.3243	1.0092	1.7378
Cohort (Col1 Risk)	1.0122	1.0000	1.0245
Cohort (Col2 Risk)	0.7643	0.5895	0.9910

R x C tables

We can also compare more than two groups

```
proc freq data=afrika.bissau;
  table ethnic*dead/norow nocol nopercent chisq;
run;
```

The null hypothesis

H_0: the risk of dying is the same in the five groups

ethnic	dead		
Frequency	1	2	Total
Balanta	37	788	825
Fula	52	1370	1422
Mandinga	49	1113	1162
Other	23	724	747
Pepel	60	1057	1117
Total	221	5052	5273

Null hypothesis H_0: the risk of dying is the same in the five groups

Statistics for Table of ethnic by dead

Statistic	DF	Value	Prob
Chi-Square	4	7.3670	0.1177
Likelihood Ratio Chi-Square	4	7.3268	0.1196
Mantel-Haenszel Chi-Square	1	1.0857	0.2974
Phi Coefficient		0.0374	
Contingency Coefficient		0.0374	
Cramer's V		0.0374	

PROC FREQ Exercise Using the bissau data:

1. Do DTP-vaccinated children (variable `dtp`) die more often than DTP-unvaccinated children?

2. Calculate the odds ratio (OR) and corresponding 95% confidence interval.

3. The variable `region` indicates the rural region of the children. Is mortality associated with region?

Logistic regression: PROC LOGISTIC

Logistic regression is like a linear regression, but here the outcome is discrete with two levels (yes/no, died/survived, ill/well).

Look again at the 2 x 2 table

	Outcome		
Exposure	Yes	No	Total
Yes	a	b	n_1
No	c	d	n_2

$$\text{odds} = \frac{p}{1-p} = \frac{a/n_1}{1-a/n_1} = \frac{a/n_1}{b/n_1} = \frac{a}{b}$$

Logistic regression for 2 x 2 table

What is modeled in a logistic regression is the natural logarithm of the odds of outcome:

$$\text{ln(odds)} = \text{ln}\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X,$$

where X is the exposure covariate. We call ln(odds) the log-odds. Assume that the exposure is coded like

$$X = \begin{cases} 1 & \text{Exposed} \\ 0 & \text{Non-exposed} \end{cases}$$

The log-odds of outcome among exposed ($X = 1$) is

$$\ln\left(\frac{p_1}{1-p_1}\right) = \beta_0 + \beta_1 \times 1 = \beta_0 + \beta_1.$$

The log-odds of outcome among non-exposed ($X = 0$) is

$$\ln\left(\frac{p_0}{1-p_0}\right) = \beta_0 + \beta_1 \times 0 = \beta_0.$$

The difference in log-odds between exposed and non-exposed is

$$\ln\left(\frac{p_1}{1-p_1}\right) - \ln\left(\frac{p_0}{1-p_0}\right) = (\beta_0 + \beta_1) - \beta_0 = \beta_1$$

Using the rule of logarithms

$$\ln(a) - \ln(b) = \ln(\frac{a}{b})$$

we get

$$\ln\left(\frac{p_1/(1-p_1)}{p_0/(1-p_0)}\right) = \beta_1$$

and this means that the odds ratio between exposed and non-exposed is

$$\text{OR} = \exp(\beta_1).$$

Estimation of the regression coefficients is done using maximum likelihood.

PROC LOGISTIC

```
proc logistic data=afrika.bissau;
  class bcg / param=ref;
  model dead(event="1")=bcg;
run;
```

Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	-2.9621	0.1041	809.3011	<.0001
bcg	1	1	-0.2810	0.1386	4.1074	0.0427

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
bcg 1 vs 2	0.755	0.575	0.991

REMEMBER the option `param=ref`

Logistic regression

For the case of a 2 x 2 table the logistic regression model is just a more complicated way of getting the OR with a general way of writing the model

$$\text{ln(odds)} = \text{ln}\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X,$$

the exposure covariate X was coded

$$X = \begin{cases} 1 & \text{Exposed} \\ 0 & \text{Non-exposed} \end{cases}$$

this general framework also works for continuous X (e.g. age).

Multiple logistic regression

$$\text{ln(odds)} = \text{ln}\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots,$$

The interpretation is still that $\exp(\beta_1)$ is an odds ratios, but now adjusted for the covariates $X_2, X_3, \cdots$.

Same idea as in multiple linear regression.

The response or outcome is discrete with two categories, but covariates $(X_1, X_2, X_3, \cdots)$ do not need to be categorical, they can also be continuous.

In SAS one uses the CLASS statement to indicate categorical variables. Variables in a MODEL statement not listed in the CLASS statement are assumed to be continuous.

Multiple logistic regression: PROC LOGISTIC

```
proc logistic data=afrika.bissau;
  class bcg / param=ref;
  model dead(event="1")=bcg agemm;
run;
```

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
bcg	1	5.4366	0.0197
agemm	1	1.5307	0.2160

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
bcg 1 vs 2	0.708	0.530	0.946
agemm	1.050	0.972	1.134

Interpretation: For each increase of 1 in agemm the odds increases with 1.050.

Multiple logistic regression: PROC LOGISTIC

The variable agemm is now used as a CLASS variable:

```
proc logistic data=afrika.bissau;
  class bcg agemm / param=ref;
  model dead(event="1")=bcg agemm;
run;
```

agemm has 7 classes: 0 to 6. SAS automatically generates seven indicator functions for each class and includes six of these in the regression model. The class not included is the reference group (per default SAS uses the highest class).

The test in TYPE 3 for agemm is a test for the hypothesis of equal risk of dying in the 7 classes. This test does not change depend on the choice of reference group.

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
bcg	1	5.2393	0.0221
agemm	6	7.3938	0.2860

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
bcg 1 vs 2	0.710	0.529	0.952
agemm 0 vs 6	1.061	0.524	2.147
agemm 1 vs 6	1.198	0.602	2.384
agemm 2 vs 6	0.825	0.403	1.687
agemm 3 vs 6	1.310	0.658	2.608
agemm 4 vs 6	1.498	0.756	2.971
agemm 5 vs 6	1.436	0.716	2.879

Change of reference group: REF=""

The variable agemm is again used as a CLASS variable but now choosing agegroup 4 as reference:

```
proc logistic data=afrika.bissau;
  class bcg agemm(ref="4") / param=ref;
  model dead(event="1")=bcg agemm;
run;
```

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
bcg 1 vs 2	0.710	0.529	0.952
agemm 0 vs 4	0.708	0.435	1.150
agemm 1 vs 4	0.799	0.502	1.271
agemm 2 vs 4	0.551	0.332	0.912
agemm 3 vs 4	0.874	0.548	1.395
agemm 5 vs 4	0.958	0.594	1.546
agemm 6 vs 4	0.667	0.337	1.323

Exercise: PROC LOGISTIC

Using the Bissau data:

1. Make a logistic regression where outcome is `dead` and exposure is `dtp`. Interpret the results and compare with the results from the exercise using `proc freq`.

2. Now control for `bcg` in the logistic regression from 1 above. What happened with the odds ratio for `dtp`?

3. Add variables `agemm` and `region` to the model as class variables. Let `region=7` be the reference group for variable `region`. Did inclusion of these variables change interpretation of effect `dtp`?

Use of SAS - December, 2010

8. Reading data into SAS

Karl Bang Christensen* and Esben Budtz-Jørgensen

Department of Biostatistics, University of Copenhagen.

*kach@biostat.ku.dk, tel: 35327491

Reading data into SAS

from text file
data lines directly in SAS program
import from, e.g., Excel

Importantly: In SPSS you can save your data as SAS data sets

Special considerations

character variables
data seperation
missing values

Reading in some "nice" data

The data file `bissau.sas7bdat` is a SAS data set and it is easy to work with

1. We copy the SAS-data set `T:\bissau.sas7bdat` to a directory on our p-drive.
2. We link this directory to a SAS library called, e.g. , `afrika` using a `libname` statement.
3. We make a SAS program that makes contains the `libname` statement. Restarting SAS, we have to submit the `libname` statement again

Lung function data: 25 patients with cystic fibrosis.

Table 12.11 Data for 25 patients with cystic fibrosis (O'Neill *et al.*, 1983)

Sub	Age	Sex	Height	Weight	BMP	FEV_1	RV	FRC	TLC	PEmax
1	7	0	109	13.1	68	32	258	183	137	95
2	7	1	112	12.9	65	19	449	245	134	85
3	8	0	124	14.1	64	22	441	268	147	100
4	8	1	125	16.2	67	41	234	146	124	85
5	8	0	127	21.5	93	52	202	131	104	95
6	9	0	130	17.5	68	44	308	155	118	80
7	11	1	139	30.7	89	28	305	179	119	65
8	12	1	150	28.4	69	18	369	198	103	110
9	12	0	146	25.1	67	24	312	194	128	70
10	13	1	155	31.5	68	23	413	225	136	95
11	13	0	156	39.9	89	39	206	142	95	110
12	14	1	153	42.1	90	26	253	191	121	90
13	14	0	160	45.6	93	45	174	139	108	100
14	15	1	158	51.2	93	45	158	124	90	80
15	16	1	160	35.9	66	31	302	133	101	134
16	17	1	153	34.8	70	29	204	118	120	134
17	17	0	174	44.7	70	49	187	104	103	165
18	17	1	176	60.1	92	29	188	129	130	120
19	17	0	171	42.6	69	38	172	130	103	130
20	19	1	156	37.2	72	21	216	119	81	85
21	19	0	174	54.6	86	37	184	118	101	85
22	20	0	178	64.0	86	34	225	148	135	160
23	23	0	180	73.8	97	57	171	108	98	165
24	23	0	175	51.1	71	33	224	131	113	95
25	23	0	179	71.5	95	52	225	127	101	195

Some of these data are in the text file `pemax.txt` on the T-drive

```
age sex height weight fev1 pemax
  7 1 109 13.1 32  95
  7 2 112 12.9 19  85
  8 1 124 14.1 22 100
  8 2 125 16.2 41  85
  8 1 127 21.5 52  95
  9 1 130 17.5 44  80

  : :   :    :  :   :

 19 2 156 37.2 21  85
 19 1 174 54.6 37  85
 20 1 178 64.0 34 160
 23 1 180 73.8 57 165
 23 1 175 51.1 33  95
 23 1 179 71.5 52 195
```

Reading in data from a text file

In the program editor, we write the program lines:

```
data sasuser.pemax;
  infile 'T:\pemax.txt' firstobs=2;
  input age sex height weight fev1 pemax;
run;
```

Note: the option `firstobs=2` tells SAS that line 2 is the first line that contains data. Log file:

```
NOTE: 25 records were read from the infile 'pemax.txt'.
      The minimum record length was 21.
      The maximum record length was 21.
NOTE: The data set SASUSER.PEMAX has 25 observations and 6
variables. NOTE: DATA statement used:
      real time           0.11 seconds
      cpu time            0.01 seconds
```

We now have the permanent SAS data set 'pemax' in the 'sasuser' library.

Data lines directly in program

```
data sasuser.pemax;
 input age sex height weight fev1 pemax;
 datalines;
  7 1 109 13.1 32  95
  7 2 112 12.9 19  85
  8 1 124 14.1 22 100

 23 1 179 71.5 52 195
 ;
run;
```

Reading in character variables

```
data sasuser.pemax;
 input age sex $ height weight fev1 pemax;
 datalines;
  7 male   109 13.1 32  95
  7 female 112 12.9 19  85
  8 male   124 14.1 22 100

 23 male   179 71.5 52 195
 ;
run;
```

Include '$' after each character variable.

Semicolon separated data

Until now, data have been nicely separated by blanks, but what if it looks a bit different.....

```
age;sex;height;weight;fev1;pemax
7;male,109;13.1;32;95
7;female;112;12.9;19;85
8;male;124;14.1;22;100
8;female;125;16.2,41;85
.....
.....
```

We now have to modify the SAS program and specify a list of possible delimiters

```
data sasuser.pemax;
    infile 'T:\pemax2.txt' firstobs=2 dlm=';,';
    input age sex $ height weight fev1 pemax;
run;
```

Using 'dlm' both comma and semicolon are regarded as delimiters. Period is not a good delimiter. Why??

Formatted input

Sometimes the values are not separated at all.

This is often useful for many binary observations, e.g. questionnaire data.

```
 7M10913.132 95
 7F11212.919 85
 8M12414.122100

23M17971.552195
```

In order to read data we have to specify where to find the data for each variable: In which column.

SAS code:

```
data sasuser.pemax;
    infile 'T:\pemax3.txt';
    input age 1-2 sex $ 3 height 4-6 weight 7-10 fev1 11-12 pemax 13-15;
run;
```

Missing values

Should be coded using '.' (period)

When looking at data-files take care if you see 9,-9, 99, 999 etc.

example:

```
data sasuser.pemax;
    input age sex height weight fev1 pemax;
    datalines;
    7 1 . 13.1 32  95
    7 2 112 12.9 19  85
    8 . 124 14.1 22 100
    .....

    .....
    ;
run;
```

Exercise: Reading in some 'ugly' data.

In the file `'orig_juul.txt'` (P-drive, juul directory) we have the original data from Anders Juul's investigation of growth hormone. Data are comma separated and appears in the following order:

```
age bmi genital height hsds hv igfbp3 mammae menarche pubestan sex
sigf1 tanner testvol weight
```

1. Read in the data into SAS
2. Check, that you have 1340 observations and 15 variables
3. We do not want to use the variables `hsds, hv, pubestan, mammae` and `genital`. Omit these from the data set.
4. Compute summary statistics: mean, median, number of missing observations, minimum and maximum.
5. Are there any missing values?
6. How are they represented?
7. Are there any strange values, which might actually be missing values? If so, make them into proper missing values, and calculate the summary statistics once more. Compare to the previous results.

Files from external programs

In general when you have files from Excel, SPSS or other programs we recommend the program *StatTransfer*, which can be used for converting almost any data-file to a SAS data set.

Alternatively, most programs will allow you to print data in a text file. This file can then be read into SAS using the previosly described methods.

Importing Excel sheets

A data set with information about adverse events from a clinical trial is in `ae.xls` on the P-drive

```
proc import out=work.adverse datafile= "P:\ae.XLS" dbms=excel replace;
     getnames=yes;
run;
```

Files from external programs

Can often be handled using 'Import Wizard'

1. Select File $\rightarrow$ Import Data
2. Select type of file to import
3. Specify where to put the generated sas-dataset (e.g. WORK)
4. Save automatically generated SAS code (PROC IMPORT)