

# Regression and graphics

Use of SAS  
March 2009

# 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 *nonparametric* 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:

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

- 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<br>0.0008 |
| obese | 0.32614<br>0.0008 | 1.00000           |

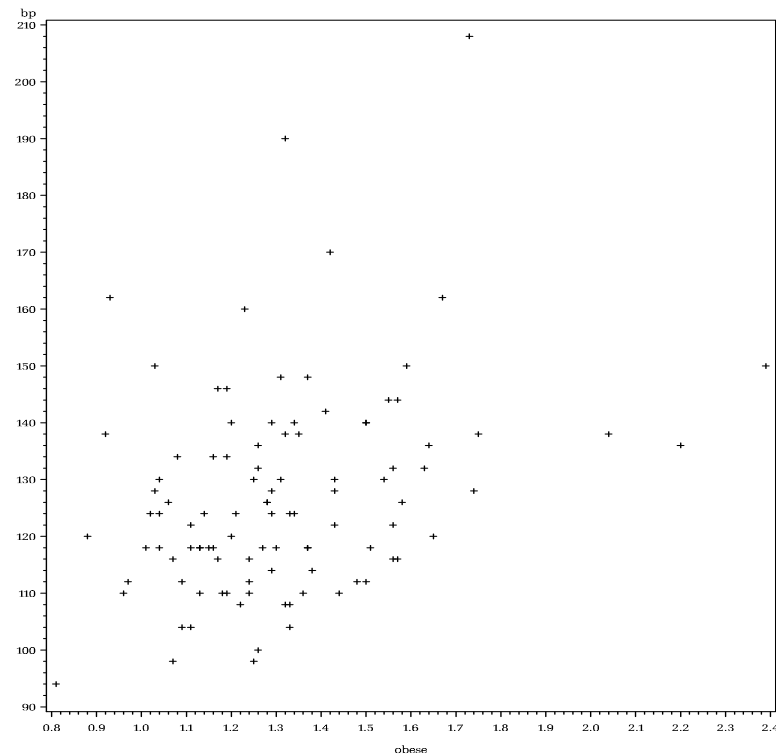
Spearman Correlation Coefficients, N = 102  
Prob > |r| under H0: Rho=0

|       | bp                | obese             |
|-------|-------------------|-------------------|
| bp    | 1.00000           | 0.30363<br>0.0019 |
| obese | 0.30363<br>0.0019 | 1.00000           |

# Scatter plot

In raw form:

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



This plot can be improved a lot....

# Linear regression

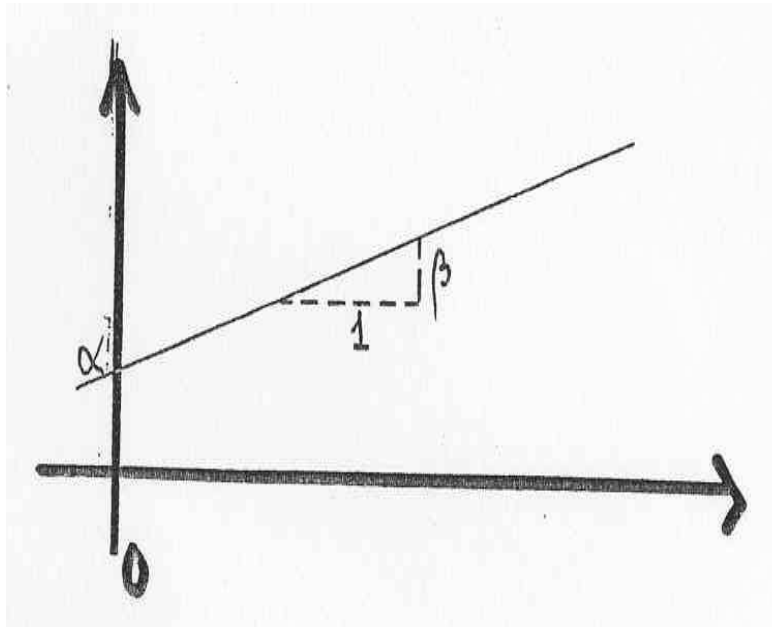
- **Y**: Response variable, outcome variable, dependent variable (here bp)
- **X**: Explanatory variable, independent variable, covariate (here obese)

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

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



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

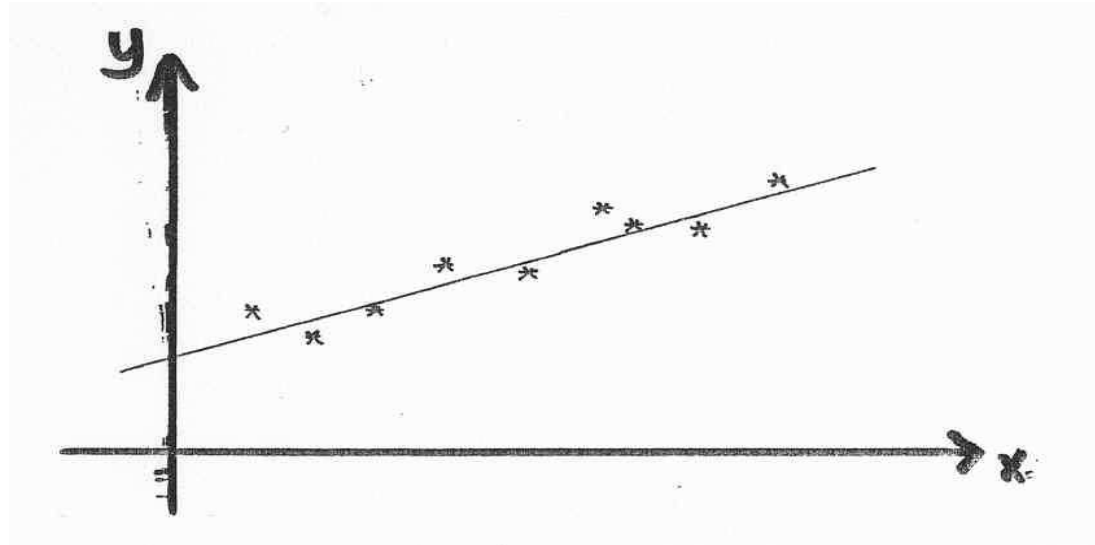


Interpretation:

- $\alpha$ : intercept, (intersection with  $Y$ -axis)  
The blood pressure for an individual with obesity 0.  
Often an illegal extrapolation.
- $\beta$ : 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  $\alpha$  and  $\beta$ , to minimize

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

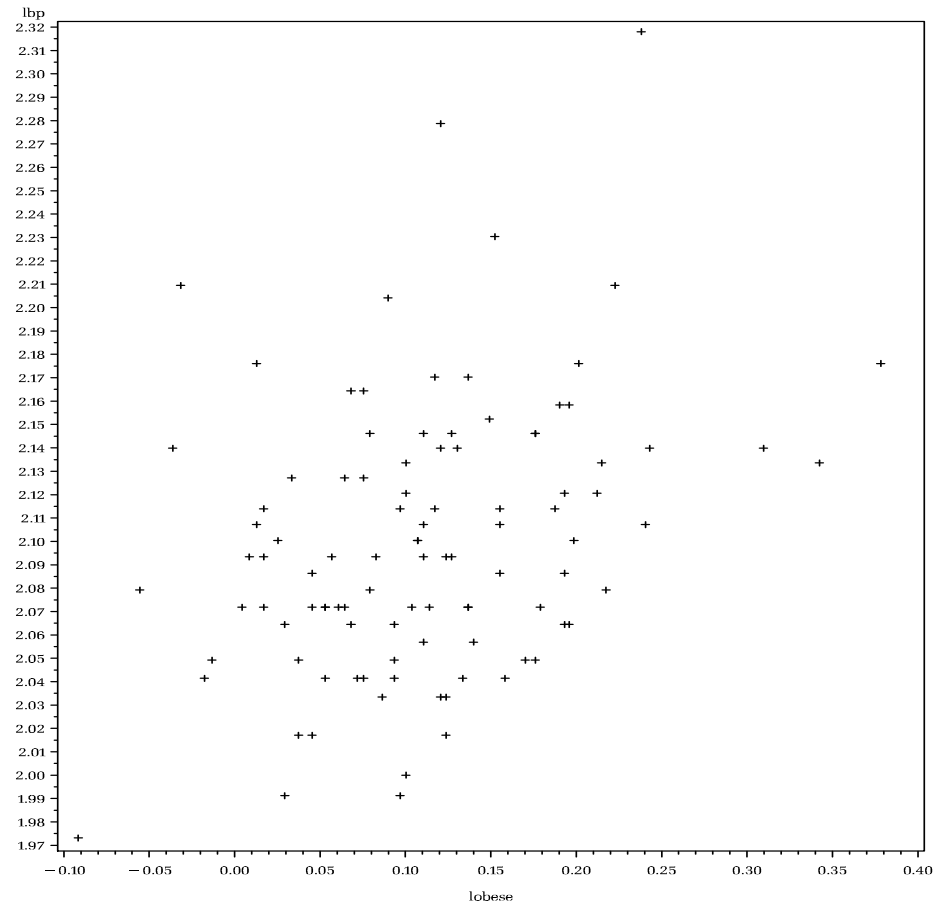
## Assumptions in linear regression

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

The last two assumptions are not quite fulfilled 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

| Source  | DF  | Sum of Squares | Mean 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

| Variable  | DF | Parameter Estimate | Standard Error | T for H0:<br>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**

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

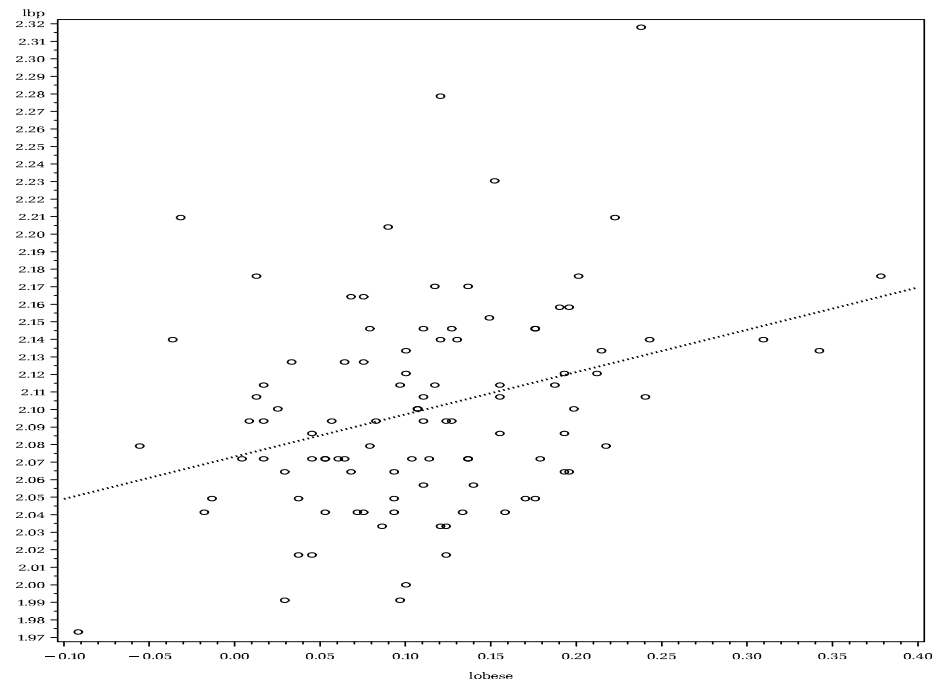
Thus, an increase of 0.241 in **lbp** corresponds to a 74% increase in **bp**

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

# Add a regression line to the plot

```
proc gplot data=sasuser.bp;  
  plot lbp*lobese;  
  symbol1 v=circle i=r1 l=33;  
run;
```

i=r1 gives the regression line - l=33 dotted line



## The variance around the regression line

$\sigma^2$  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:

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

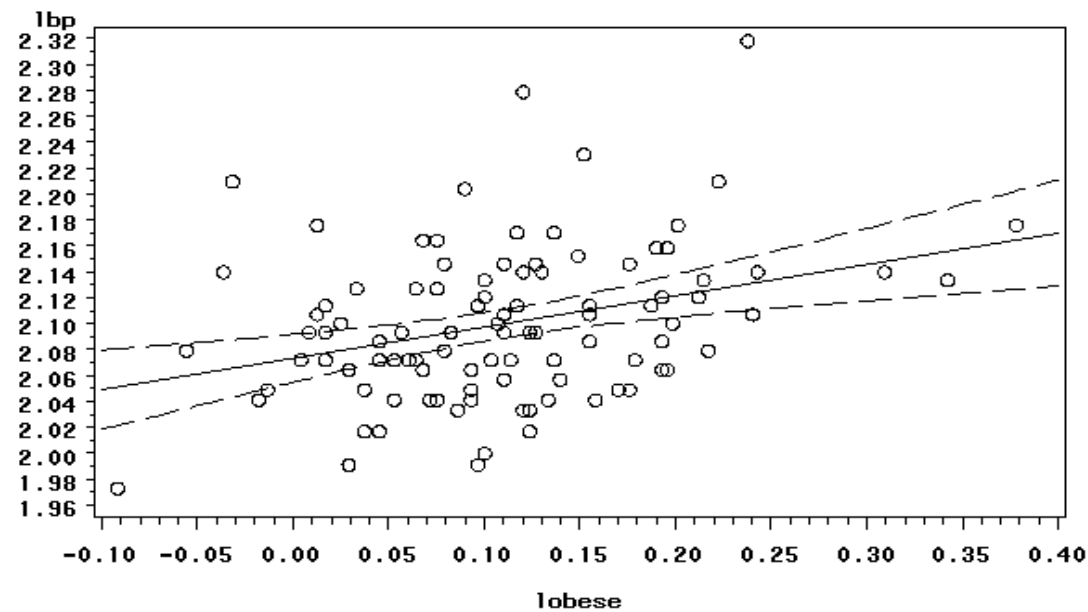
use the option `clb` in SAS:

```
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:

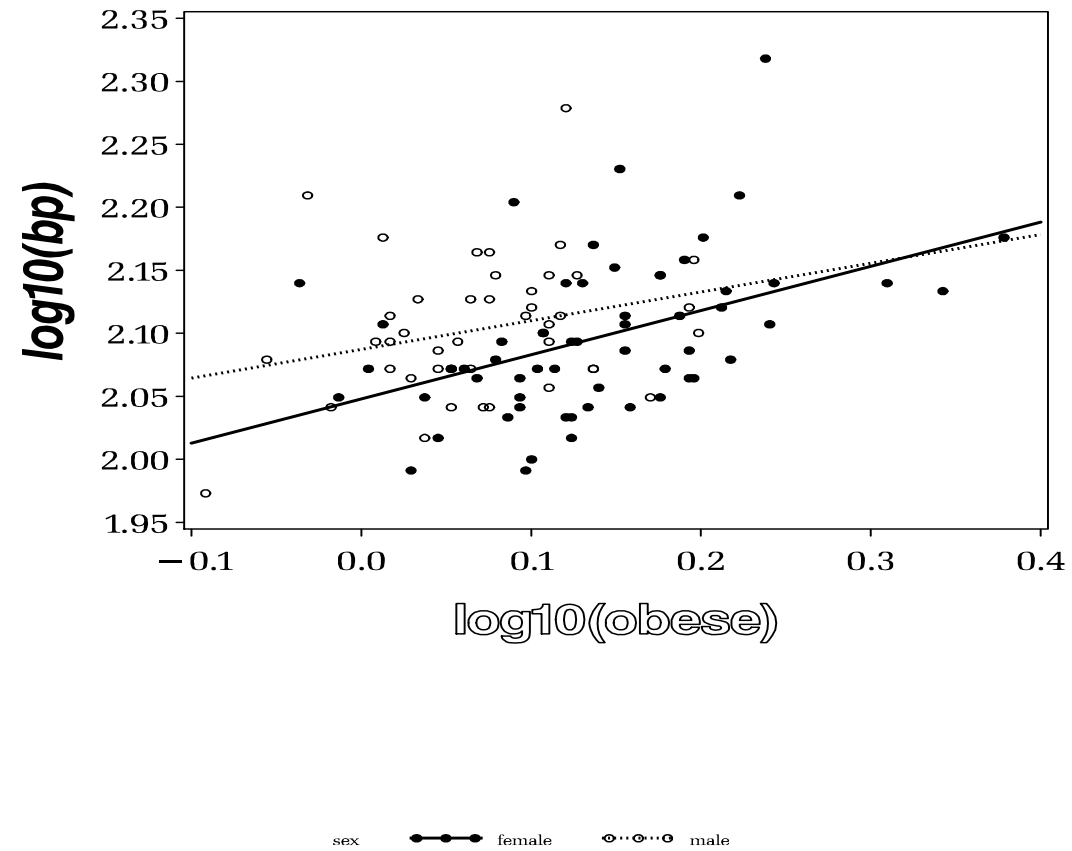
```
proc gplot;  
  plot bp*obese;  
run;
```

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=r1 c=BLACK l=1 w=2;  
symbol2 v=circle i=r1 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'

*obesity and blood pressure*



## 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/r1/r1cli95/r1clm95
- `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)   | ◻           | +      | ⊕           |
| X        | ×           | " (double quote) | ♠           | >      | ♂           |
| STAR     | *           | # (pound sign)   | ♥           | .      | ♀           |
| SQUARE   | ◻           | \$ (dollar sign) | ♦           | <      | ℎ           |
| DIAMOND  | ◊           | % (percent)      | ♣           | ,      | ⊙           |
| TRIANGLE | △           | & (ampersand)    | ♣           | /      | ψ           |
| HASH     | #           | ' (single quote) | ♣           | ?      | ℙ           |
| Y        | Y           | = (equals)       | ☆           | (      | ℳ           |
| Z        | Z           | — (hyphen)       | ⊙           | )      | ℴ           |
| PAW      | ⋯           | @ (at)           | ♀           | :      | *           |
| 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

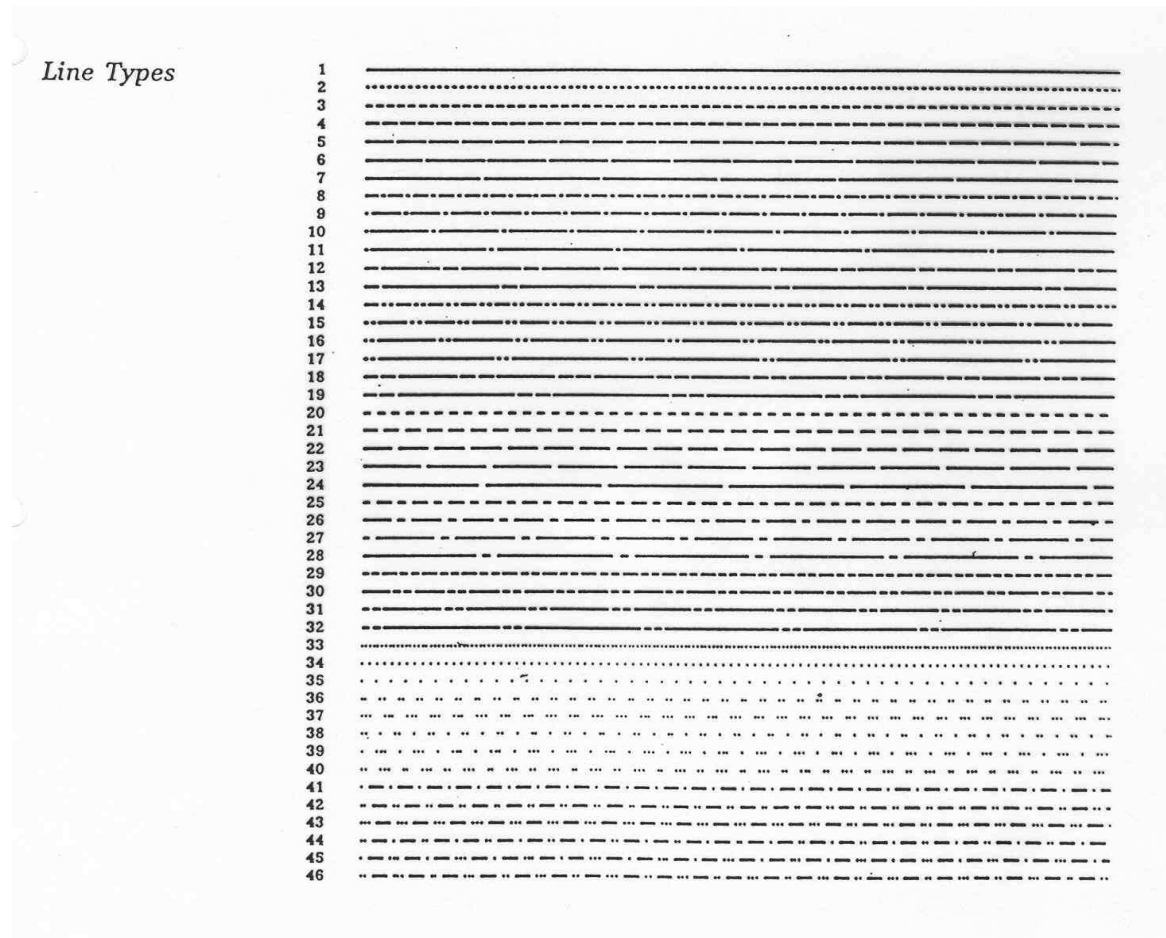
| If your data have . . .               | Then you might choose one of these methods . . . | And specify INTERPOL=    |
|---------------------------------------|--|--------------------------|
| one Y value for each X value          | join   | JOIN                     |
|                                       | <u>fitting a regression line</u>                 | 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 specified.
- `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     | <i>A B C a b c 1 2 3</i>        |              |
| Century               |           |                                 |              |
| Bold                  | CENTB     | <b>A B C a b c 1 2 3</b>        | CENTBU       |
| Bold Empty            | CENTBE    | A B C a b c 1 2 3               |              |
| Bold Italic           | CENTBI    | <b><i>A B C a b c 1 2 3</i></b> | CENTBIU      |
| Bold Italic Empty     | CENTBIE   | <i>A B C a b c 1 2 3</i>        |              |
| Expanded              | CENTX     | <b>A B C a b c 1 2 3</b>        | CENTXU       |
| Expanded Empty        | CENTXE    | A B C a b c 1 2 3               |              |
| Expanded Italic       | CENTXI    | <b><i>A B C a b c 1 2 3</i></b> | CENTXIU      |
| Expanded Italic Empty | CENTXIE   | <i>A B C a b c 1 2 3</i>        |              |
| German                | GERMAN    | Œ Ø € a b c 1 2 3               | GERMANU      |
| German Italic         | GITALIC   | Œ Ø € 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   | <b><i>A B C a b c 1 2 3</i></b> | TITALICU     |
| Serif Italic          | ITALIC    | <i>A B C a b c 1 2 3</i>        | ITALICU      |
| Old English           | OLDENG    | Æ Æ Œ a b c 1 2 3               | OLDENGU      |
| Script                | SCRIPT    | <i>A B C a b c 1 2 3</i>        |              |
| Cscript               | • CSCRIPT | <i>A B C a b c 1 2 3</i>        |              |
| Simulate              | SIMULATE  | A B C a b c 1 2 3               | SIMULATE     |
| Swiss                 | SWISS     | <b>A B C a b c 1 2 3</b>        | SWISSU       |
| Empty                 | SWISSE    | A B C a b c 1 2 3               |              |
| Bold                  | SWISSB    | <b>A B C a b c 1 2 3</b>        | SWISSBU      |
| Bold Empty            | • SWISSBE | A B C a b c 1 2 3               |              |
| Bold Italic           | • SWISSBI | <b><i>A B C a b c 1 2 3</i></b> | 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

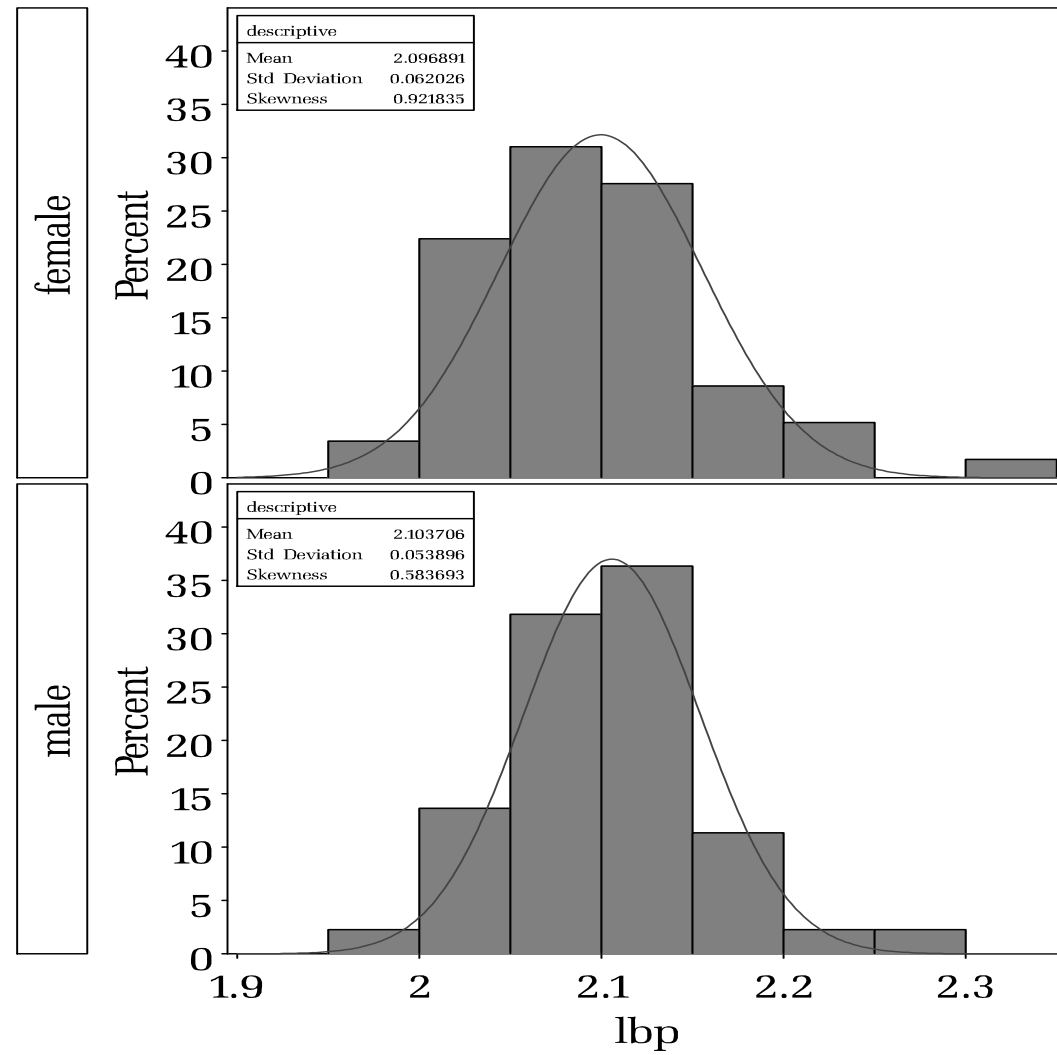
class sex: histogram for both values of sex

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;
```

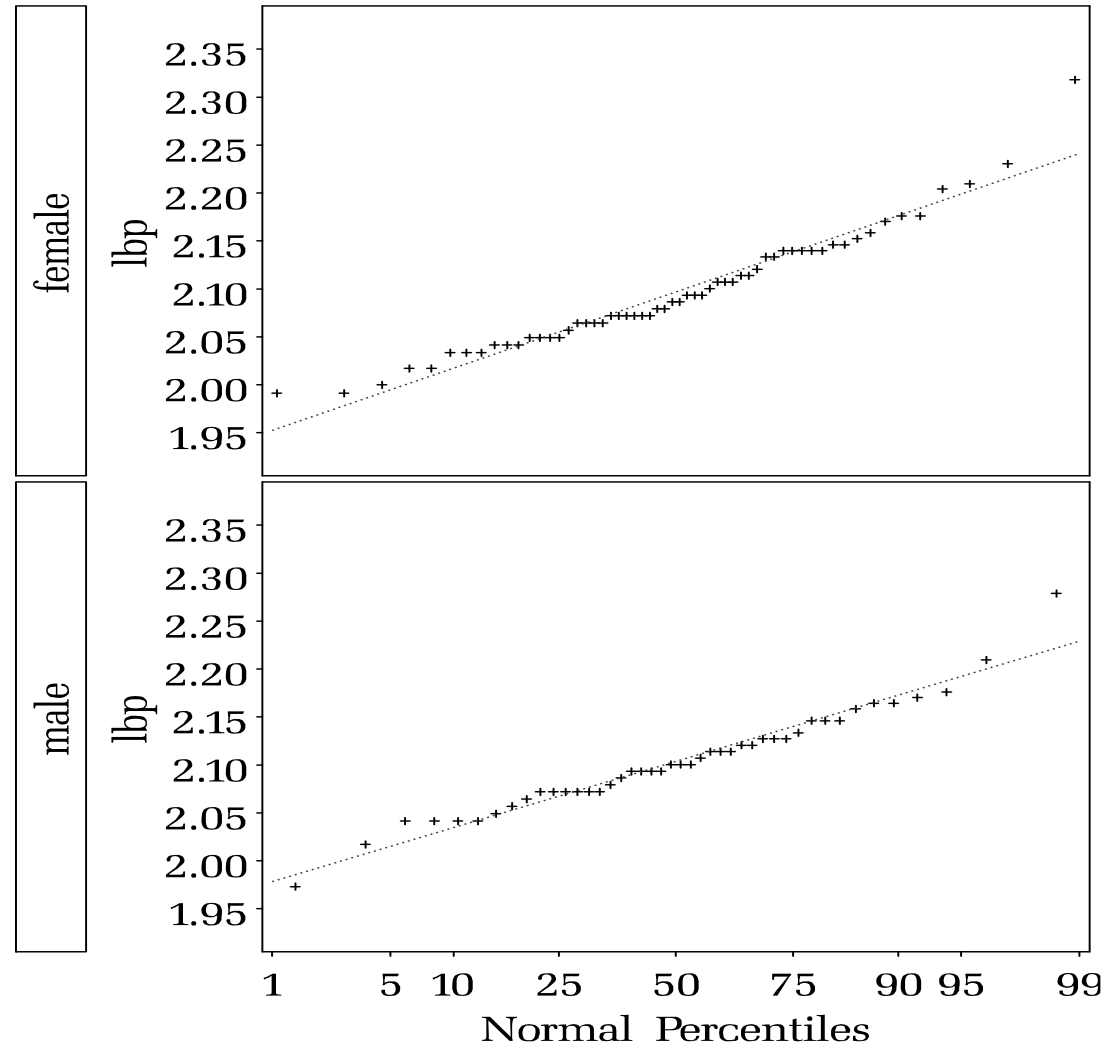
probplot: gives a probability plot for variable lbp

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`

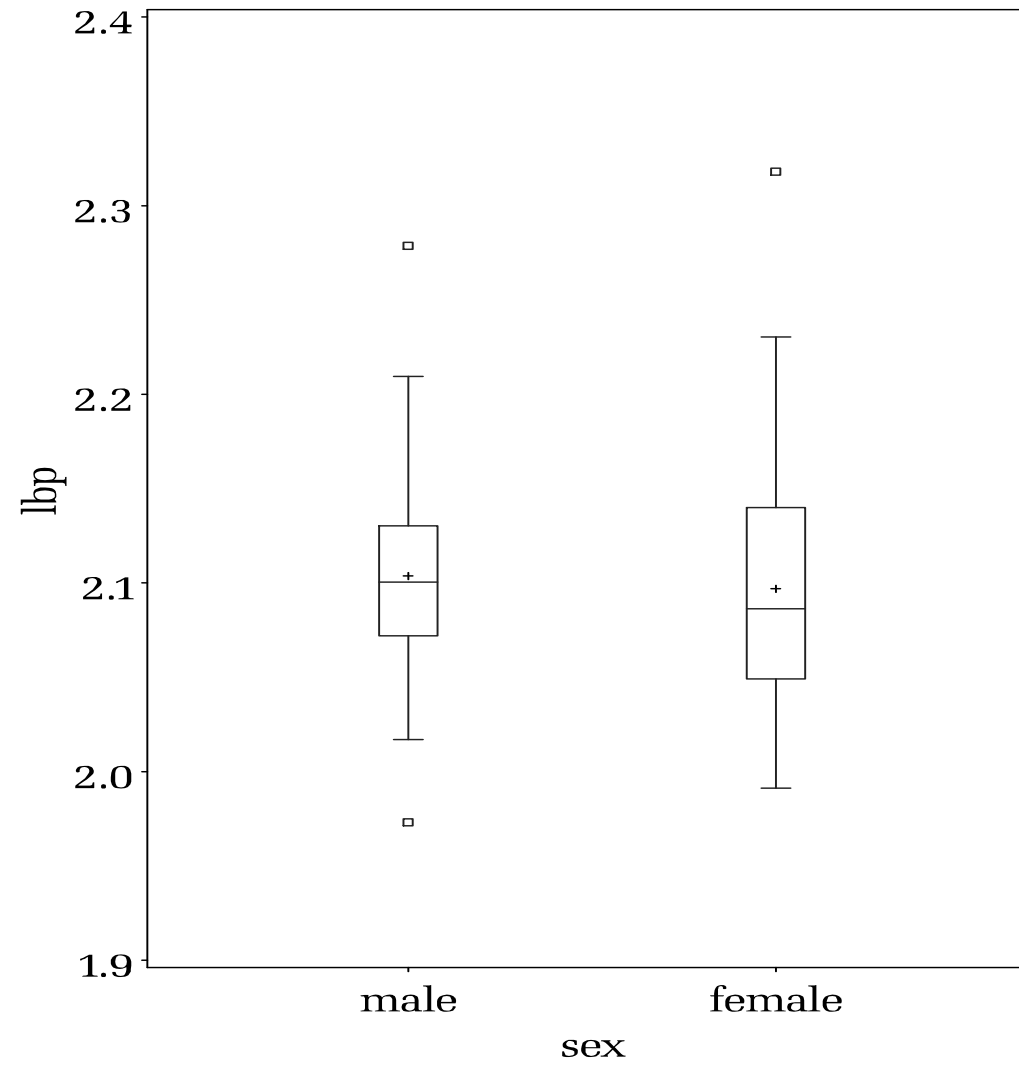
`lbp*sex`: the distribution of `lbp` for each value of `sex`

`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, output 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

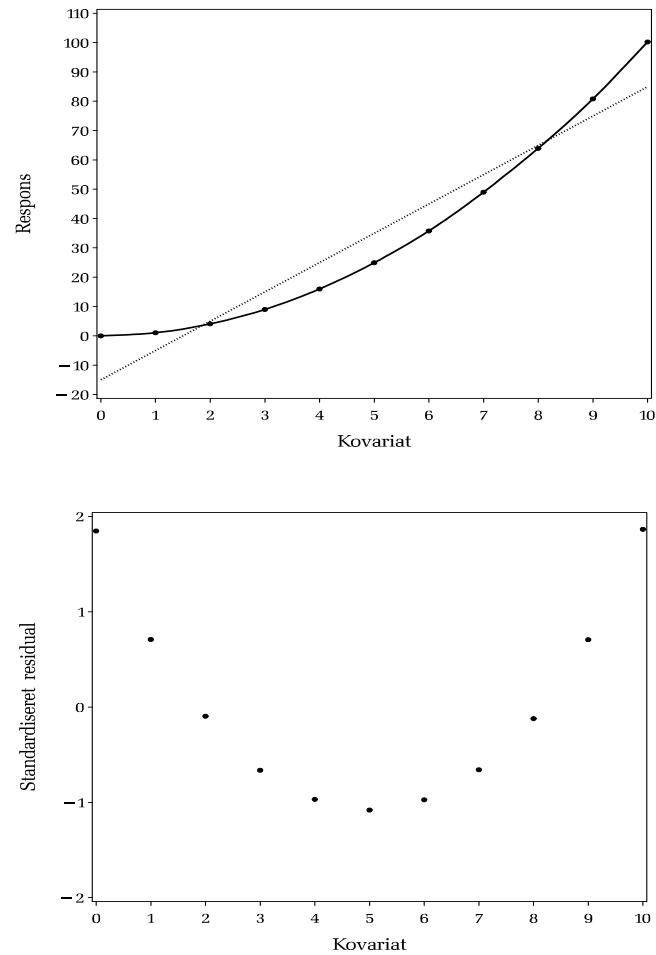
Residuals:  $\hat{\varepsilon}_i = y_i - \hat{y}_i = y_i - (\hat{\alpha} + \hat{\beta}x_i)$

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

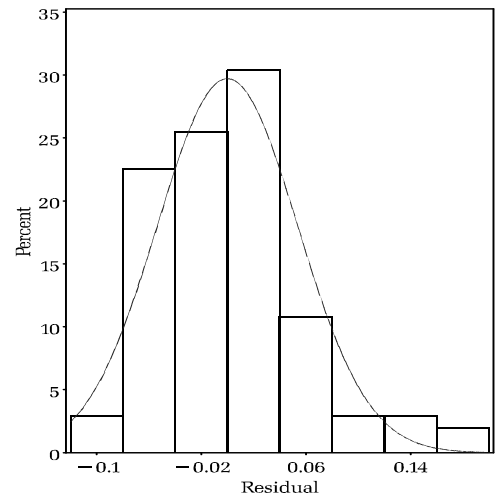
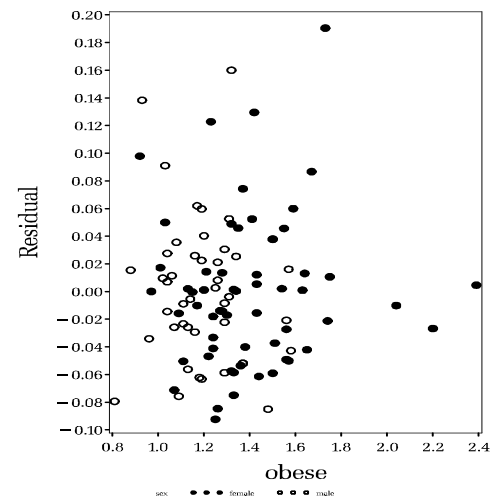
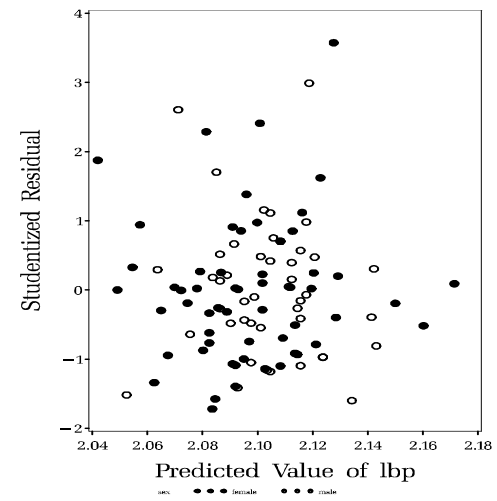
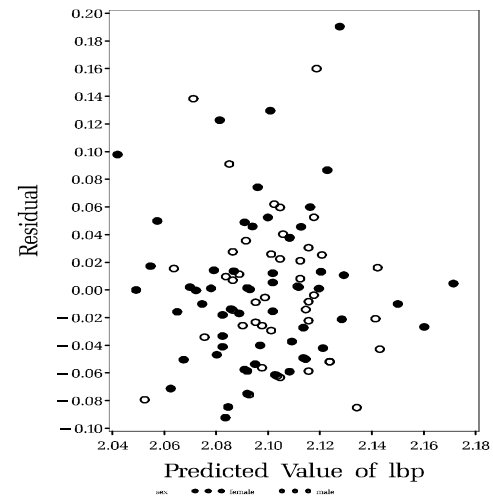
- Ordinary residuals = model deviations:  $\hat{\varepsilon}_i = y_i - \hat{y}_i$   
in SAS denoted `r`
- **Normalised** residuals, also denoted **standardised** residuals or Student residuals  
in SAS denoted `student`

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;
```



## 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.