4. Regression and graphics

Use of SAS
January 2013

Contents

Example: Obesity and blood pressure

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

proc print data=other.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:

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

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

Correlations in SAS

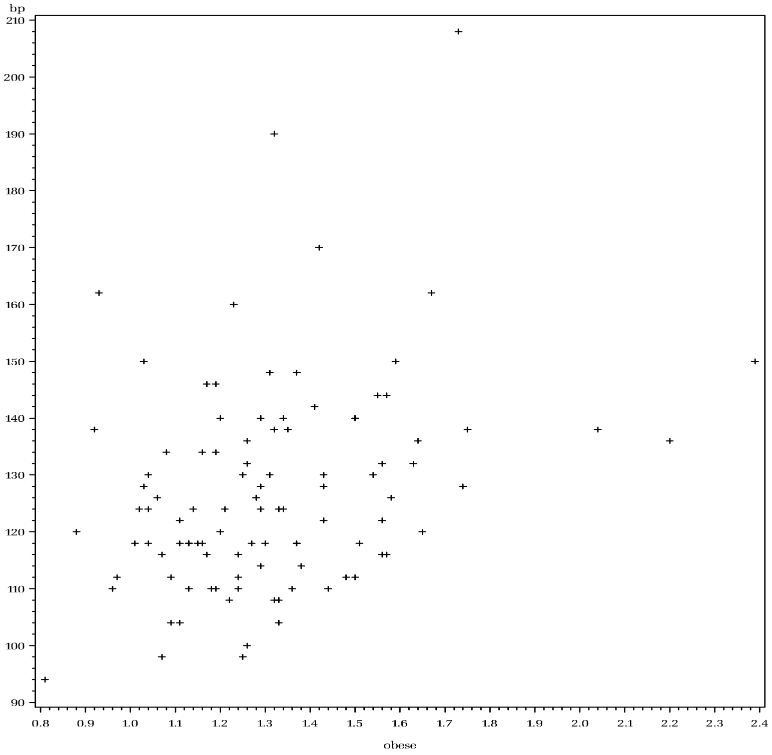
```
proc corr data=other.bp pearson spearman;
    var bp obese;
run;
```

```
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 data=other.bp;
  plot bp*obese;
run;
```



This plot can be improved a lot....

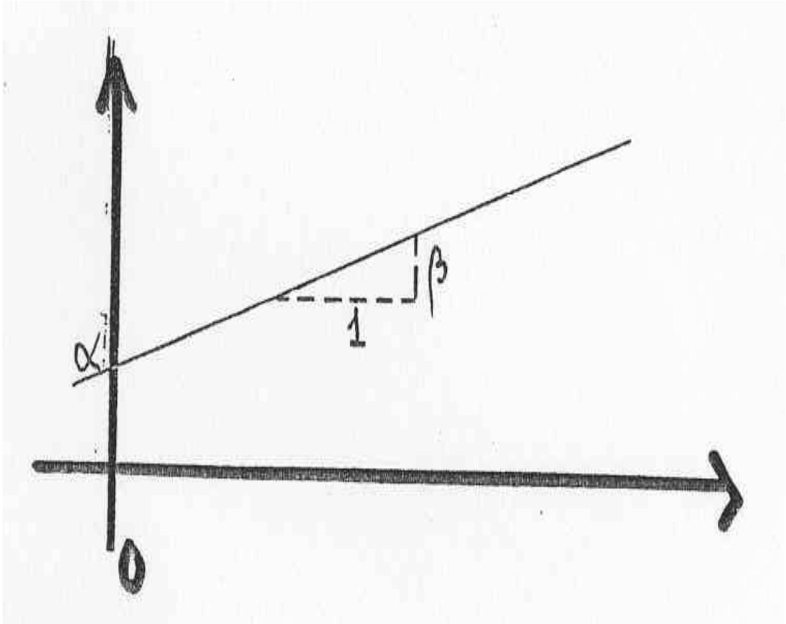
Linear regression

- **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: $Y = \alpha + \beta X$

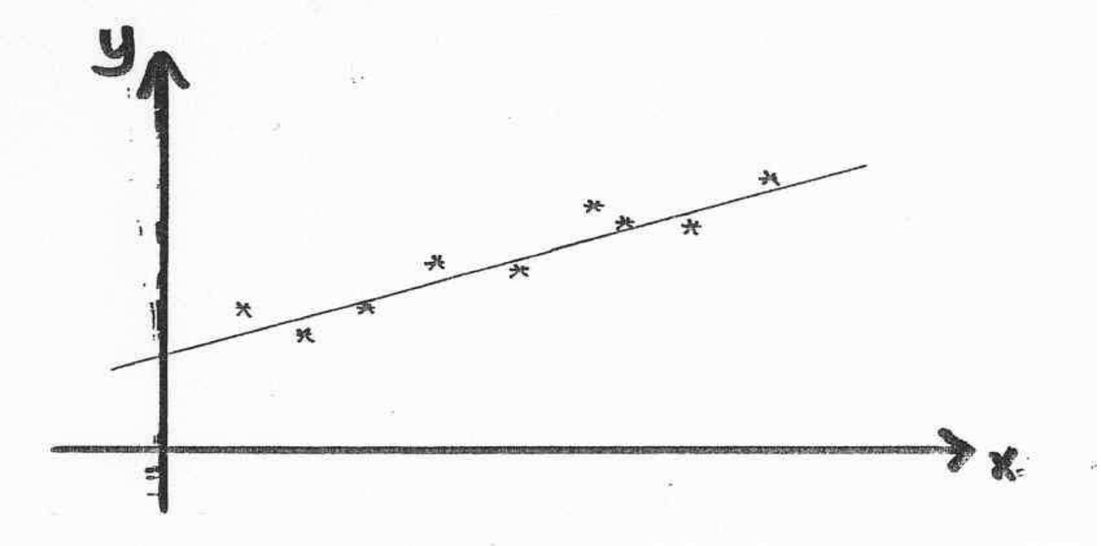


Interpretation:

- 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

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

Regression in SAS

```
proc reg data=other.bp;
  model bp=obese;
run;
```

```
                         The REG Procedure
                           Model: MODEL1
                      Dependent Variable: bp

              Number of Observations Read          102
              Number of Observations Used          102


                        Analysis of Variance

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

Model                     1      3552.41931     3552.41931     11.90   0.0008
Error                   100           29846      298.45541
Corrected Total         101           33398


        Root MSE             17.27586    R-Square     0.1064
        Dependent Mean      127.01961    Adj R-Sq     0.0974
        Coeff Var            13.60094
```

```
                        Parameter Estimates

                     Parameter       Standard
Variable     DF       Estimate          Error    t Value    Pr > |t|

Intercept     1       96.81793        8.91960      10.85      <.0001
obese         1       23.00135        6.66701       3.45      0.0008
```

Interpretation

The estimated relation is

$\text{bp} = 96.8 + 23.0 \times \text{obese}$

Interpretation: When `obese` increases with one unit, `bp` will increase with 23.0 units

One unit `obese` increase in `obese` is a lot

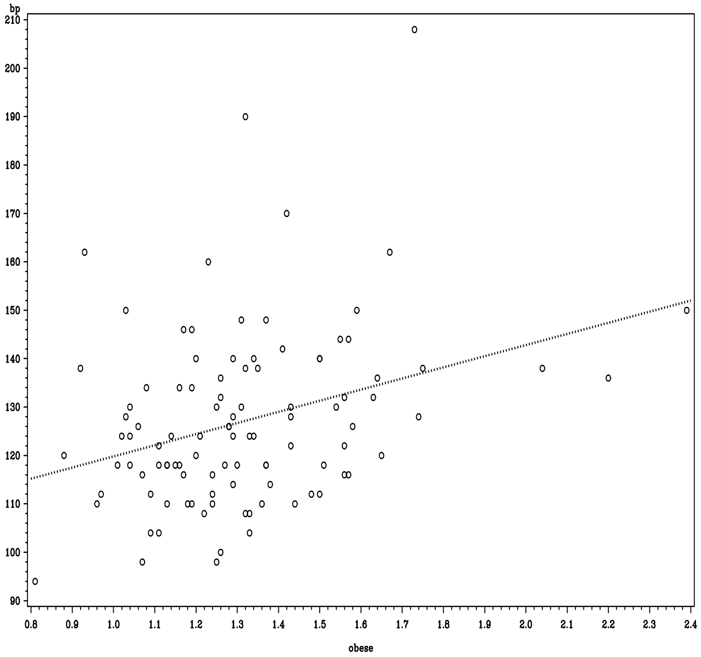
Alternative interpretation:

When `obese` increases with 0.1 units, `bp` will increase with 2.3 units

Add a regression line to the plot

```
proc gplot data=other.bp;
  plot bp*obese;
symbol1 v=circle i=rl l=33;
run;
```

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



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 298.5

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

$$s = \sqrt{s^2} = 17.28$$

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:

$$
\begin{aligned}
& \hat{\beta} \pm t_{97.5\%}(n-2) \times se(\hat{\beta}) \\
= \quad & 23.001 \pm 1.984 \times 6.667 = (9.774; 36.229)
\end{aligned}
$$

use the option `clb` in SAS:

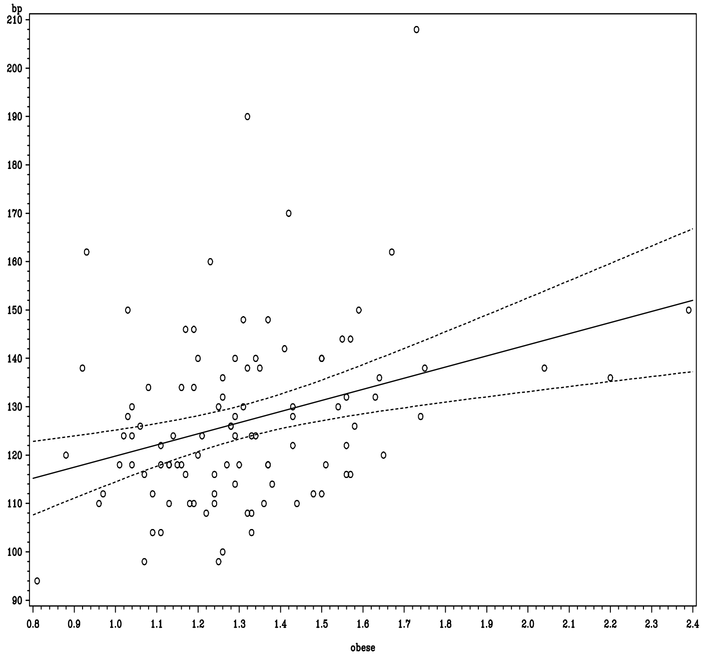
```
proc reg data=other.bp;
  model bp=obese / clb;
run;
```

```
                        Parameter       Standard
Variable       DF        Estimate          Error    t Value    Pr > |t|

Intercept       1        96.81793        8.91960      10.85      <.0001
obese           1        23.00135        6.66701       3.45      0.0008

                           Parameter Estimates

            Variable       DF        95% Confidence Limits

            Intercept       1        79.12169       114.51417
            obese           1         9.77420        36.22851
```

Confidence interval for regression line

```
proc gplot data=bp;
  plot bp*obese;
symbol1 v=circle i=rlclm95 l=1;
run;
```

`i=irclm95` gives confidence limits



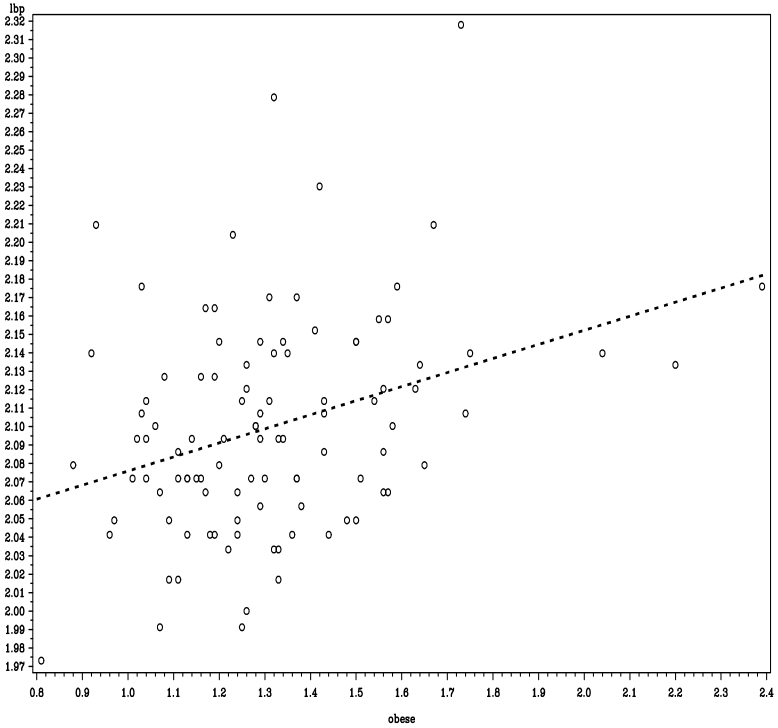
Assumptions in linear regression

- 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 fulfilled here, so we try a logarithmic transformation of the outcome:

```
data bp;
set other.bp;
lbp=log10(bp);
run;
```

After the logarithmic transformation of blood pressure:



Is the relationship still linear?

Regression coefficents - log-transformed outcome

```
                       Parameter Estimates

                    Parameter      Standard
Variable    DF       Estimate         Error    t Value    Pr > |t|

Intercept    1        1.99956       0.02857      69.99      <.0001
obese        1        0.07636       0.02136       3.58      0.0005
```

Interpretation - log-transformed outcome

The estimated relation is
$\log_{10}(\text{bp}) = 2.000 + 0.0764 \times \text{obese}$

Interpretation: When `obese` increases with one unit, $\log_{10}$-`bp` will increase with 0.0764 units

This can be *back-transformed* to the original scale:

- $\log_{10}(Y_2) - \log_{10}(Y_1) = 0.0764 \Rightarrow$
  $Y_2/Y_1 = 10^{0.0764} = 1.19$

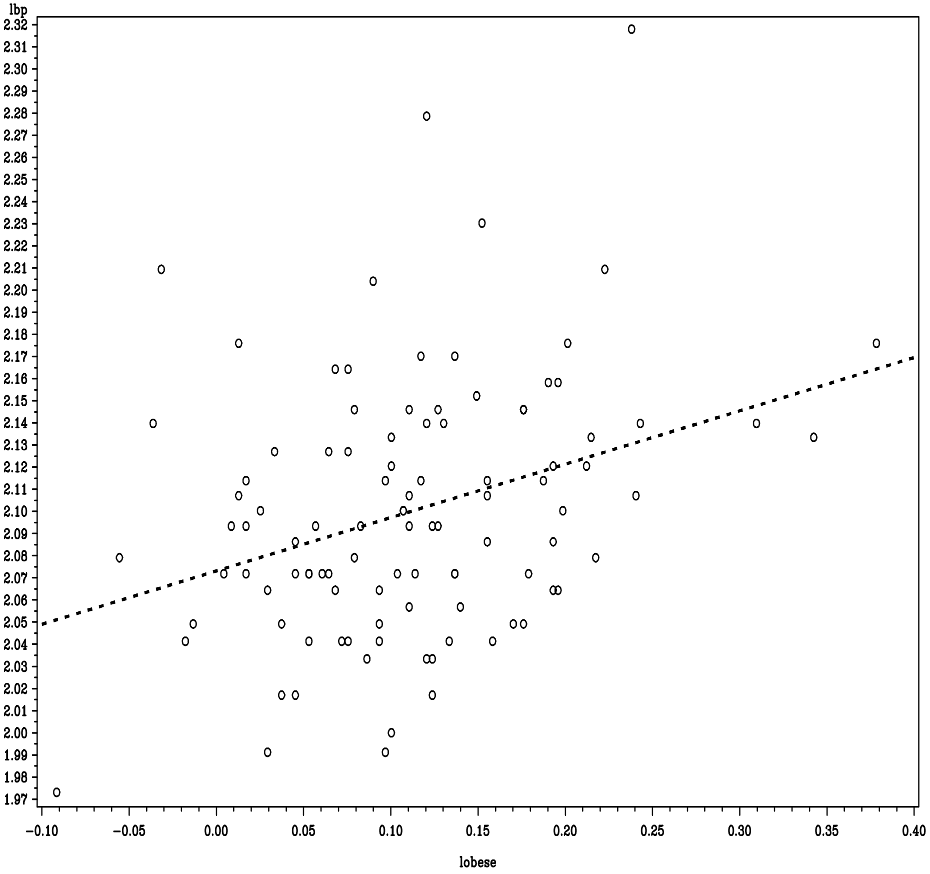
  Thus, an increase of 0.0764 in `lbp` corresponds to an 19% increase in `bp`

Conclusion: a one unit increase in `obese` results in a 19% increase in `bp`

Log-transformation of both variables

The relationship between log-bp and obese seemed not to be linear so we also transform the covariate

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

proc gplot;
  plot lbp*lobese;
symbol v=circle i=rl c=BLACK l=33 w=2;
run;
```



```
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 of log-log-realtionship

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 *back-transformed* 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`

A 10-fold increase in `obese` is unrealistic. If I want to look at the effect of a doubling, I would use `LOG2` (instead of `LOG10`).

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 log(SIGF1) and age for prepubertal children.
4. Use PROC REG to do a regression analysis of log(SIGF1) vs. age for prepubertal children.
5. Also do the untransformed analysis, i.e., SIGF1 vs age. Are the modelling assumptions satisfied?

Scatter plots in SAS: PROC GPLOT

In the raw form:

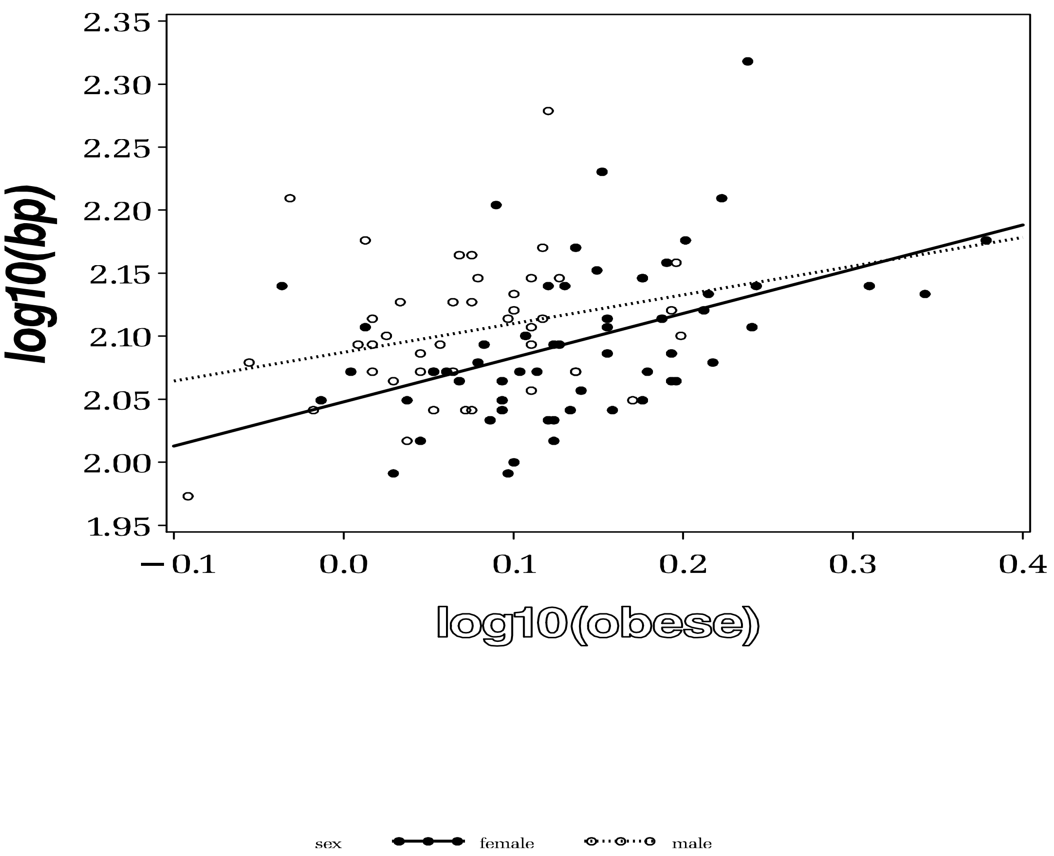
```
proc gplot data=bp;
  plot bp*obese;
run;
```

more code can give nicer output:

```
proc gplot data=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’

obesity and blood pressure



Plotting symbol depend on sex

Achived with the statement

```
plot lbp*lobese=sex
```

Females gets the firts plotting symbol (`symbol1`)

Males gets the second plotting symbol (`symbol2`)

You have to define the symbols

Symbol statements

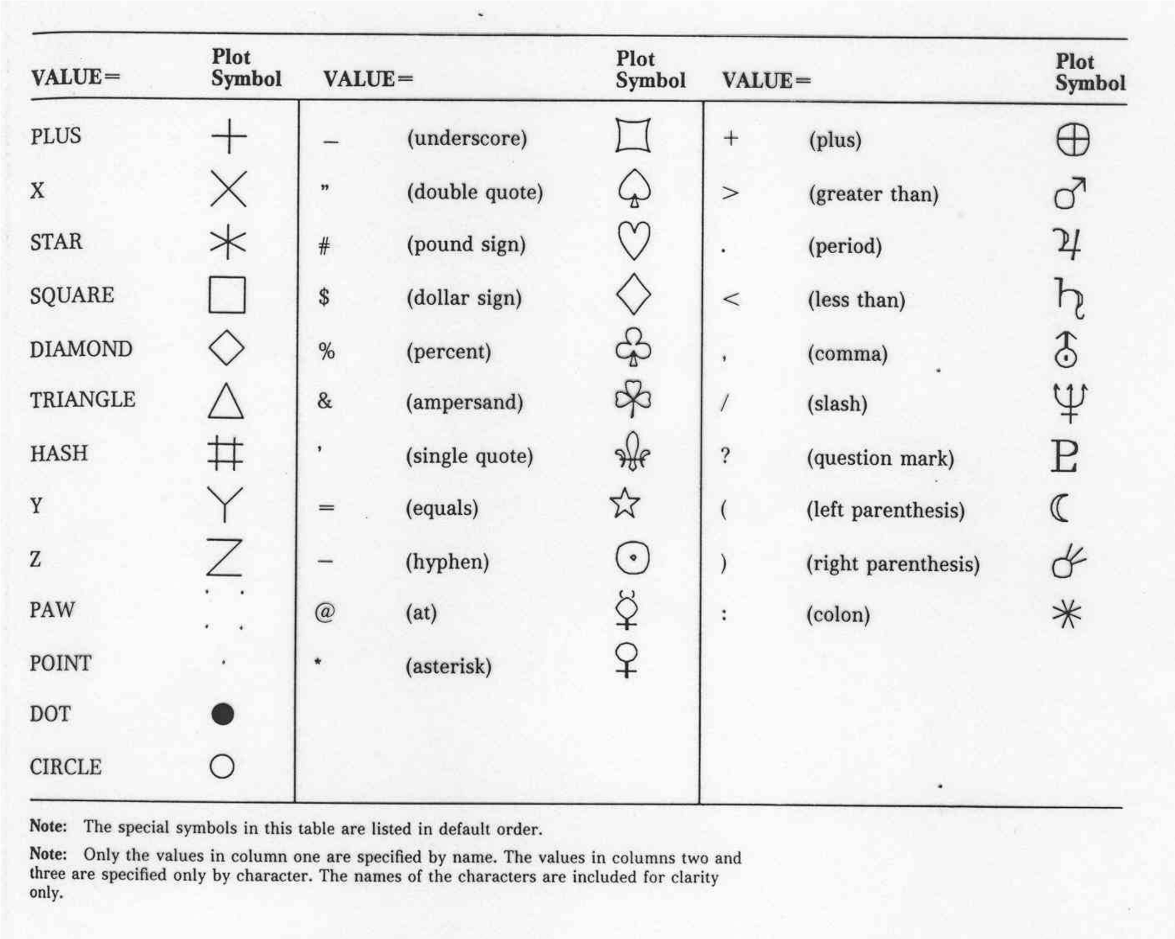
One symbol statement for each group
(each value of `sex`)

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`: color 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` statement

- `i=none`: data points not connected
- `i=rl`: fits a regression line
- `i=rq`: fits 2. degree polynomium
- `i=spline`: fits 'smooth' curve to the points
- `i=join`: data points connected with straight lines (first sort data)

Interpolation methods, more information

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

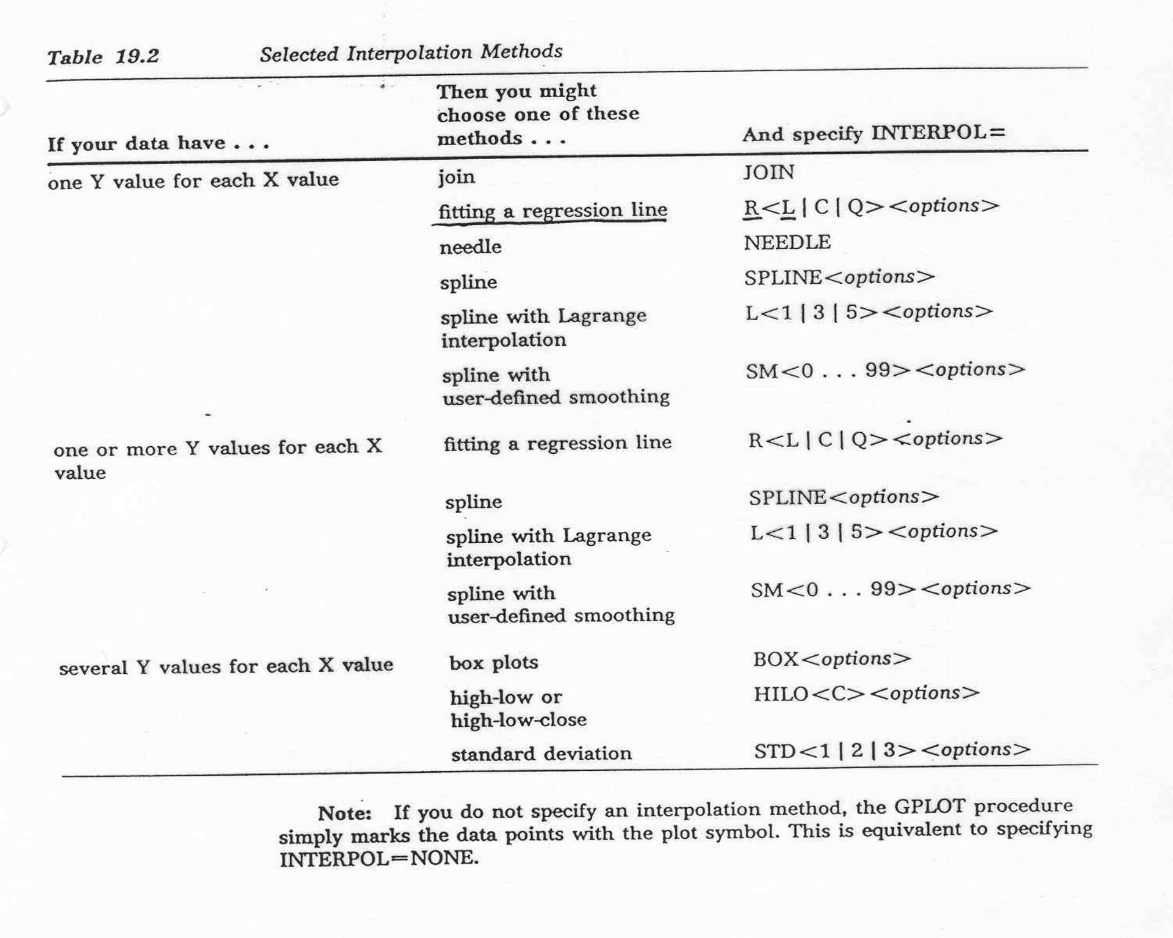


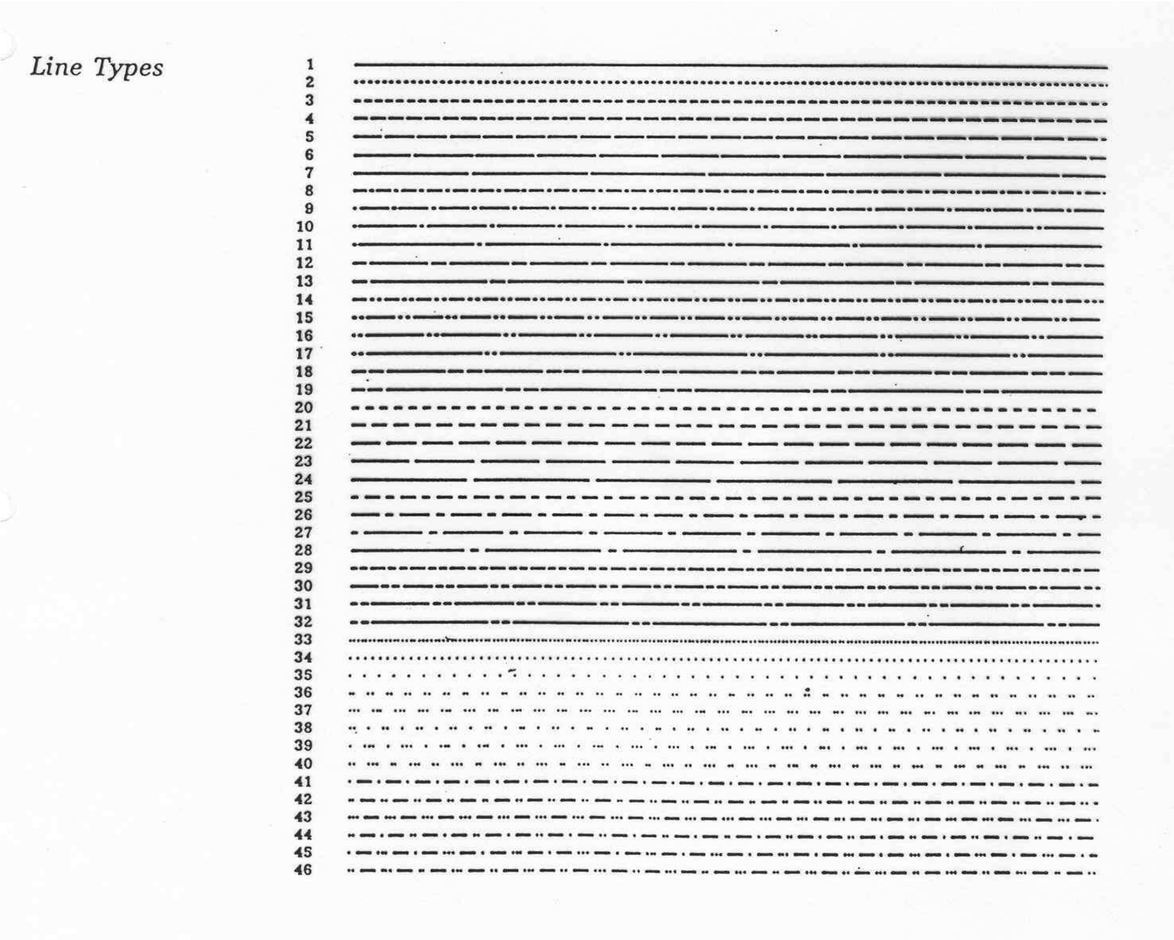
Table 19.2 *Selected Interpolation Methods*

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

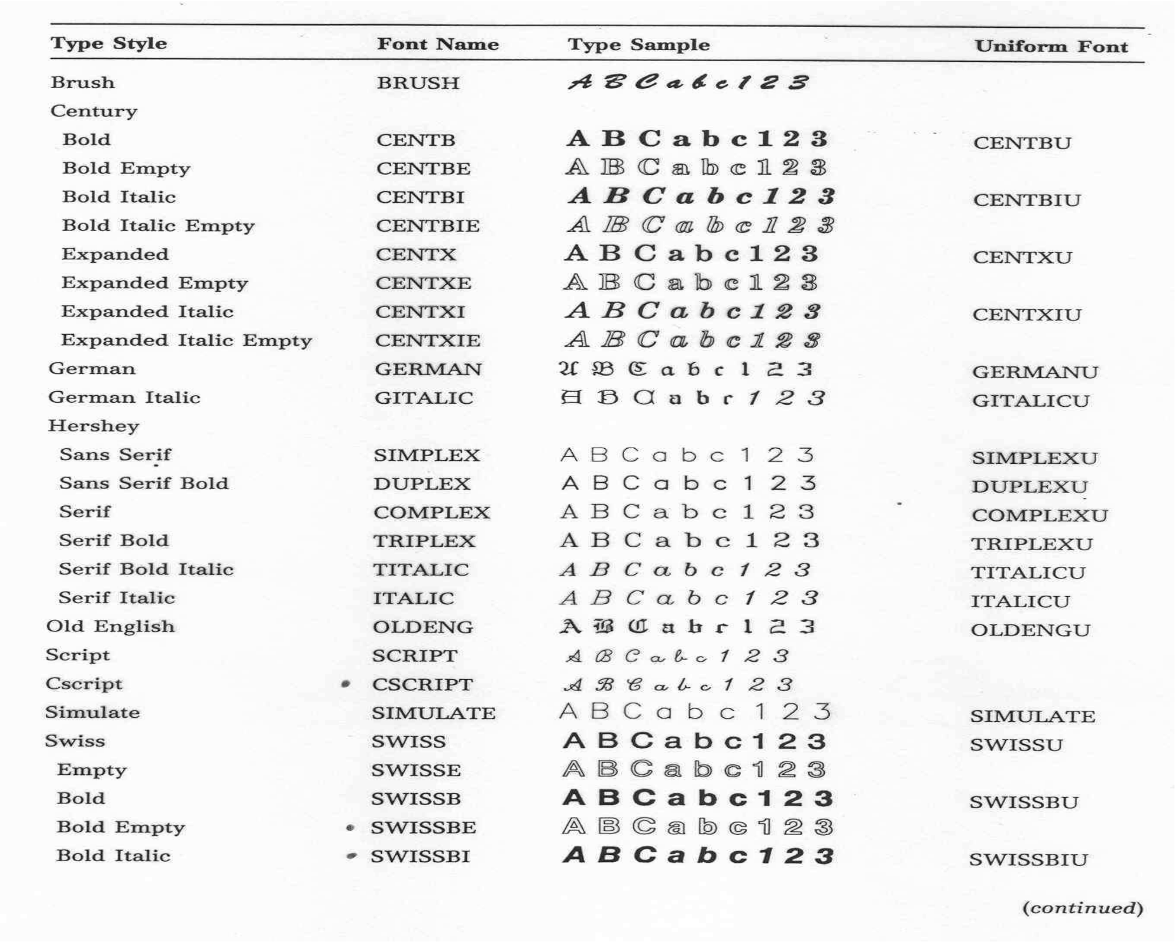


AXIS specifications

- `haxis=axis1 vaxis=axis2`: horizontal axis is axis1 and vertical axis is axis2. axis1 and axis2 must be 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	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=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

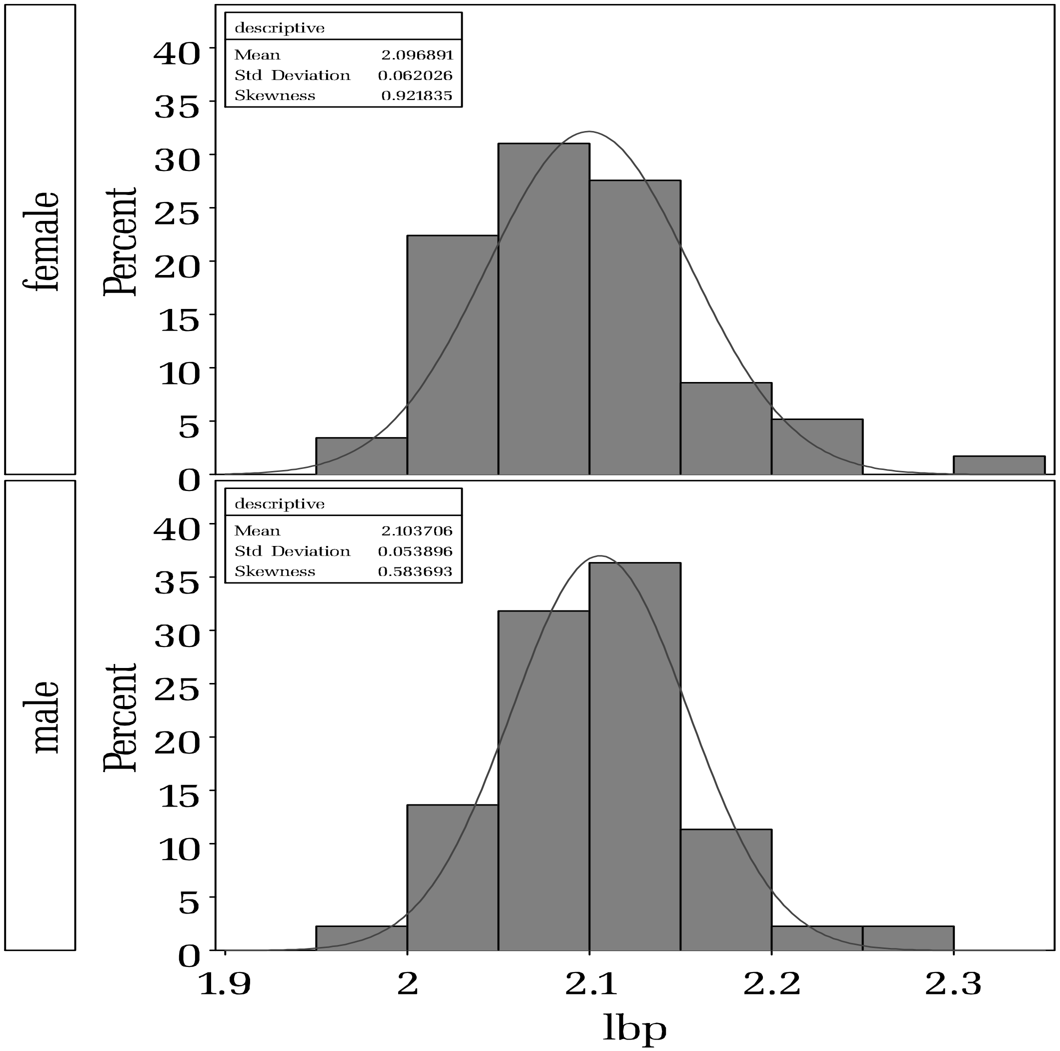
`class sex`: histogram for both values of `sex`

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

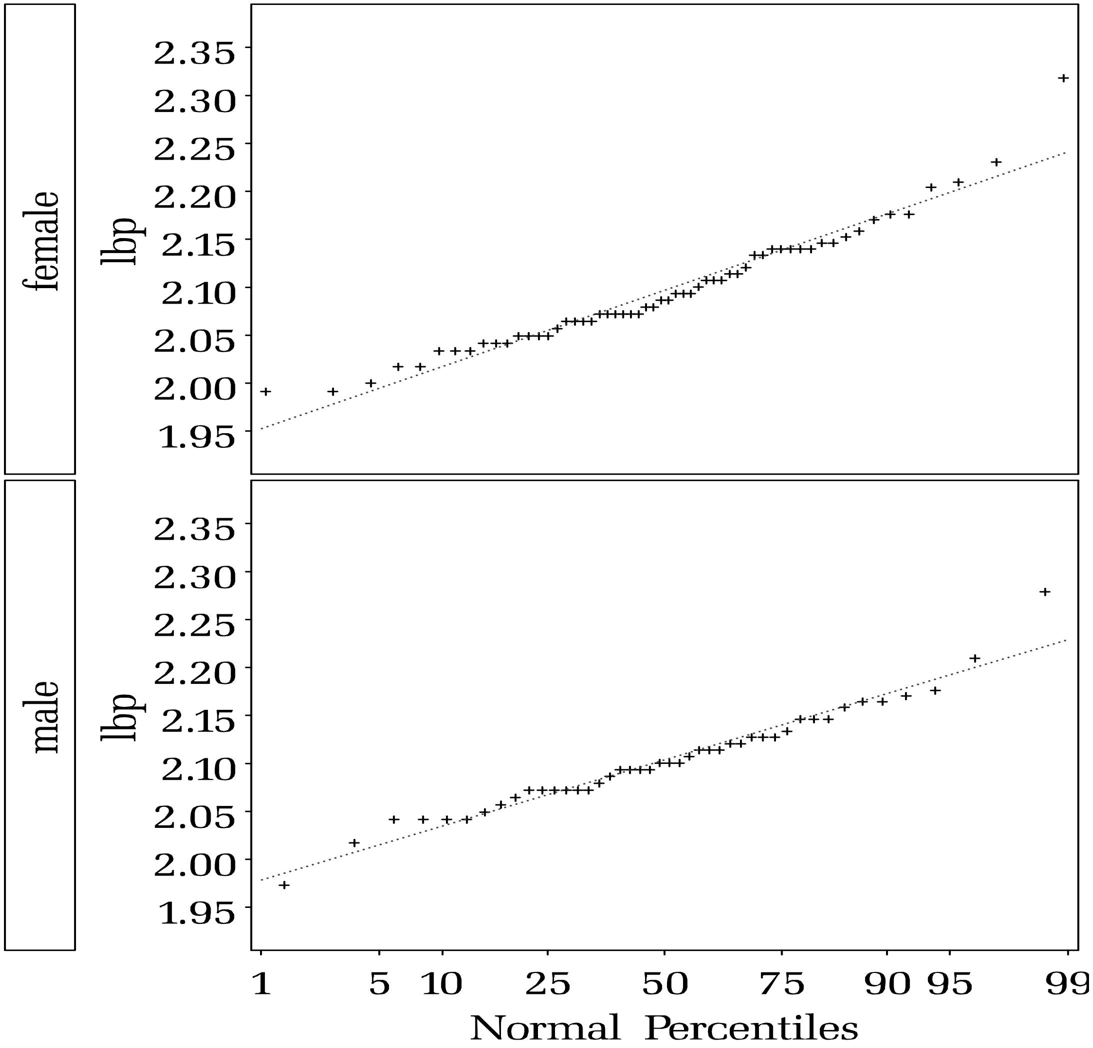
`probplot`: gives a probability plot for variable `lbp`

`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=bp;
  plot lbp*sex / height=3 boxstyle=schematic;
run;
```

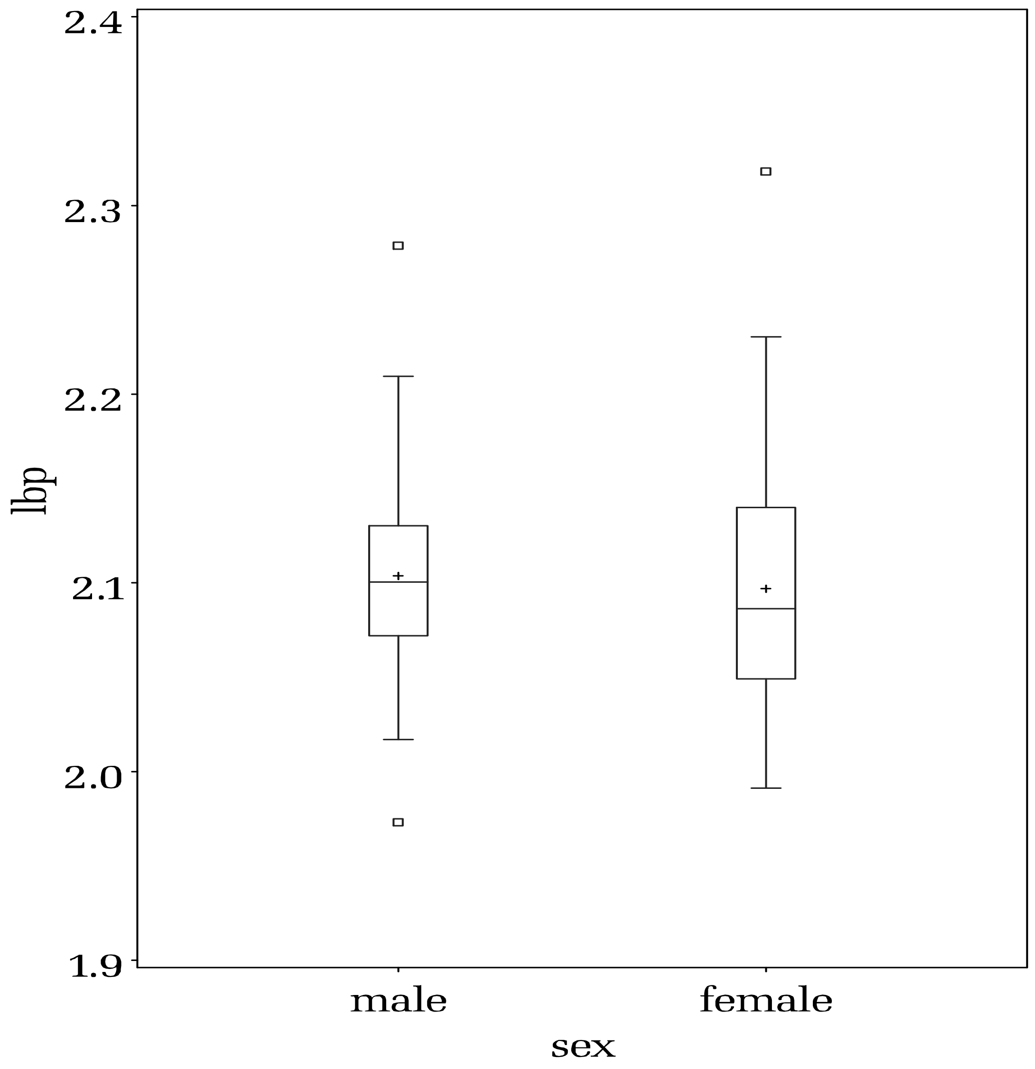
use `proc boxplot` *not* `proc univariate`

`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. A pdf-file can also be generated.

Regression results into WORD

Using similar statements for `PROC REG` will allow you to copy regression results into WORD, e.g.,

```
ods rtf file='p:\resultater.rtf' BODYTITLE STARTPAGE=NO STYLE=journal;
      ODS NOPROCTITLE;
      proc reg data=other.bp;
        model bp=obese;
      run;
      quit;
ods rtf close;
```

Graphical model control in regression analysis

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

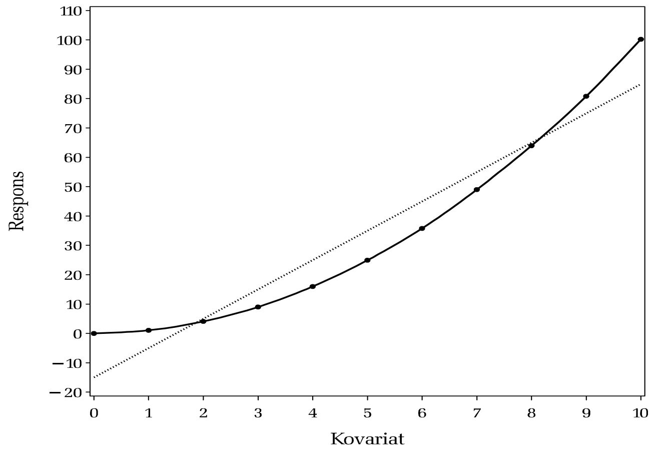
Residuals are plotted against:

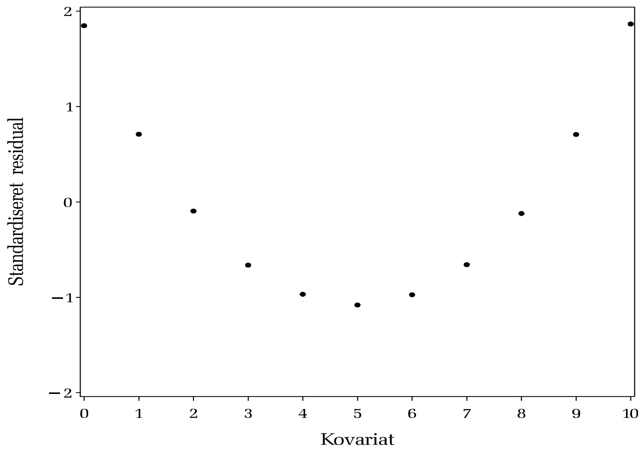
- the explanatory variable x_i
  – to check linearity
- the fitted values $\hat{y}_i$
  – to check variance homogeneity

Figures should give an impression of pure scatter.

Whether residuals follow a normal distribution can be explored using a *histogram* or a *probability plot*.

Residual-plots and linearity





Various types of residuals

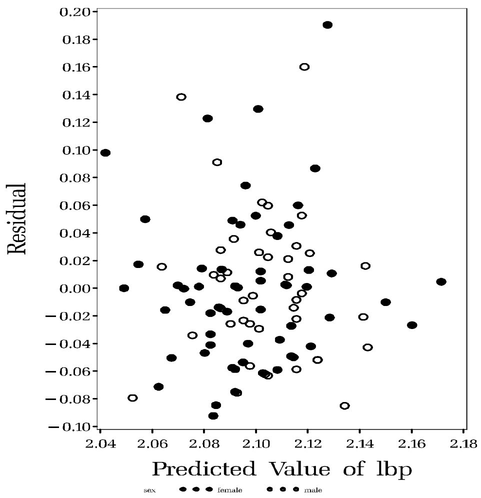
- Ordinary residuals = model deviations: $\hat{\varepsilon}_i = y_i - \hat{y}_i$
  in SAS denoted `r`
- **Normalized** residuals, also denoted **standardized** residuals
  or Student residuals
  in SAS denoted `student`

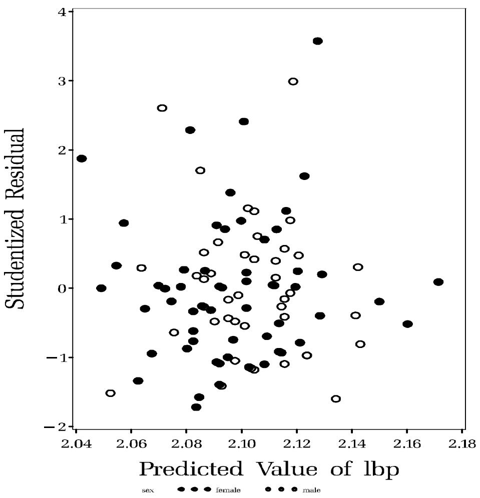
In the procedure **REG** we can calculate **and save** all these quantities for later use:

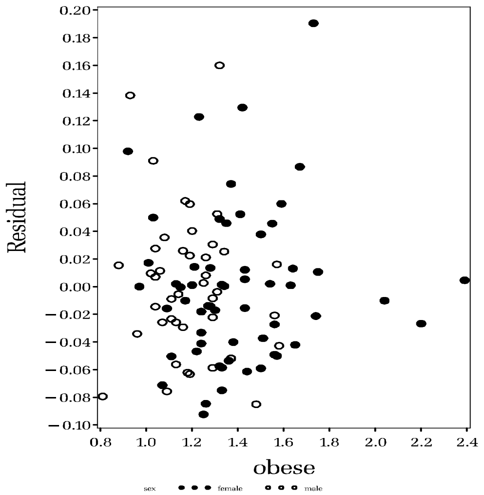
```
proc reg data=bp;
  model lbp=lobese;
output out=res p=yhat r=resid student=student;
run;
```

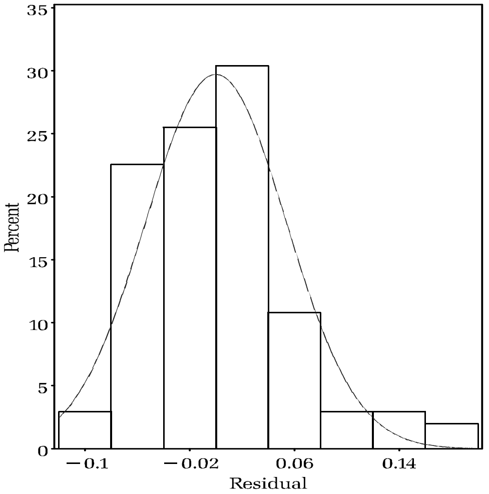
Here, we get a new data set `res` (`work.res`) containing 3 new variables (`yhat`, `resid`, `student`), which we may then use, e.g. to make a residual plot:

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


Residual
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
−0.10
2.04
2.06
2.08
2.10
2.12
2.14
2.16
2.18
Predicted Value of lbp
sex
female
male


Studentized Residual
4
3
2
1
0
−1
−2
2.04
2.06
2.08
2.10
2.12
2.14
2.16
2.18
Predicted Value of lbp
sex
female
male


Residual
0.20
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−0.08
−0.10
0.8
1.0
1.2
1.4
1.6
1.8
2.0
2.2
2.4
obese
sex
female
male


Percent
35
30
25
20
15
10
5
0
−0.1
−0.02
0.06
0.14
Residual

Exercise: Regression and graphics II

Consider again the regression analysis of log(SIGF1) 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.
3. Create RTF-files for your regression output and graphs.