3

Linear Regression

This chapter is dedicated to the measures of association between two numeric variables (linear correlation and non-parametric correlation) and to the linear regression model: estimation of the regression coefficients, fitting and prediction from new data with confidence intervals.

3.1. Measures of association between two numeric variables

3.1.1. *Bivariate descriptive statistics*

The command `summarize` provides information about the central tendency, the dispersion and the range of the values observed for a list of variables. It is therefore quite possible to use it to summarize the distribution of two numeric variables. In the case of the weight of babies (`bwt`) and mothers (`lwt`), the following results are available (the mothers' weights are expressed in pounds, and they are initially converted in kilograms):

```
. replace lwt = lwt/2.2
. summarize bwt lwt

lwt was int now float
(189 real changes made)

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
         bwt |       189    2944.587    729.2143        709       4990
         lwt |       189    59.00673    13.89972   36.36364   113.6364
```

Note that the option `detail` allows more detailed information to be provided (quartiles, skewness and kurtosis, etc.).

Before calculating any linear or monotonic measure of association, it is advised that one should visually inspect a scatter plot that describes the covariation between the two sets of measurements. To do this, we use the command `scatter` specifying the two variables of interest: the first will be reported on the *y*-axis and the second on the x-axis:

```
. scatter bwt lwt
```

Superimposing a lowess curve on the previous graph gives an idea of the linearity of the relationship between the two variables and possible local deviations. To do this, it suffices to combine two `twoway` commands, delimiting them with parentheses. In the following case, the first command `(scatter bwt lwt)` displays the scatterplot, and the second `(lowess bwt lwt)` displays a lowess curve (Figure 3.1). Note that, `lowess` could be replaced by `lfit` to draw the regression line:

```
. twoway (scatter bwt lwt) (lowess bwt lwt), legend(off) ytitle("bwt")
```

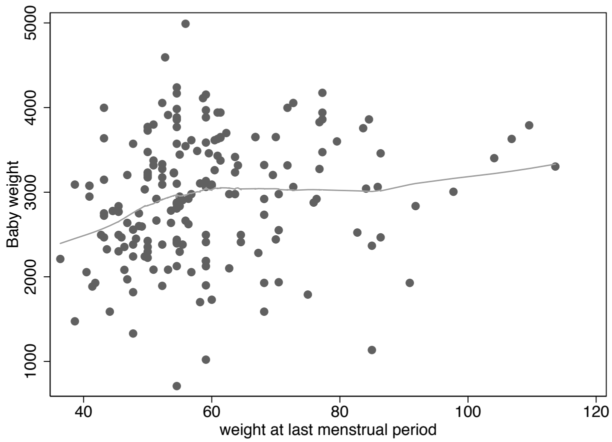

Figure 3.1. *Babies' (g) and mothers' weight (kg)*

Note that with the recent versions of Stata, the previous expression can be formulated:

```
. scatter bwt lwt || lowess bwt lwt, legend(off) ytitle("bwt")
```

In this case, the operator `||` is employed to separate the multiple graphics instructions.

In the case where more than two variables are involved, the command `graph matrix`, followed by the list of variables of interest, makes it possible to display all the scatterplots cross-tabulating the variables two by two.

3.1.2. *Pearson's correlation*

The command providing the Pearson's correlation coefficient between two numeric variables is `correlate`:

```
. correlate lwt bwt

(obs=189)

             |      lwt      bwt
-------------+------------------
         lwt |   1.0000
         bwt |   0.1857   1.0000
```

Note that an option `means` could be added to the previous command in order to simultaneously display the results produced by `summarize`. Normally reserved for the case of correlations between more than two variables, it is also possible to employ the command `pwcorr`. Nonetheless, the latter has the advantage of providing the results of the test of the null hypothesis for the parameter of interest:

```
. pwcorr lwt bwt, obs sig

             |      lwt      bwt
-------------+------------------
         lwt |   1.0000
             |
             |      189
             |
         bwt |   0.1857   1.0000
             |   0.0105
             |      189      189
             |
```

The command `corrci` provides an estimate of the confidence interval for the Pearson coefficient (by default, by making use of the inverse Fisher transformation). The option `level()` allows the desired confidence level to be modified:

```
. corrci lwt bwt
```

```
(obs=189)

          correlation and 95% limits
lwt bwt      0.186    0.044    0.320
```

3.1.3. *Non-parametric correlation*

When it is preferable to work with a measure of correlation based on the ranks of the observations, the Spearman coefficient of correlation will be employed with the command `spearman`:

```
. spearman lwt bwt

 Number of obs =      189
Spearman's rho =        0.2489

Test of Ho: lwt and bwt are independent
    Prob > |t| =        0.0006
```

The command `spearman` also works with more than two variables and returns, such as `pwcorr`, a matrix of correlation coefficients with eventually the degree of significance (adding the option `stats(rho p)`).

3.2. Linear regression

3.2.1. *Estimation of the model parameters*

In Chapter 2, we have seen that the command `regress` (p. 40) is utilized when modeling the relationship between two variables, one being considered as a response variable. In this case, to model the relationship between the babies' weight (response variable) and the mothers' weight (explanatory variable), the following syntax would be used:

```
. regress bwt lwt

      Source |       SS       df       MS              Number of obs =     189
-------------+------------------------------           F(  1,   187) =    6.68
       Model |   3448639.3     1   3448639.3           Prob > F      =  0.0105
    Residual |  96521016.5   187  516155.168           R-squared     =  0.0345
-------------+------------------------------           Adj R-squared =  0.0293
       Total |  99969655.8   188  531753.488           Root MSE      =  718.44
```

```
------------------------------------------------------------------------------
         bwt |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         lwt |   9.744038   3.769686     2.58   0.011     2.307461    17.18061
       _cons |   2369.623   228.4932    10.37   0.000     1918.868    2820.379
------------------------------------------------------------------------------
```

For any regression model in Stata, the variable response comes first, followed by the explanatory variable(s). This command provides the analysis of variance (ANOVA) table for the regression as well as the regression coefficients table (with confidence intervals), whose displaying may be omitted by including the option `notable`. It is always possible to redisplay the results of the regression model by simply typing the name of the command, `regress` (this is valid for the other regression models in Stata).

It is also possible to store or display the regression coefficient for the variable `lwt` only (slope of the regression line) by operating on the values returned by Stata. For example, the following statement displays the requested result:

```
. display _b[lwt]
9.7440378
```

This is a so-called *post-estimation* result, as discussed in Chapter 2. The command `ereturn list` provides the list of the postestimation values stored by Stata. In this case, the regression coefficients (y-intercept and slope) are stored in an object called `e(b)`. If the previous instruction `regress` is slightly modified, it can be verified that these values are individually accessible:

```
. regress bwt lwt, noheader coeflegend
```

```
------------------------------------------------------------------------------
         bwt |      Coef.  Legend
-------------+----------------------------------------------------------------
         lwt |   9.744038  _b[lwt]
       _cons |   2369.623  _b[_cons]
------------------------------------------------------------------------------
```

The value of the coefficient of determination can also be displayed by making use of `e(r2)`. In the following illustration, we insert the command `display` with a combination of text and numeric value (rounded to two decimal places):

```
. display "Coefficient of determination = " %3.2f e(r2)*100 " %"
Coefficient of determination = 3.45 %
```

When using `display`, the numerical results displayed on-screen can be formatted by prefixing the name of the variable whose content is to be displayed by a formatting instruction `%x.yf` (`x` values, including `y` decimal places).

The regression line can be represented graphically by means of the command `twoway lfit bwt lwt`, but as was mentioned in the previous section, this command can be combined with a simple scatter plot, for instance:

```
. twoway (scatter bwt lwt) (lfit bwt lwt)
```

3.2.2. *Pointwise and interval-based prediction*

In Stata, the general principle consists of estimating the parameters of a regression model to then proceed with postestimation commands. This is valid for the calculation of the predicted values or residuals of the regression model. When we want to calculate the fitted values for the model (predicted values of `bwt` for the observed values of `lwt`), we will make use of the command `predict` after an estimation command such as `regress`. The predicted values will always correspond to the latest regression model:

```
. predict double p, xb
```

The option `xb` (which is the default) provides the fitted values of the previous model. It is important to remember that a variable name be specified, here `p`, to store the predictions! The option `double` makes it possible to restrict the size of the memory storage of the predicted values.

The previous command does not provide any confidence intervals. However, it is not difficult to obtain the standard error for the fitted values and calculate from predicted values the associated confidence intervals. Take the case of the fitted values:

```
. predict sep, stdp
. generate lci = p - 1.96*sep
. generate uci = p + 1.96*sep
```

The variables `sep`, `lci` and `uci` correspond to the standard error and the upper and lower bounds of the 95% CI, respectively. These values could be used to manually display the regression line and its 95% confidence interval (the command `line` allows lines to be drawn in Stata), but it is easier and faster to use the command `lfitci` as shown hereafter (Figure 3.2):

```
. twoway (lfitci bwt lwt) (scatter bwt lwt)
```

The option stdp will be replaced by stdf to obtain the standard error for the prediction of new observations.

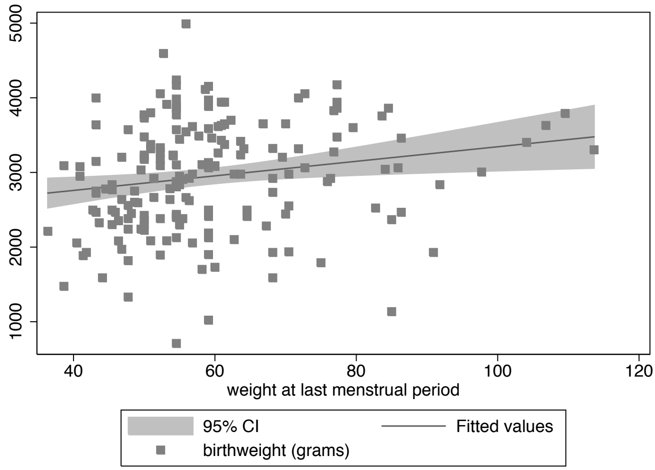

Figure 3.2. *Regression line (and 95% intervals) for the relation between the babies' weight (g) and the mothers' weight (kg)*

3.2.3. *Model diagnostic*

The command estat provides a certain amount of information concerning the goodness of fit of the model and helps to diagnose potential collinearity problems (estat vif) in the case where the model includes several explanatory variables:

```
. estat ic

-----------------------------------------------------------------------------
       Model |    Obs    ll(null)   ll(model)     df          AIC         BIC
-------------+---------------------------------------------------------------
           . |    189    -1513.56   -1510.242      2     3024.485    3030.968
-----------------------------------------------------------------------------
               Note:  N=Obs used in calculating BIC; see [R] BIC note
```

It is also possible to use the external command fitstat (to be installed from the internet, findit fitstat) for a more detailed summary of the goodness of fit of the model:

```
. fitstat
```

```
Measures of Fit for regress of bwt

Log-Lik Intercept Only:      -1513.560   Log-Lik Full Model:          -1510.242
D(187):                       3020.485   LR(1):                           6.635
                                         Prob > LR:                       0.010
R2:                              0.034   Adjusted R2:                     0.029
AIC:                            16.003   AIC*n:                        3024.485
BIC:                          2040.278   BIC':                           -1.393
BIC used by Stata:            3030.968   AIC used by Stata:            3024.485
```

To obtain the residuals of the regression model (difference between the observed values and those predicted according to the model), we always make use of the command `predict`, specifying this time an option among: `residuals` (raw residuals), `rstandard` (standardized residuals) or `rstudent` (studentized residuals). In the series of instructions that follows, the three types of residuals are calculated, and their numerical summary is displayed using `summarize` with values rounded to five decimal places:

```
. predict double r, rstandard
. predict double rr, residuals
. predict double rrr, rstudent
. format r-rrr %9.5f
. summarize r-rrr, format

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
           r |       189    -0.00044     1.00210   -3.06017    2.89709
          rr |       189    -0.00000   716.52611  -2.19e+03   2.08e+03
         rrr |       189    -0.00107     1.00806   -3.13139    2.95644
```

In order to visualize the distribution of the residuals, a counts histogram can be employed by means of the command `histogram`, or a density curve representation. Figure 3.3 provides an illustration:

```
. kdensity r
```

It is also interesting to look at the distribution of the residuals with respect to the predicted values to verify the consistency of the variance and the absence of specific variation patterns in the residuals. To do this, it suffices to combine two `twoway` commands, on the same principle as in section 3.1.1 (Figure 3.4):

```
. twoway (scatter r p) (lowess r p), yline(0, lcolor(black) lpattern(dash))
        legend(off)
```

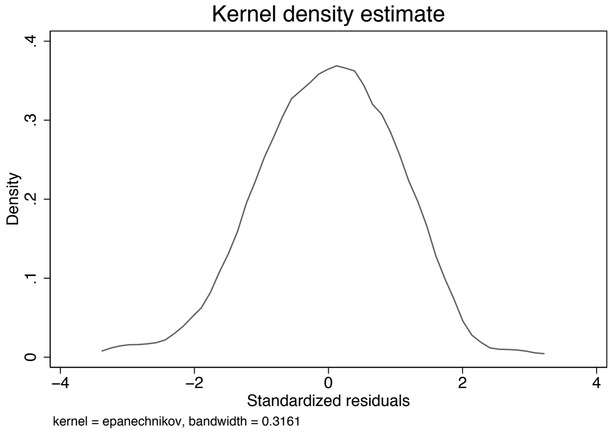

Figure 3.3. *Distribution of the residuals for the regression model of the babies' weight according to the mothers' weight*

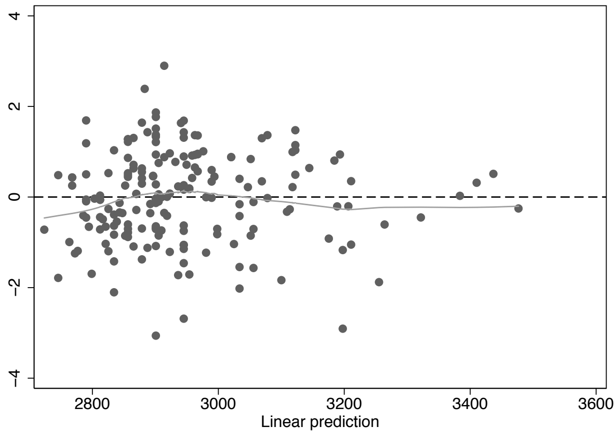

Figure 3.4. *Residuals and predicted values by the regression model*

In fact, the same graph can be obtained directly employing the command `rvfplot` that provides a representation of the residuals according to the predicted values (Figure 3.5):

```
. rvfplot, mlabel(smoke)
```

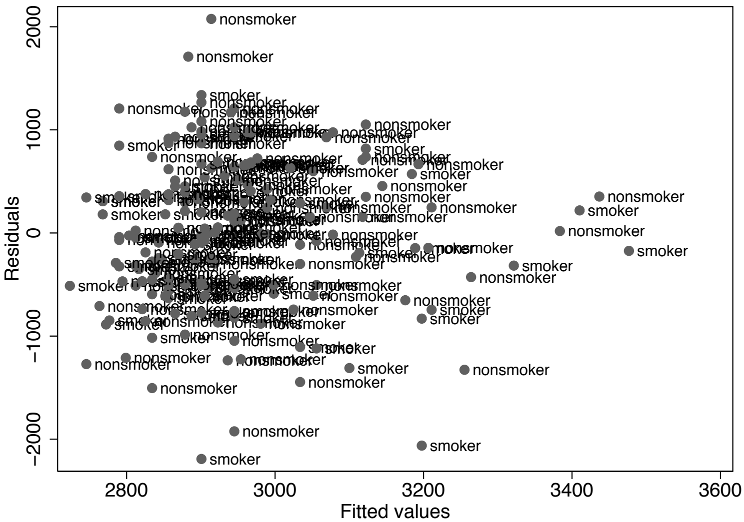

Figure 3.5. *Residuals and predicted values by the regression model including individuals annotation*

3.3. Multiple linear regression

The extension to the multiple regression model does not really raise any problems from the point of view of the instructions: the command `regress` receives the name of the response variable followed by all the explanatory variables. As it is often useful to transform certain variables, or to center them on their average, we will use this section to indicate how to center an explanatory variable. Knowing that when entering `summarize` this command generates a certain amount of information that can be used after (see `return list`), it is not difficult to center the variable relative to the mothers' weights on its average by proceeding as indicated in the following way:

```
. quietly: summarize lwt
. generate lwts = lwt - r(mean)
```

Note that if one intended to standardize the variable `lwt` (that is not only to subtract the average to each observation but also to normalize by the standard deviation), the expression above should be replaced by:

```
. generate lwts = (lwt - r(mean)) / r(sd)
```

By making use of the commands egen, the same result could be accomplished with a command such as:

```
. egen lwts = std(lwt)
```

Finally, the regression model, including the mothers' weight centered on their average and the frequency of visits to the gynecologist in the first trimester of pregnancy, would be written as:

```
. regress bwt lwts ftv i.race, noheader

------------------------------------------------------------------------------
         bwt |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        lwts |   10.14028   3.891895     2.61   0.010     2.461807    17.81876
         ftv |   11.82137   49.17071     0.24   0.810    -85.18953    108.8323
             |
        race |
          2  |  -450.1195   158.1307    -2.85   0.005    -762.1019   -138.1371
          3  |  -239.2497   114.4963    -2.09   0.038    -465.1442   -13.35526
             |
       _cons |    3081.94   83.91919    36.73   0.000     2916.372    3247.507
------------------------------------------------------------------------------
```

Note that the operator i. has been included to inform Stata to manipulate the variable race as a categorical variable, as indicated in Chapter 2 (section 2.4).

3.4. Key points

– The correlate and spearman commands enable the calculation of the Pearson and the Spearman correlation coefficients. A scatter diagram can be constructed with twoway scatter in order to visualize the shape of the cloud of points.

– The regress command is employed in the context of the linear regression and follows the same notation as in the case of the ANOVA (response variable followed by the explanatory variables).

– A number of postestimation commands allow for information to be obtained about the goodness of fit of the model (fitstat). These commands also make it possible to calculate the fitted values or to predict the expected values for new observations (predict).

3.5. Further reading

As in Chapter 2, Mitchell's book [MIT 12] provides numerous graphic illustrations (command `marginsplot`) applied to the case of the linear model. The Stata applications proposed by Vittinghoff *et al.* [VIT 05] contribute to deepening the key concepts in the construction and the interpretation of a multiple regression model.

3.6. Applications

1) In [EVE 01], the authors focused on measuring malnutrition among 25 patients aged from 7 to 23 years and suffering from cystic fibrosis. Information of the anthropometric characteristics and pulmonary function of these patients was included in the calculations (height, weight, etc.). The data are available in the file `cystic.dat`.

– calculate the linear correlation coefficient between the variables `PEmax` and `Weight`, as well as its 95% confidence interval;

– display the totality of the numerical data in the form of scatterplots, that is 45 graphs arranged in the form of a "dispersion matrix";

– calculate all of the Pearson and Spearman correlations between the numerical variables;

– calculate the correlation between `PEmax` and `Weight`, controlling the age (`Age`) (partial correlation). Graphically represent the covariation between `PEmax` and `Weight` highlighting the two most extreme terciles for the variable `Age`.

Tabulated data are available in a text file that can be imported by entering the command `insheet`. However, it is also possible to import the data with `infile`, which has the advantage of operating even when the data do not originate from an Excel-like table:

```
. infile int (Sub Age Sex) Height Weight BMP FEV RV FRC ///
  TLC PEmax using "cystic.dat" in 2/26, clear
```

In the above instruction, it is explicitly required that Stata encode the first three variables as integers and not as real numbers, which does not currently improve much but makes it possible to save memory space for large data sets. As was done with R, the variable `Sex` will be recoded in the first instance into a categorical variable to avoid any confusion:

```
. label define labsex 0 "M" 1 "F"
. label values Sex labsex
. tabulate Sex
```

The estimation of the Pearson correlation coefficient is carried out with the command `correlate`, but this does not provide any options for the confidence interval. External commands such as `ci2` can be employed that work on the same principle as the internal command `ci` (for averages and proportions) in the parametric or non-parametric case (Spearman) or either `corrci`, only in the parametric case:

```
. correlate PEmax Weight
. corrci PEmax Weight
```

To test this correlation coefficient against the hypothesis $H_0 : \rho = 0$, the command `pwcorr` can be entered with the option `sig`. In the case of several numerical variables, this command also allows the correlation matrix to be calculated as will be mentioned hereafter:

```
. pwcorr PEmax Weight, sig
```

To display the totality of the scatterplots, the `graph matrix` command is employed specifying the list of variables that we would like to present in the chart. Here, the variable `sex` will be omitted:

```
. graph matrix Age Height-PEmax
```

The correlations for each pair of variables are obtained with the `pwcorr` command, and the same notation will be used to indicate the variables that interest us:

```
. pwcorr Age Height-PEmax
```

Concerning Spearman correlations, `pwcorr` will be replaced by `spearman`:

```
. spearman Age Height-PEmax, stats(rho)
```

The partial correlation between the `PEmax` and `Weight` variables taking `Age` into account is obtained by means of the command `pcorr`; the variable of interest must be located in the first position in the list of variables and by default Stata displays all partial correlation coefficients for the other variables:

```
. pcorr PEmax Weight Age
```

In order to compute terciles of the `Age` variable, an auxiliary variable can be created with `egen` and the function `cut`. But we can directly utilize the `xtile` command whose usage is easier:

```
. xtile Age3 = Age, nq(3)
```

Finally, for scatterplots, here is a first solution:

```
. scatter PEmax Weight if Age3 != 2, mlab(Age3)
```

We could also (second solution) perform two calls to the `scatter` command, one following the other, each time restricting the sample to the sole classes of `Age3` that are of interest (1 and 3, in this case):

```
. scatter PEmax Weight if Age3 == 1, msymbol(circle) || scatter PEmax ///
  Weight if Age3 == 3, msymbol(square) ///
  legend(label(1 "1st tercile") label(2 "3rd tercile"))
```

2) In the Framingham study, we can access data about systolic blood pressure (`sbp`) and about the body mass index (`bmi`) of 2,047 men and 2,643 women [DUP 09]. The main interest lies in the relationship between these two variables (after logarithmic transformation) in men and women separately. The data are available in the `Framingham.csv` file.

– graphically represent the variations between blood pressure and BMI (`bmi`) in men and women;

– are the linear correlation coefficients estimated for men and women significantly different at 5%?

– estimate the parameters of the linear regression model considering the blood pressure as a response variable and the BMI as an explanatory variable, for these two subsamples. Give a 95% confidence interval for the estimation of the respective slopes.

Since the data are available in "conventional" CSV (using the comma as field separator), they can be very easily imported by entering the `insheet` command:

```
. insheet using "Framingham.csv"
. describe, simple
. list in 1/5
```

To facilitate the interpretation and reading of the tables of results, more informative labels will be associated with the categorical variable `sex`:

```
. label define labsex 1 "M" 2 "F"
. label values sex labsex
. tabulate sex
```

To display a summary of the number of missing data for each of the variables in this data table, we can use the following command:

```
. misstable summarize
```

It can thus be seen that the `scl` and `bmi` variables include 33 and 9 missing values, respectively. Hence, the corrected table of the counts by sex:

```
. tabulate sex if bmi < .
```

In Stata, it is not possible to display transparent markers in a scatterplot. In the case where the number of observations is high, it is therefore best to change the default symbol type (filled circle) and display small circles (`oh` or `Oh`):

```
. scatter sbp bmi, by(sex) msymbol(Oh)
```

The linear correlation coefficient between the variables `sbp` and `bmi` according to sex can be obtained by means of `correlate` after stratification upon the factor `sex`. It should be noted that the option `by` appears in first place and that it is necessary to sort the data as a first step, and the addition of the command `sort` immediately after stratification:

```
. by sex, sort: correlate sbp bmi
```

For the regression model, we will first transform the `bmi` (explanatory variable) and `sbp` (response variable) variables by making use of a logarithmic transformation:

```
. gen logbmi = log(bmi)
. gen logsbp = log(sbp)
```

It is then possible to visually verify through a histogram (or a QQ-plot) that this transformation has been able to correctly reduce the distributions of these two variables close to the normal. In order to simultaneously display the four histograms, each of the figures can be saved in the Stata format (gph) and then be combined with the `graph combine` command:

```
. histogram bmi, saving(gphbmi)
. histogram logbmi, saving(gphlogbmi)
. histogram sbp, saving(gphsbp)
. histogram logsbp, saving(gphlogsbp)
. graph combine gphbmi.gph gphlogbmi.gph gphsbp.gph gphlogsbp.gph
```

The regression model stratified by gender poses no major problem, and unlike R, it is not necessary to calculate the confidence intervals for the slopes with a separated command because they are directly provided in the table of results returned by Stata:

```
. regress logsbp logbmi if sex == 1, noheader
. regress logsbp logbmi if sex == 2, noheader
```

3) The data available in the `quetelet.csv` file provide information on systolic blood pressure (`PAS`), the Quetelet index (`QTT`), age (`AGE`) and tobacco consumption (`TAB` $= 1$ if smoking, 0 otherwise) for a sample of 32 men over the age of 40.

– indicate the value of the linear correlation coefficient between systolic blood pressure and the Quetelet index, with a 90% confidence interval;

– give estimates of the parameters of the regression line of the systolic blood pressure on the Quetelet index;

– test if the slope of the regression line is different from 0 (considering a 5% Type I error);

– graphically represent the variations in blood pressure with respect to the Quetelet index, distinctly showing smokers and non-smokers with symbols or different colors, and draw the regression line whose parameters have been estimated in the preceding regression model;

– repeat the previous analysis by restricting the sample to smokers.

Loading the data raises no specific difficulties because these have been exported from Excel and are in CSV format. However, care should be taken to ensure that the field delimiter type be specified, here a semicolon:

```
. insheet using "quetelet.csv", delim(";") clear
. describe
```

After importing, it can be observed that the variable `qtt` is not recognized as a number as such but has been coded as a string of characters. This is explained by the fact that the fractional part is separated from the integer portion by a comma and not a dot. It is therefore necessary to convert this variable into a number, which can be achieved with the `destring` command and the `dpcomma` option:

```
. destring qtt, dpcomma replace
```

Next, the variable `tab` is recoded into a qualitative variable including more informative labels:

```
. label define ltab 0 "NF" 1 "F"
. label values tab ltab
. list in 1/5
```

Finally, the numerical summary of the variable is obtained with the `summarize` command:

```
. summarize pas-tab
```

Since the `tab` variable is internally coded as $0/1$, the average corresponds to the relative frequency of smokers, that is 53%.

The linear correlation coefficient between the `pas` and `qtt` variables can be estimated, as well as its confidence interval, with the `corrci` command. The `level(90)` option enables the confidence level to be specified:

```
. corrci pas qtt, level(90)
```

The `pwcorr` command will be utilized whenever it is desirable to test whether the correlation is zero.

The `regress` command makes it possible to perform a linear regression for a response variable (placed first in the list of variables) and one or more explanatory variables. It is used as follows:

```
. regress pas qtt
```

By default, a table of the ANOVA for the regression model and a table of the model coefficients table are obtained, here the *y*-intercept (70.58) and the slope of the regression line (21.49). The Student's test associated with the slope allows its statistical significance to be evaluated in the light of the data. When the regression coefficients have to be manipulated, it is possible to extract them from the results table as shown here:

```
. matrix b = e(b)
. svmat b
. display "slope = " b1 ", y-intercept = " b2
```

With regard to displaying the scatterplot with both regression lines superimposed, we utilize a combination of `lfit` (to draw the regression line) and `scatter` (to display the observations). The disadvantage is that it is necessary to manually build the legend. Note that it is required that the regression lines be presented over the whole range of the `qtt` variable, hence the `range(2, 5)` option:

```
. twoway lfit pas qtt if tab == 0, range(2 5) lpattern(dot) ||
  scatter pas qtt if tab == 0, msymbol(square) ||
  lfit pas qtt if tab == 1, range(2 5) ||
  scatter pas qtt if tab == 1, msymbol(circle)
  legend(label(1 "") label(2 "NF") label(3 "") label(4 "F"))
```

The regression of `qtt` on `pas` restricting the analysis to the sole observations of the group smoker (`tab == 1`) does not raise concern because as it has been seen in the previous application, it suffices to add an `if tab == 1` option:

```
. regress pas qtt if tab == 1

      Source |       SS       df       MS              Number of obs =      17
-------------+------------------------------           F(  1,     15) =   19.40
       Model |  2088.16977     1  2088.16977           Prob > F      =  0.0005
    Residual |  1614.30082    15  107.620055           R-squared     =  0.5640
-------------+------------------------------           Adj R-squared =  0.5349
       Total |  3702.47059    16  231.404412           Root MSE      =  10.374

------------------------------------------------------------------------------
         pas |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
```

```
         qtt |   20.11804   4.567193     4.40   0.001      10.3833    29.85278
       _cons |   79.25533   15.76837     5.03   0.000     45.64585    112.8648
------------------------------------------------------------------------------
```

4) Based on the birth weight data [HOS 89], the aim is to study the relationship between the babies' weight (considered as a numeric variable, `bwt`) and two characteristics of the mother: her weight (`lwt`) and her ethnic origin (`race`).

– graphically represent the relationship between the babies' weights and the mothers' weights, according to the ethnicity of the mothers;

– estimate the linear regression parameters considering the baby's weights as a response variable and the mothers' weights centered on their average as an explanatory variable. Is the estimated slope significant at the usual 5% threshold?

– estimate the parameters of the linear regression, where this time the explanatory variable is the mothers' ethnicity, the response variable remaining the babies' weight. Compare the significance of the model in its whole with the results obtained from one-way ANOVA (ethnicity);

– what is the weight predicted for a baby whose mother weighs 60 kg?

– give a 95% confidence interval for a pointwise prediction in average.

Importing data in text format would pose no particular difficulties, but in order to simplify the task we will use the data available online. The dataset on birth weight can be imported by typing `webuse` as follows:

```
. webuse lbw
```

We will verify that the variables are correctly the same as those found in the `birthwt` file in R.

To represent the relationship between the babies' and the mothers' weight taking their ethnicity into account, a simple scatter plot can be constructed with the following instructions:

```
. sepscatter bwt lwt, separate(race)
```

Here, we are employing a command downloadable from the SSC site (`findit sepscatter`) to print out multiple scatterplots in the same graphic using a third conditioning variable. This command avoids combining multiple `scatter` statements, separated by `||`.

For the regression model, an auxiliary variable `lwtc` has been initially generated, representing the mothers' weight centered on their average, the latter being obtained from `summarize`. Then, it suffices that the response variable and the explanatory variable be indicated to the `regress` command as follows:

```
. quietly: summarize lwt
. gen lwtc = lwt/r(mean)
. regress bwt lwtc
```

According to the results, a significant relationship between the two variables is found thereof.

Considering the `race` variable as an explanatory variable, it is necessary to prefix it with the operator `i.` so that Stata properly manipulate the `race` variable as a categorical variable:

```
. regress bwt i.race
```

The reference category is the first level of `race` in order for the regression coefficients printed out to correspond to the mean deviations with respect to the average babies' weight in the "white" category. In comparison, an ANOVA model would give the following results (here, there is no need to specify `i.race`):

```
. oneway bwt race
```

It can be verified that the previous command (or alternatively, simply `regress bwt i.race, notable`) properly yields the same result for the test of all of the regression model.

The predicted value for a child's weight whose mother weighs 60 kg can be obtained from the coefficients of the regression line or by utilizing the `margins` command, which is more flexible to use. In the first case, the postestimation values returned by the `regress` command will be employed (accessible via `ereturn list`), that is:

```
. replace lwt = lwt/2.2
. regress bwt lwt
. display _b[_cons] + _b[lwt]*40
```

In the second case, the instruction to be inserted amounts to:

```
. margins, at(lwt=40)
```

A 95% confidence interval is automatically provided by `margins`. The `level(90)` option would provide a 90% confidence interval.