SAS 6.1. Correlation and regression

1. Correlation

Example: Calculate correlations of several variables.

```
DATA CORR_EG;
      INPUT GENDER $ HEIGHT WEIGHT AGE;
DATALINES;
M  68    155   23
F  61    99    20
F  63    115   21
M  70    205   45
M  69    170   .
F  65    125   30
M  72    220   48
;
PROC CORR DATA=CORR_EG;
      TITLE 'EXAMPLE OF A CORRELATION MATRIX';
      VAR HEIGHT WEIGHT AGE;
RUN;
```

*0.0003 is the p value this is for linear correlation

Each time PROC CORR prints a correlation coefficient, it also prints a probability associated with the coefficient (i.e., p-value), which gives the probability of obtaining a sample correlation coefficient as large as or larger than the one obtained by chance alone.

The significance of a correlation coefficient is a function of the magnitude of the correlation and the sample size. With a large number of data points, even a small correlation coefficient can be significant.

It is important to remember that correlation indicates only the strength of a relationship – it does not imply causality. For example, we would probably find a high positive correlation between the number of hospital in each of the 50 states versus the number of household pets in each state. It does not mean that pets make people sick and therefore make more hospitals necessary! The most plausible explanation is that both variables (number of pets and number of hospitals) are related to population size.

An important assumption concerning a correlation coefficient is that each pair of x.y data points is independent of any other pair. That is, each pair of points has to come from a separate subject. Otherwise we can not compute a valid correlation coefficient.

2. Partial correlations

A researcher may wish to determine the strength of the relationship between two variables when the effect of other variables has been removed.

To remove the effect of one or more variables from a correlation, use a PARTIAL statement to list those variables whose effects you want to remove

```
PROC CORR DATA=CORR_EG nosimple;
      TITLE 'EXAMPLE OF A partial
      CORRELATION'; VAR HEIGHT WEIGHT;
      Partial age;
RUN;
```

As you can see in the following listing, the partial correlation between height and weight is now lower than before (0.91934), although it is still significant (p=0.0272)

```
                 Pearson Partial Correlation Coefficients, N = 6
                        Prob > |r| under H0: Partial Rho=0

                        HEIGHT                        WEIGHT

            1.00000                       0.91934
  HEIGHT
                                          0.0272

            0.91934                       1.00000
  WEIGHT
            0.0272
```

3. Linear regression

(1) We run linear regression for the dataset displayed in previous subsection first, and then discuss the output.

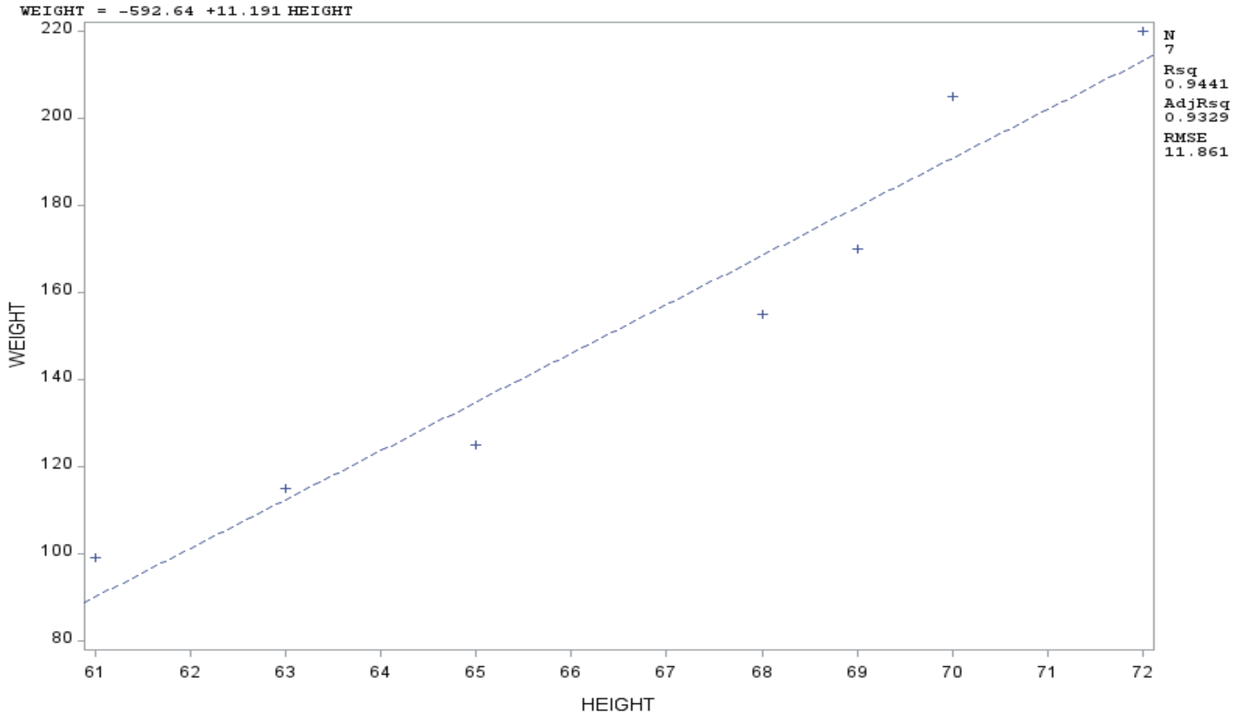
```
PROC REG DATA=CORR_EG;
      TITLE 'REGRESSON LINE FOR HEIGHT-WEIGHT DATA';
      MODEL WEIGHT=HEIGHT;
RUN;
```

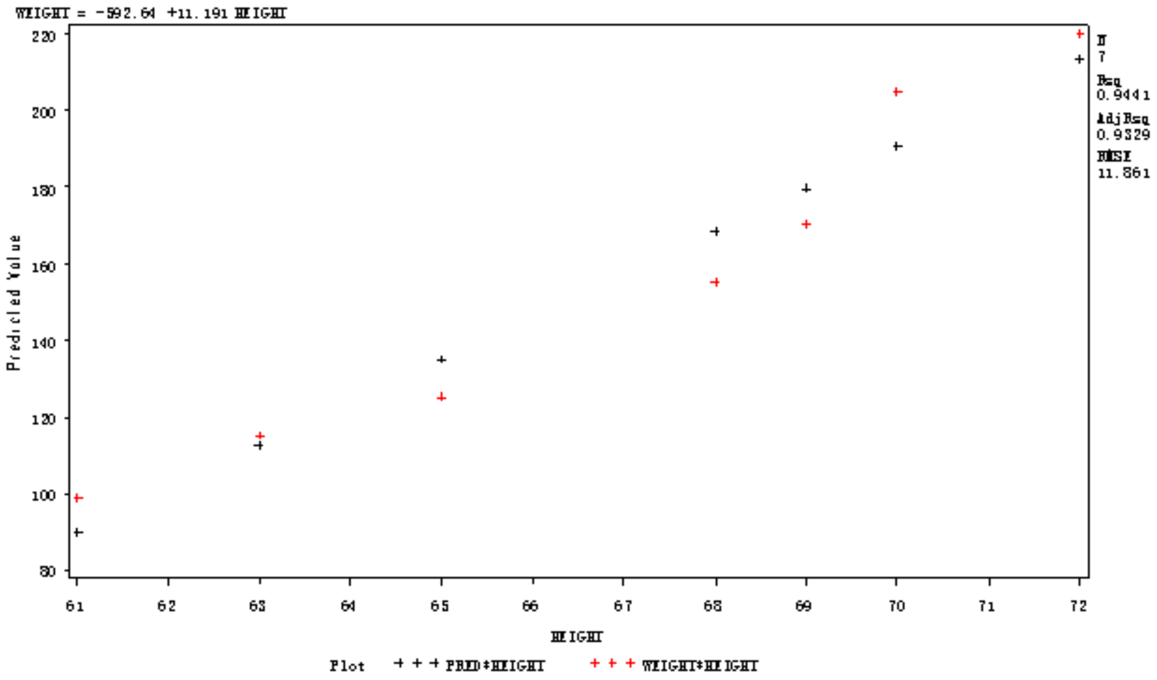
(2) We also plot the points on the regression line. In particular, the statements PREDICTED and RESIDUAL are used to plot predicted and residual values. A common option is OVERLAY, which is used to plot more than one graph on a single set of axes.

```
PROC REG DATA=CORR_EG;
MODEL WEIGHT=HEIGHT;
PLOT WEIGHT*HEIGHT;
RUN;

PROC REG DATA= CORR_EG;
      MODEL WEIGHT=HEIGHT;
      PLOT PREDICTED.*HEIGHT        WEIGHT*HEIGHT/ OVERLAY;
RUN;
```

REGRESSON LINE FOR HEIGHT-WEIGHT DATA





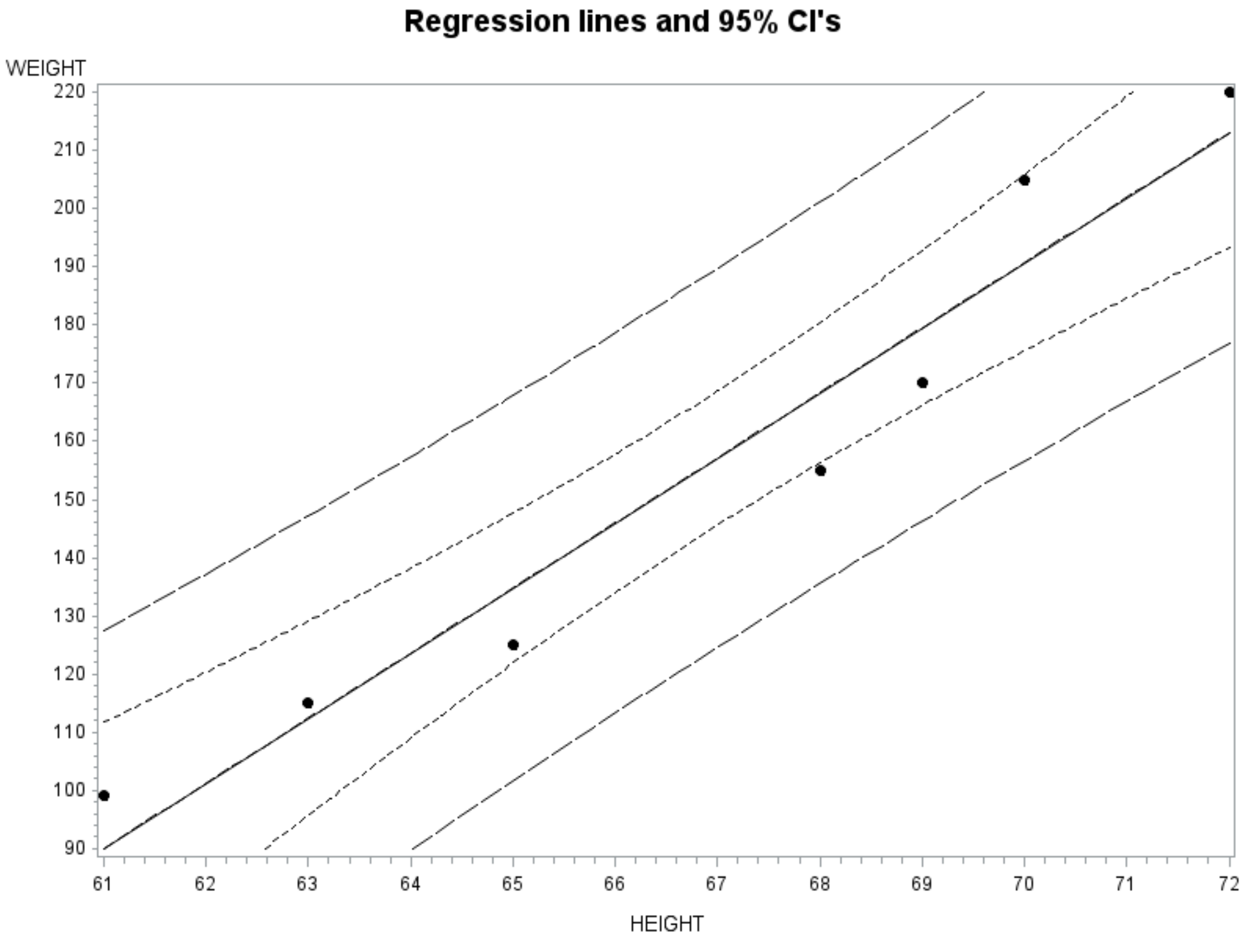
Predicted. And residual. (the periods are part of these keywords) are used to plot predicted and residual values.

(3) We could also plot the residuals and confidence limits. The most useful statistics besides the predicted values includes: RESIDUAL, L95, U95, L95M, U95M.

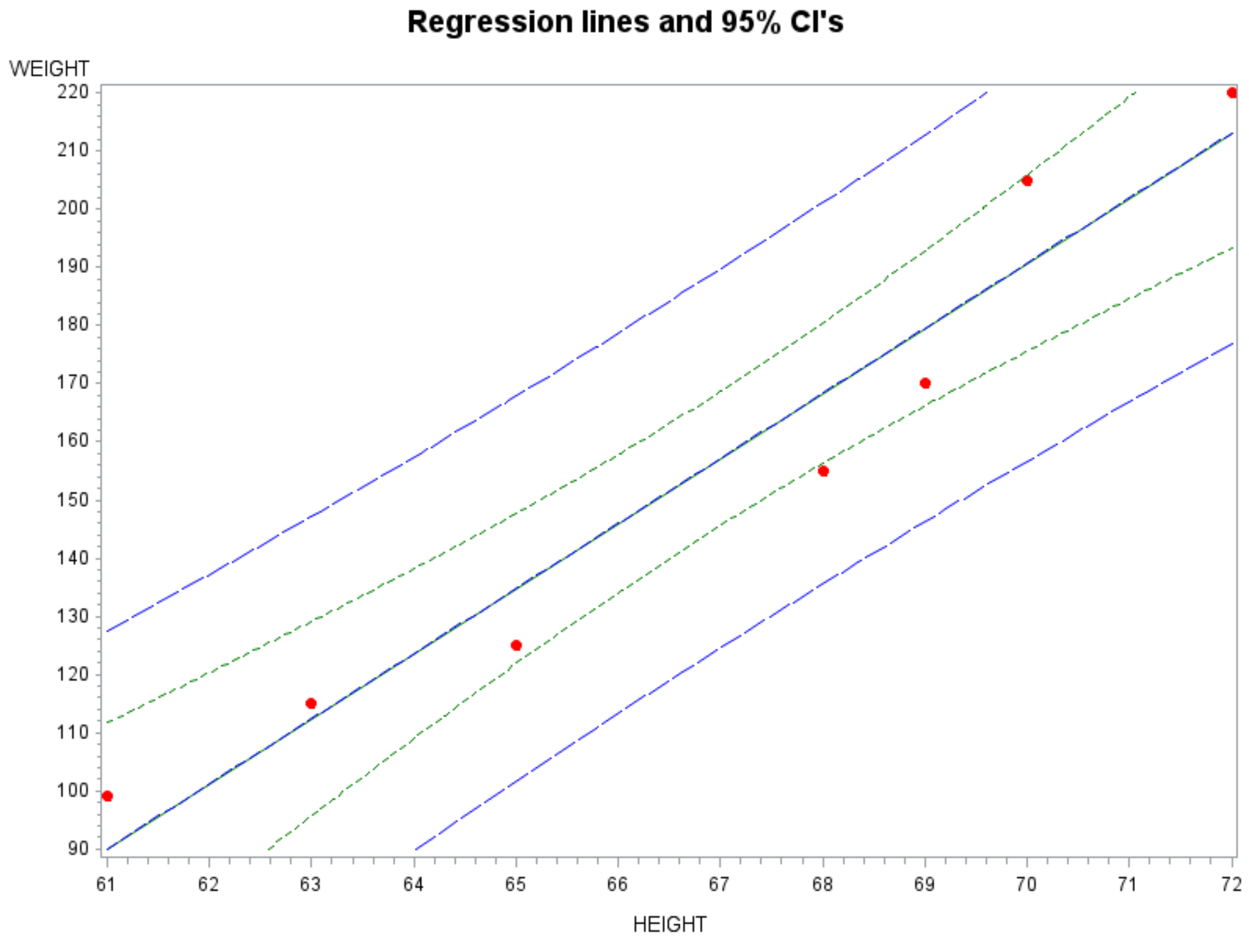
```
GOPTIONS
CSYMBOL=BLACK;
SYMBOL1 VALUE=DOT;
SYMBOL2 VALUE=NONE I=RLCLM95;
SYMBOL3 VALUE=NONE I=RLCLI95 LINE=3;
PROC GPLOT DATA=CORR_EG;
       TITLE "Regression lines and 95% CI's";
       PLOT WEIGHT*HEIGHT =1
                    WEIGHT*HEIGHT =2
                    WEIGHT*HEIGHT =3 / OVERLAY;
RUN;
```

*Note the use of the csymbol option on the goptions statement to set the color to black for all the plotting symbols.
the first symbol statement requests dots as the plotting symbols
the second and third symbol statements set value equal to “none”, which hides the points, the I(interpolation) option in symbol2 requests a regression line and the 95% CI about the mean of y(rlclm95),
the symbol3 statement asks for the confidence interval about the individual y-value(cli95), in addition, this symbol statement uses line=3 to generate a different type of line for this confidence interval
the plot statement is requesting the overlayed plots, and the =1 =2 and =3 following each of the plot requests indicates which symbol statement to use for each of the plots

*in the following plot, notice that the 95% CI about the mean of y is the narrower band(small dashed line), and the 95% CI for individual y-values is represented by the wider larger-dashed lines;



```
SYMBOL1 VALUE=DOT color=red;
SYMBOL2 VALUE=NONE I=RLCLM95 color=green;
SYMBOL3 VALUE=NONE I=RLCLI95 LINE=3 color=blue;
PROC GPLOT DATA=CORR_EG;
TITLE "Regression lines and 95% CI's";
PLOT WEIGHT*HEIGHT =1
WEIGHT*HEIGHT =2
WEIGHT*HEIGHT =3 / OVERLAY;
RUN;
```



(4) We could add a quadratic term to the regression equation as follows. For example, after the INPUT statement, we can include a line such as the

following: HEIGHT2 = HEIGHT * HEIGHT;

or

HEIGHT2 = HEIGHT**2;

Then, we should write MODEL and PLOT statements.

```
DATA CORR_EG;
        SET CORR_EG;
        HEIGHT2=HEIGHT**2
        ;
RUN;
```

```
PROC REG DATA=CORR_EG;
         MODEL WEIGHT = HEIGHT HEIGHT2;
         PLOT R.*HEIGHT;
RUN;
```

*R. is short for residual.

When you run this model, some improvement is achieved comparing to the original one. Rsquare is 0.9743, an improvement over the 0.9441 obtained with the linear model.

(5) Sometimes, we need to transform data before running regressions. In the following example, either by clinical judgment or by careful inspection of the graph, we decide that the relationship is not linear.

```
DATA HEART;
   INPUT DRUG_DOSE HEART_RATE;
DATALINES;
2  60
2  58
4  63
4  62
8  67
8  65
16 70
16 70
32 74
32 73
;


PROC GPLOT DATA=HEART;
   PLOT HEART_RATE*DRUG_DOSE;
RUN;
PROC REG DATA=HEART;
   MODEL HEART_RATE=DRUG_DOSE;
RUN;
```

```
Symbol value =dot color=black I=sm;
 PROC GPLOT DATA=HEART;
    PLOT HEART_RATE*DRUG_DOSE;
RUN;
*I=SM produces a smooth line through the data points
```

We see an approximately equal increase in heart rate each time the dose is doubled. Therefore, if we plot log dose against heart rate we can expect a linear relationship. We then add mathematical equations to define new variables by placing these statements between the INPUT and DATALINES statements.

```
   DATA HEART_LOG;
   SET HEART;
   L_DRUG_DOSE = LOG(DRUG_DOSE);
RUN;
```

```
PROC GPLOT DATA=HEART_LOG;
   PLOT HEART_RATE*L_DRUG_DOSE;
RUN;
QUIT;

PROC REG DATA=HEART_LOG;
   MODEL HEART_RATE=L_DRUG_DOSE;
   PLOT R.*HEART_RATE;
RUN;
```

Notice that the data points are now closer to the regression line. The MEAN SQUARE ERROR term is smaller and r-square is large, confirming our conclusion that dose versus heart rate fits a logarithmic curve better than a linear one.

*Some variables are frequently transformed: Income, sizes of groups, and magnitudes of earthquakes are usually presented as logs or in some other transformation