Linear regression with LaTeX

Battista Benciolini

June 10, 2024

Contents

Linear regression with LaTeX - available in CTAN
Battista Benciolini - contact: benciolinibattista at gmail dot com

1 Introduction: first description of the problem

I start with a quote from ArsTEXnica (April 2021, number 31, page 73):

> The physicist Mario Rossi is investigating a phenomenon, presumably linear, and he performs measurements in his laboratory to verify his hypothesis; he measures the quantity x which generates the phenomenon and he measures also one of the characteristics y showed by the phenomenon under the effect of the stimulation x.
>
> ...
>
> Subsequently Mario graphs the data of the table to judge if the points reasonably follow a linear trend or not; in this regard he computes the parameters of the regression line and he draws this line on the graph in order to judge the quality of the obtained results.
>
> ...
>
> Being a LaTeX user, he thinks to kill two birds with one stone: using LaTeX to draw the graph with the experimental data consisting in the x, y points and, at the same time, to compute the parameter a e b of the regression line $y = ax + b$, and finally to draw also this line on the same graph.

A summary description of the the problem is therefore the following. A set of data pairs is available and each pair is represented as a point in the plain. A straight line is searched that optimally approximates the points. The first step is therefore the choice of an optimality criterion. This choice is the topic of the next section. From the text we also know that the possible deviation of y with respect to the model is quite larger than the uncertainty of x.
After reading the description of the problem of Mario Rossi I tried to produce a solution. In this work I will use y_1 and y_2 instead of x and y for the two measured quantities that will become the first and second coordinate, or abscissa and ordinate, in the Cartesian plane.
The problem can be treated as a mere problem of approximation or alternatively as an estimation problem in the frame of a probabilistic description of the uncertainty. The two treatments are conceptually different. The probabilistic treatment produces some more results, but the estimation of the parameters is the same. On the other hand the treatment as an approximation problem is in some sense more immediate and requires a less extended theoretical background. For this reason it will be preferred here. I consider the original problem and also a variation of it based on the assumption that the two variables are known with the same uncertainty. The two considered situations will prove to be quite different.

2 Geometric definition of there optimality criteria

For each point given in the plane we can consider the corresponding point with the same abscissa and belonging to the line. Remember that the line is exactly what has to be determined. The distance between the given point and the just defined point on the line is a reasonable measure of the discrepancy between the empirical data and the corresponding theoretical model. The distances we are speaking about are the length of the segments shown in the leftmost scheme of figure (1).

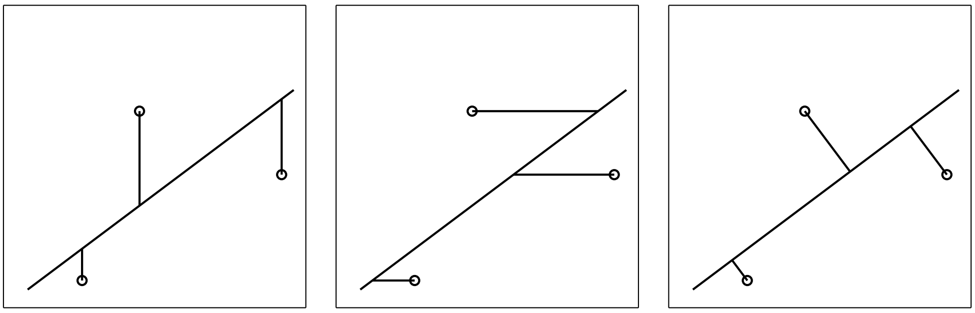

Figure 1: The three kinds of segments used in the definition of the objective function

To obtain a global discrepancy measure that considers all the points at once we perform the sum of the squares of the lengths of the mentioned segments. It is now clear that the two coordinates of the points are treated quite differently and play a different role in the definition of the optimality criterion. This choice is reasonable when the measuring errors only (or mainly) affect the second coordinate. The optimal line is the line that minimize the just defined global discrepancy. The procedure for the determination of the optimal line is named linear regression. In this work it is named *classical linear regression.* We can easily exchange the role of the two quantities, i.e. we can imagine that the first quantity is affected by errors. The problem is not conceptually different. The segments plotted in the central picture of figure (1) represent the discrepancy between the empirical data and the model. This other procedure is named *classical linear regression with inverted role of the coordinates.*

The situation is really different if the two coordinates have to be treated symmetrically. In this case the discrepancy between the empirical data and the model must be defined in a purely geometrical way. Just the line and the points enter in the definition without any special role for any predefined direction. With these requirements it is quite natural to use the distance of each point from the line. Remember that the distance of a point from a line is intended along the shortest path, i.e. measured in the direction orthogonal to the line itself. The rightmost scheme of figure (1) shows the segments that are considered. The global measure of discrepancy is again obtained as the sum of the squares of the length of the mentioned orthogonal segments. The procedure that obtain the optimal line that minimize the just defined global discrepancy is named *symmetrical linear regression.*

Some arguments of the present section will be repeated in section 8 from the algebraic and computational point of view.

3 General information on the proposed solution, including limitations

The code that implements the solution is recorded in two files, that are a package (sty) file and a main interactive document. The file `linearregression.sty` provides several commands that can be used in any document. The file `mainlinearregression.tex` provides a simple interactive user interface. The package described in the sections 6 and 8 (user manual and implementation) provides the functions that execute the various needed operations, i.e. data input, computations, printing the numerical results and generating a graphic representation of data and results. Some auxiliary functions complete the package. The design of the output (tables and plots) includes some arbitrary choices. The style of the graphic output is quite minimalist (e.g.: no colors, no variations of line styles).

4 Some comments about the programming aspect of the package and its documentation

Large part of the code is written using the `expl3` language. (Is it also named simply L3 ? Does expl still means experimental ?) I have tried to be compliant with the various recommendations and prescriptions for a correct use of the language, but I probably only partly succeeded.
Different more elegant and more coherent solutions probably exist both for the general structure of the package and for some specific part of the code, but this is what I have been able to do. Some perhaps problematic aspects are mentioned here after
Several used variables are global and they are accessed by various functions. This makes the various parts of the package quite connected to each other and creates strong dependencies.
The layered programming style is only partially applied. The partition between document command and lower level functions is present, but part of the low level code is directly in the document commands. Variants are not used.
One more remarks concern the documentation. I was uncertain about the opportunity of using the class `l3doc`. I decided to remain using `ltxdoc`. This is the reason why I do not use the environment `macro` and the command `\cs` in the documentation of some auxiliary functions named according with the `expl3` standard. (I have just an interim far from optimal solution for a reasonable formatting.)

5 A ready to use simple user interface

The main file asks the user for the name of a file containing the data and generates a one (or two) page output.

```
\documentclass[a4paper]{article}
\usepackage{lmodern}
\usepackage{linearregression}
\begin{document}
\pagestyle{empty}
\lraskfilename
\lrcomputation
\lrplot{12.0}{+}{+}{-}{-}
\lrprint
\end{document}
```

6 A user manual for the package

The various analysis of a data set and the representation of the data and of the results is obtained with a sequence of several commands. The main operations are: (i) selection of the data file, (ii) data imput and computation, (iii) printing of a table, (iv) printing of a picture (that can be repeated with different parameters). It is generally convenient to put the table and the picture(s) in a proper floating environment. The commands for the four mentioned operations are described here after. The first needed operation is to set the name of the data file.

`\lrfilename` This is done with the command `\lrfilename{`⟨*file*⟩`}` that has a mandatory argument. The argument is the name of the data file. As an alternative the command

`\lraskfilename` `\lraskfilename` can be used. It asks the user to type the name of the data file in the terminal.

`\lrcomputation` The macro `\lrcomputation` reads the data and performs all the computations. The results of the computations remain available in internal variables and are then used by the macro that print them or generates a plot.

`\lrprint` The macro `\lrprint` generates a table with all the estimated parameters and some information about the data.

`\lrplot` The macro `\lrplot{`⟨*imagewidth*⟩`}{`⟨*key1*⟩`}{`⟨*key2*⟩`}{`⟨*key3*⟩`}{`⟨*key4*⟩`}` really generates the plot. The first argument is the width of the plot, while the height is computed according to the distribution of the points. The other four arguments are referred to the data points, to the lines determined with classical regression, with classical regression with inverted role of the coordinates and with symmetric regression. The four items, i.e. the set of points and the three lines, are drawn or not according to the corresponding character found in `key`*i*. Each item is not plotted if the character is a `-`, it is plotted in any other case. Furthermore the lines are accompanied by a label made by the corresponding `key`, unless it is just a `+`.

Few words are necessary about the format of the data file. Each record of the file hold the two values related to a point. The two values must be separated by any number (one is needed as a minimum) of space and comma characters. No character different from space can be accepted before the first value and after the second value.

7 An example

The data reported here after will be available in `sampledata.txt` and will be used in the example presented in this section .

11	-0.546	0.107	17	-0.203	-0.292	23	0.181	-2.616	29	-0.931	-1.613
12	1.093	-0.510	18	1.517	0.779	24	0.619	1.859	30	-1.070	0.592
13	1.440	1.995	19	0.559	-1.341	25	-0.223	-1.915	31	2.341	0.413
14	1.414	0.991	20	-0.462	-0.437	26	0.629	-0.534	32	1.993	-0.111
15	0.735	1.585	21	-0.785	-0.661	27	-1.989	-2.300	33	-2.357	-0.312
16	-1.848	-0.235	22	-0.558	0.397	28	-0.241	1.098	34	-1.975	0.140

The analysis of the sample data and the generation of a numeric table is operated by a code similar to the following (see table 1).

```
\lrfilename{sampledata.txt}
\lrcomputation
\begin{table}
  \lrprint
  \caption{Analysis of ... }
\label{tab:sampledata}\end{table}
```

Data File:	sampledata.txt
Number of points:	24
Mean values of the coordinates:	-0.02779166666666667 -0.1217083333333333
Minimum values of the coordinates:	-2.357 -2.616
Maximum values of the coordinates:	2.341 1.995
Second order moments abscissa mixed ordinate	1.628921831597222 0.5549238975694446 1.446702873263888
Slope and intercept of optimal line (estimated with errors in ordinate)	0.3406694457678917 -0.112240561653034
Slope and intercept of optimal line (estimated with errors in abscissa)	2.607029323480962 -0.04925464338492489
Components of unit vector along the line Slope and intercept of optimal line (estimated with symmetric regression)	0.7622383487354221 0.6472964542750848 0.8492047865985363 -0.09810751697244898

Table 1: Analysis of the sample data

The generation of some different graphical representation of the data and of the results is operated by a code similar to the following (see figures 2).

```
\begin{figure}
\lrplot{10.}{-}{AA}{BB}{S}
\lrplot{10.}{+}{-}{-}{+}
\caption{LEFT The three lines are obtained with the three optimality criteria. (AA) classical linear regression; (BB) classical linear regression with inverted role of the coordinates; (S) symmetric linear regression. RIGHT Data points and line estimated with symmetric linear regression.}
 \label{fig:sampledataB}  \end{figure}
```

8 A package for linear regression and the theory behind it

8.1 Math preliminaries and notation

The coordinates of a set of m points on the plane are available. A straight line is searched that optimally approximates the points.
The coordinates of a generic point are y_1 and y_2 and they are collected in the vector $\underline{y}$. Any given point is identified with the index i. (Explicit indices $(\ldots)_1$ or $(\ldots)_2$ always refer to the first or second coordinate of a point or to the first or second component of a vector in the plane. Symbolic index $(\ldots)i$ always refers to the different points. Few formulas require both indices $(\ldots)_{1i}$, $(\ldots)_{2i}$.)
With more then two points a criterion of best approximation is needed to select the optimal line that describes the data.

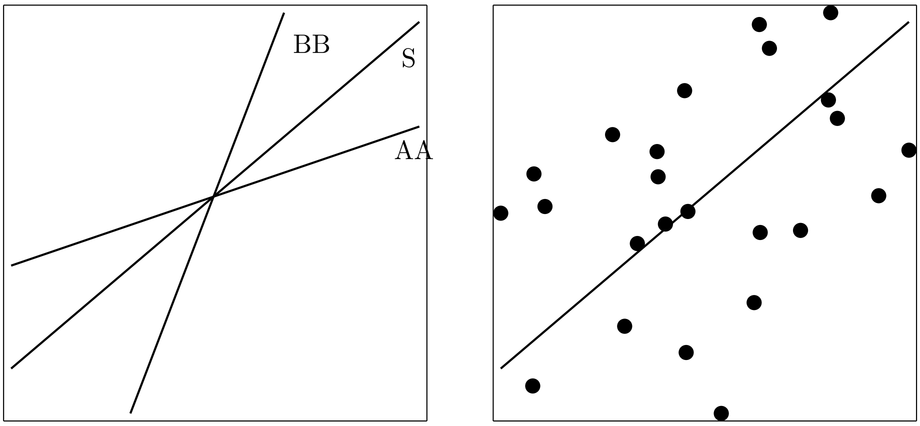

Figure 2: LEFT The three lines are obtained with the three optimality criteria. (AA) classical linear regression; (BB) classical linear regression with inverted role of the coordinates; (S) symmetric linear regression. RIGHT Data points and line estimated with symmetric linear regression.

Lower case symbols are used for scalars. Lower case underlined symbols are used for vectors in the plane. Upper case symbols are used for matrices.
It is possible that certain data generate an ambiguity or a singularity in the computation. The following mathematical treatment of the problem do not mention these situations and the code does not deal with them.

8.2 Package declaration, required package and definition of variables

The various macro will be provided in a package file that is introduced as usual. Most of the macros require the LaTeX3 syntax.

```
\ProvidesPackage{linearregression}[2024-06-10]
\RequirePackage{pict2e}
\ExplSyntaxOn
```

The variables used in the package are defined hereafter.

```
\ior_new:N  \g_BBLR_file_ior
\tl_new:N   \g_BBLR_file_name_tl
\int_new:N  \g_BBLR_number_of_points_int
\fp_new:N   \g_BBLR_abscissa_fp
\fp_new:N   \g_BBLR_ordinate_fp
\fp_new:N   \g_BBLR_mean_abscissa_fp
\fp_new:N   \g_BBLR_mean_ordinate_fp
\fp_new:N   \g_BBLR_abscissa_SecOrdMoment_fp
\fp_new:N   \g_BBLR_ordinate_SecOrdMoment_fp
\fp_new:N   \g_BBLR_mixed_SecOrdMoment_fp
\fp_new:N   \g_BBLR_slope_A_fp
\fp_new:N   \g_BBLR_slope_B_fp
\fp_new:N   \g_BBLR_slope_S_fp
\fp_new:N   \g_BBLR_intercept_A_fp
\fp_new:N   \g_BBLR_intercept_B_fp
```

```
\fp_new:N   \g_BBLR_intercept_S_fp
\fp_new:N   \g_BBLR_cos_fp
\fp_new:N   \g_BBLR_sin_fp
\fp_new:N   \g_BBLR_sig_sin_fp
\fp_new:N   \g_BBLR_eig_diff_fp
\fp_new:N   \g_BBLR_diag_diff_fp
\tl_new:N   \g_BBLR_file_line_tl
\fp_new:N   \g_BBLR_min_abscissa_fp
\fp_new:N   \g_BBLR_min_ordinate_fp
\fp_new:N   \g_BBLR_max_abscissa_fp
\fp_new:N   \g_BBLR_max_ordinate_fp
\fp_new:N   \g_BBLR_min_draw_abscissa_fp
\fp_new:N   \g_BBLR_max_draw_abscissa_fp
\bool_new:N \g_BBLR_data_eof_bool
\int_new:N  \g_BBLR_record_length_int
\int_new:N  \g_BBLR_rec_count_int
\int_new:N  \g_BBLR_first_separator_int
\int_new:N  \g_BBLR_last_separator_int
\str_const:Nn  \c_BBLR_space_str {~}
\str_const:Nn  \c_BBLR_comma_str {,}
\str_const:Nn  \c_BBLR_plus_str  {+}
\str_const:Nn  \c_BBLR_minus_str {-}
\bool_new:N  \g_BBLR_plot_points_bool
\bool_new:N  \g_BBLR_plot_lineA_bool
\bool_new:N  \g_BBLR_plot_lineB_bool
\bool_new:N  \g_BBLR_plot_lineS_bool
\fp_new:N   \g_BBLR_base_fp
\fp_new:N   \g_BBLR_height_fp
\fp_new:N   \g_BBLR_Xbase_fp
\fp_new:N   \g_BBLR_Xheight_fp
\fp_new:N   \g_BBLR_Dabscissa_fp
\fp_new:N   \g_BBLR_Dordinate_fp
\fp_new:N   \g_BBLR_diameter_fp
\fp_gset:Nn \g_BBLR_diameter_fp{0.2}
\fp_new:N   \g_BBLR_line_base_length_fp
\fp_new:N   \g_BBLR_scale_factor_fp
\str_new:N  \c_BBLR_point_code_str
\str_new:N  \g_BBLR_labelA_str
\str_new:N  \g_BBLR_labelB_str
\str_new:N  \g_BBLR_labelS_str
```

8.3 Preparing data input

`\lrfilename` The command `\lrfilename` records the file name passed as argument.

```
\NewDocumentCommand{\lrfilename}{m}{
\tl_gset:Nn \g_BBLR_file_name_tl {#1}
}
```

`\lraskfilename` The command `\lraskfilename` asks for the data file name from the terminal.

```
\NewDocumentCommand{\lraskfilename}{}{
\ior_get_term:nN  {filename ? } \g_BBLR_file_name_tl
\tl_trim_spaces:N \g_BBLR_file_name_tl
}
```

8.4 Main command declaration, computation of first and second order moments

`\lrcomputation` The command `\lrcomputation` reads the data file and performs all the relevant computations to solve the proposed problem.

```
\NewDocumentCommand{\lrcomputation}{}{%
```

In the sequel it will results that the first and second order moments of the data provide everything needed to solve the problem. The barycenter of the data is defined as

$$\underline{\overline{y}} = \frac{1}{m} \sum_{i=1}^{m} \underline{y}_i. \tag{1}$$

It is convenient to scan the data to accumulate the sum that appears in (1). The coordinates of each point are read from the file and they are immediately used. It is therefore not necessary to globally record the data.

```
\bool_gset_false:N \g_BBLR_data_eof_bool
\int_zero:N \g_BBLR_number_of_points_int
\fp_zero:N  \g_BBLR_mean_abscissa_fp
\fp_zero:N  \g_BBLR_mean_ordinate_fp
\ior_open:Nn \g_BBLR_file_ior \g_BBLR_file_name_tl
\bool_until_do:Nn \g_BBLR_data_eof_bool {
  \ior_str_get:NN \g_BBLR_file_ior \g_BBLR_file_line_tl
  \if_eof:w \g_BBLR_file_ior
    \bool_gset_true:N \g_BBLR_data_eof_bool
  \else:
    \int_incr:N \g_BBLR_number_of_points_int
    \BBLR_decode_data:
    \fp_gset:Nn \g_BBLR_mean_abscissa_fp
    {\g_BBLR_mean_abscissa_fp  + \g_BBLR_abscissa_fp}
    \fp_gset:Nn \g_BBLR_mean_ordinate_fp
    {\g_BBLR_mean_ordinate_fp + \g_BBLR_ordinate_fp}
  \fi:
}
```

Loop ended. Now close the file and divide by the number of points.

```
\ior_close:N \g_BBLR_file_ior
\fp_gset:Nn \g_BBLR_mean_abscissa_fp
{\g_BBLR_mean_abscissa_fp / \g_BBLR_number_of_points_int}
\fp_gset:Nn \g_BBLR_mean_ordinate_fp
{\g_BBLR_mean_ordinate_fp / \g_BBLR_number_of_points_int}
```

The barycentric coordinates are defined for each point

$$\underline{v}_i = \underline{y}_i - \underline{\overline{y}} \tag{2}$$

and the empirical dispersion matrix is defined as:

$$C = \frac{1}{m} \sum_{i=1}^{m} \underline{v}_i \underline{v}_i^{\mathsf{T}}. \tag{3}$$

Superscript as in $()^{\mathsf{T}}$ means transpose. The elements of C are the second order central moments and they are denoted as:

$$C = \begin{pmatrix} k_{11} & k_{12} \\ k_{12} & k_{22} \end{pmatrix}. \tag{4}$$

A second scan of the data is performed to compute the sums that appears in (3) and to determine the the extremal values of the coordinates. Record scan can be regulated by a record counter, because the the number of points is now known.

```
\fp_zero:N  \g_BBLR_abscissa_SecOrdMoment_fp
\fp_zero:N  \g_BBLR_ordinate_SecOrdMoment_fp
\fp_zero:N  \g_BBLR_mixed_SecOrdMoment_fp
\fp_gset_eq:NN  \g_BBLR_min_abscissa_fp  \g_BBLR_mean_abscissa_fp
\fp_gset_eq:NN  \g_BBLR_min_ordinate_fp  \g_BBLR_mean_ordinate_fp
\fp_gset_eq:NN  \g_BBLR_max_abscissa_fp  \g_BBLR_mean_abscissa_fp
\fp_gset_eq:NN  \g_BBLR_max_ordinate_fp \g_BBLR_mean_ordinate_fp
\ior_open:Nn   \g_BBLR_file_ior \g_BBLR_file_name_tl
\int_zero:N    \g_BBLR_rec_count_int
\int_do_until:nn
{\g_BBLR_rec_count_int = \g_BBLR_number_of_points_int}
{
    \ior_str_get:NN \g_BBLR_file_ior \g_BBLR_file_line_tl
    \int_incr:N \g_BBLR_rec_count_int
    \BBLR_decode_data:
    \fp_gset:Nn \g_tmpa_fp
    {\g_BBLR_abscissa_fp - \g_BBLR_mean_abscissa_fp}
    \fp_gset:Nn \g_tmpb_fp
    {\g_BBLR_ordinate_fp - \g_BBLR_mean_ordinate_fp}
    \fp_gset:Nn \g_BBLR_abscissa_SecOrdMoment_fp
    {\g_BBLR_abscissa_SecOrdMoment_fp + \g_tmpa_fp * \g_tmpa_fp}
    \fp_gset:Nn \g_BBLR_mixed_SecOrdMoment_fp
    {\g_BBLR_mixed_SecOrdMoment_fp + \g_tmpa_fp * \g_tmpb_fp}
    \fp_gset:Nn  \g_BBLR_ordinate_SecOrdMoment_fp
    {\g_BBLR_ordinate_SecOrdMoment_fp + \g_tmpb_fp * \g_tmpb_fp}
\fp_gset:Nn  \g_BBLR_min_abscissa_fp
{min(\g_BBLR_min_abscissa_fp, \g_BBLR_abscissa_fp)}
\fp_gset:Nn  \g_BBLR_min_ordinate_fp
{min(\g_BBLR_min_ordinate_fp, \g_BBLR_ordinate_fp)}
\fp_gset:Nn  \g_BBLR_max_abscissa_fp
{max(\g_BBLR_max_abscissa_fp, \g_BBLR_abscissa_fp)}
\fp_gset:Nn  \g_BBLR_max_ordinate_fp
{max(\g_BBLR_max_ordinate_fp, \g_BBLR_ordinate_fp)}
}
\ior_close:N \g_BBLR_file_ior
\fp_gset:Nn \g_BBLR_abscissa_SecOrdMoment_fp
{\g_BBLR_abscissa_SecOrdMoment_fp / \g_BBLR_number_of_points_int}
\fp_gset:Nn \g_BBLR_mixed_SecOrdMoment_fp
{\g_BBLR_mixed_SecOrdMoment_fp / \g_BBLR_number_of_points_int}
\fp_gset:Nn \g_BBLR_ordinate_SecOrdMoment_fp
{\g_BBLR_ordinate_SecOrdMoment_fp / \g_BBLR_number_of_points_int}
\fp_gset:Nn \g_BBLR_Dabscissa_fp
{\g_BBLR_max_abscissa_fp - \g_BBLR_min_abscissa_fp }
\fp_gset:Nn \g_BBLR_Dordinate_fp
{\g_BBLR_max_ordinate_fp - \g_BBLR_min_ordinate_fp }
```

A single pass algorithm exists, but it is numerically less stable.

8.5 Classical linear regression

A line in the plane is described by the equation

$$y_2 = ay_1 + b \tag{5}$$

that contains the parameters a and b. For each point it is possible to define a distance or a discrepancy of the experimental data with respect to the model. In the given problem the second coordinate is much more affected by errors than the first coordinate. It is therefore reasonable to define the approximation error of each point as

$$e_i = y_{2i} - ay_{1i} - b \tag{6}$$

i.e. the difference between the empirical value y_{2i} and its model counterpart $ay_{1i} + b$. The global discrepancy between the data and the model is measured by the least square objective function defined by:

$$\psi = \sum_{i=1}^{m} e_i^2 \tag{7}$$

and the parameters a and b will be determined just by the minimization of the function ψ defined in (7).
In the present treatment of the regression problem as a pure approximation problem the definition of ψ in (7) seams quite arbitrary. It is anyway a convenient choice.
Expression (6) can be rewritten in the different form

$$e_i = v_{2i} - av_{1i} + \overline{y}_2 - a\overline{y}_1 - b \tag{8}$$

so that the function to be minimized can be expressed as the sum of two quadratic functions:

$$\psi = \sum_{i=1}^{m} (v_{2i} - av_{1i})^2 + m(\overline{y}_2 - a\overline{y}_1 - b)^2 \tag{9}$$

and the minimum can be attained considering the two terms one at a time. The second term in the right-hand side of (9) vanishes if the choice of b is:

$$b = \overline{y}_2 - a\overline{y}_1. \tag{10}$$

The first term in the right-hand side of (9) becomes:

$$\psi_{(a)} = m\left(k_{22} - 2ak_{12} + a^2k_{11}\right). \tag{11}$$

Searching the minimum of ψ w.r.t. a is therefore the search of the abscissa of the vertex of a parabola with axis parallel to the second coordinated axis. The result is:

$$a = k_{12}/k_{11} \tag{12}$$

Now the slope a and the intercept b can be actually computed.

```
\fp_gset:Nn \g_BBLR_slope_A_fp
{\g_BBLR_mixed_SecOrdMoment_fp / \g_BBLR_abscissa_SecOrdMoment_fp }
\fp_gset:Nn \g_BBLR_intercept_A_fp
{\g_BBLR_mean_ordinate_fp - \g_BBLR_slope_A_fp * \g_BBLR_mean_abscissa_fp}
```

The empirical data and the estimated values of a and b can be used to compute the value actually attained by the residuals e_i and by the function ψ. Then the index

$$\hat{\sigma}_0^2 = \psi/(m-2) \tag{13}$$

can be used to evaluate the general quality of the data and of the model. This claim is clearly quite generic. A complete understanding of this evaluation would require to treat the linear regression problem in the framework of the probabilistic estimation theory. The used notation is derived from that theory.
If the role of the two coordinates is exchanged the result for a becomes (still with reference to (5))

$$a = k_{22}/k_{12}. \tag{14}$$

A complete treatment of this different situation would include the redefinition of e_i and of ψ. The slope and the intercept can be computed according with the different assumption.

```
\fp_gset:Nn \g_BBLR_slope_B_fp
{\g_BBLR_ordinate_SecOrdMoment_fp / \g_BBLR_mixed_SecOrdMoment_fp}
\fp_gset:Nn \g_BBLR_intercept_B_fp
{\g_BBLR_mean_ordinate_fp - \g_BBLR_slope_B_fp * \g_BBLR_mean_abscissa_fp}
```

8.6 Symmetric linear regression

If both the coordinates of the experimental points are affected by the same uncertainty it is advisable to use a more symmetric optimality criterion and it is convenient to use a different model equation.
The same line can be described by a different equation, i.e.

$$x_1 y_1 + x_2 y_2 = f \tag{15}$$

or in vector form:

$$\underline{x}^{\mathsf{T}} \underline{y} = f. \tag{16}$$

The parameters in (16) are the scalar f and the elements of the vector $\underline{x}$, i.e. x_1 and x_2. The line described by (16) is obviously invariant when the three parameters are simultaneously scaled by a constant. The normalization condition

$$\underline{x}^{\mathsf{T}} \underline{x} = 1, \tag{17}$$

supplemented by $f \geq 0$, is quite convenient because the parameters will assume a significant geometrical meaning: $\underline{x}$ is the unit vector orthogonal to the line and f is the distance of the line from the origin. The expression

$$d = f - \underline{x}^{\mathsf{T}} \underline{y} \tag{18}$$

is the distance of the generic point $\underline{y}$ from the line with a sign that is positive for points on the same side of the origin.
The distance of each given point from the desired optimal line is denoted by d_i. It has a clear intrinsic geometrical meaning and it does not privileges one coordinate w.r.t. the other. The function to be minimized by the optimal line is

$$\phi = \frac{1}{m} \sum_{i=1}^{m} d_i^2. \tag{19}$$

The parameters of (16) are determined by the minimization of the function ϕ that can be expressed as:

$$\phi = \frac{1}{m}\sum_{i=1}^{m}(\underline{x}^{\mathsf{T}}\underline{y}_i - f)^2 \tag{20}$$

and then, after some algebraic manipulations:

$$\phi = \underline{x}^{\mathsf{T}}C\underline{x} + (f - \underline{x}^{\mathsf{T}}\overline{\underline{y}})^2. \tag{21}$$

The function ϕ is composed (as it was the function ψ) by the sum of two parts. The second term in the right-hand side of (21) vanishes if the choice of f is:

$$f = \underline{x}^{\mathsf{T}}\overline{\underline{y}}. \tag{22}$$

Then it is necessary to minimize the function

$$\phi_{(\underline{x})} = \underline{x}^{\mathsf{T}}C\underline{x} \tag{23}$$

with the constrain $\underline{x}^{\mathsf{T}}\underline{x} = 1$. It can be proved that the function $\phi_{(\underline{x})}$ is stationary if $\underline{x}$ is an eigenvector of C.
The function $\phi_{(\underline{x})}$ and the constrain must be combined using a Lagrange multiplier:

$$\Phi = \underline{x}^{\mathsf{T}}C\underline{x} + \lambda(1 - \underline{x}^{\mathsf{T}}\underline{x}). \tag{24}$$

Then the stationarity points of Φ must be determined. Equating to zero the derivatives of Φ gives

$$C\underline{x} = \lambda\underline{x} \tag{25}$$

i.e. $\underline{x}$ is an eigenvector of C.
The same result is obtained with the following argument. The function $\phi_{(\underline{x})}$ is stationary if its first variation is zero. The variation of $\underline{x}$ is named $\underline{\delta}$. It must respect the constrain, that becomes $\underline{\delta}^{\mathsf{T}}\underline{x} = 0$. The first variation of $\phi_{(\underline{x})}$ is $2\underline{\delta}^{\mathsf{T}}C\underline{x}$, and it is zero if and only if the following implication is valid: $\underline{\delta}^{\mathsf{T}}\underline{x} = 0 \implies \underline{\delta}^{\mathsf{T}}C\underline{x} = 0$, and the implication is valid if and only if the vector $C\underline{x}$ has the same direction of $\underline{x}$, i.e. if $\underline{x}$ is an eigenvector of C.
The result on the optimal line can be described geometrically in the following way: (i) the optimal line includes the barycenter of the data; (ii) the optimal line is orthogonal to the eigenvector of C corresponding to the minimum eigenvalue.
The obtained result is also valid in $\mathbb{R}^n$. A set of points in $\mathbb{R}^n$ must be approximated by an $(n-1)$-dimensional affine subspace. (Other more general situations can be considered.)
The trace of the matrix C, denoted as $\mathrm{tr}(C)$, is a measure of the global dispersion of the set of points. The minimum eigenvalue $\lambda_{\min}$ of C is a measure of the dispersion of the set of points with respect to the optimal affine subspace. Therefore the index

$$\frac{n\lambda_{\min}}{\mathrm{tr}(C)} \tag{26}$$

can be used as an indicator of the relative residual dispersion of the data around the optimal line. The defined index is dimensionless and it is always between 0 and 1.
For the actual computation of $\underline{x}$ it is convenient to consider the spectral factorization of the matrix C, i.e. $C = X\Lambda X^{\mathsf{T}}$ where Λ is a diagonal matrix whose diagonal

elements are the eigenvalues of C and X is an orthonormal matrix whose columns are the eigenvectors of C. The spectral factorization exists for any symmetric matrix, but it is specially simple for a 2×2 matrix.

$$\begin{pmatrix} k_{11} & k_{12} \\ k_{12} & k_{22} \end{pmatrix} = \begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} c & s \\ -s & c \end{pmatrix} \tag{27}$$

The eigenvalues can be easily obtained because their sum is the trace of C

$$\lambda_1 + \lambda_2 = k_{11} + k_{22} \tag{28}$$

and their product is the determinant of the same matrix. Therefore after some manipulations it results:

$$\lambda_1 - \lambda_2 = \sqrt{(k_{11} - k_{22})^2 + 4k_{12}^2} \tag{29}$$

and the two eigenvalues are then immediately obtained.
It is convenient to compute the difference of the two diagonal elements of the dispersion matrix and the difference of its eigenvalues.

```
\fp_gset:Nn   \g_BBLR_diag_diff_fp
{\g_BBLR_abscissa_SecOrdMoment_fp - \g_BBLR_ordinate_SecOrdMoment_fp}
\fp_gset:Nn   \g_BBLR_eig_diff_fp
{sqrt(\g_BBLR_diag_diff_fp * \g_BBLR_diag_diff_fp +
        4 * \g_BBLR_mixed_SecOrdMoment_fp * \g_BBLR_mixed_SecOrdMoment_fp)}
```

The computation of c and s is obtained from (27) taking into account that $c^2+s^2 = 1$. From (27) it results:

$$k_{11} - k_{22} = (\lambda_1 - \lambda_2)(c^2 - s^2) \tag{30}$$

and also

$$k_{12} = (\lambda_1 - \lambda_2)cs \tag{31}$$

that is only used to determine the sign of cs. The expression for the parameters c and s are:

$$c = \sqrt{\frac{1}{2} + \frac{k_{11} - k_{22}}{2(\lambda_1 - \lambda_2)}} \tag{32}$$

$$s = \operatorname{sgn}(k_{12})\sqrt{\frac{1}{2} - \frac{k_{11} - k_{22}}{2(\lambda_1 - \lambda_2)}} \tag{33}$$

The parameters s and c are the sine and cosine of the angle between the axis of y_1 and the eigenvector corresponding to the maximum eigenvalue.
They are computed using the already defined elements.

```
\fp_gset:Nn   \g_BBLR_cos_fp%
{sqrt((1 + \g_BBLR_diag_diff_fp / \g_BBLR_eig_diff_fp) / 2)}
\fp_gset:Nn \g_BBLR_sig_sin_fp {\fp_sign:n {\g_BBLR_mixed_SecOrdMoment_fp}}
\fp_gset:Nn   \g_BBLR_sin_fp
{\g_BBLR_sig_sin_fp*sqrt((1-\g_BBLR_diag_diff_fp / \g_BBLR_eig_diff_fp) / 2)}
```

The vector $\underline{x}$ is :

$$\underline{x} = \operatorname{sgn}(-s\overline{y}_1 + c\overline{y}_2) \begin{pmatrix} -s \\ c \end{pmatrix}. \tag{34}$$

The parameter a of model (5) can be obtained as:

$$a = s/c \tag{35}$$

Now the slope and the intercept of the optimal line corresponding to the symmetric criterion can be computed.

```
\fp_gset:Nn \g_BBLR_slope_S_fp
{\g_BBLR_sin_fp / \g_BBLR_cos_fp }
\fp_gset:Nn \g_BBLR_intercept_S_fp
{\g_BBLR_mean_ordinate_fp - \g_BBLR_slope_S_fp * \g_BBLR_mean_abscissa_fp}
}
```

The theoretical treatment of the proposed problem and the implementation of its numerical solution end here.

8.7 Print of table of results

`\lrprint` The command `\lrprint` prints some info on the data and the results of the computations in tabular form.

```
\NewDocumentCommand{\lrprint}{}{
\begin{center}
\begin{tabular}{| l | r |} \hline
 Data~File:                & \g_BBLR_file_name_tl                   \\ \hline
 Number~of~points:         & \int_use:N\g_BBLR_number_of_points_int \\ \hline
 Mean~values~of~the~coordinates:              &%
 $\fp_use:N \g_BBLR_mean_abscissa_fp$ \\ &
 $\fp_use:N \g_BBLR_mean_ordinate_fp$ \\ \hline
 Minimum~values~of~the~coordinates:        &%
 $\fp_use:N \g_BBLR_min_abscissa_fp$ \\ &
 $\fp_use:N \g_BBLR_min_ordinate_fp$    \\ \hline
  Maximum~values~of~the~coordinates:    &%
 $\fp_use:N \g_BBLR_max_abscissa_fp$ \\ &
 $\fp_use:N \g_BBLR_max_ordinate_fp$    \\ \hline
 {Second~order~moments}\phantom{xxxxxxxxx}{abscissa} &%
 $\fp_use:N \g_BBLR_abscissa_SecOrdMoment_fp$ \\
 \multicolumn{1}{|r|}{mixed}   & %
$\fp_use:N \g_BBLR_mixed_SecOrdMoment_fp$ ~ \\
\multicolumn{1}{|r|}{ordinate}   & %
$\fp_use:N \g_BBLR_ordinate_SecOrdMoment_fp$  \\ \hline
  Slope~and~intercept~of~optimal~line  & $\fp_use:N \g_BBLR_slope_A_fp$ \\
 (estimated~with~errors~in~ordinate)&$\fp_use:N \g_BBLR_intercept_A_fp$\\ \hline
  Slope~and~intercept~of~optimal~line & $\fp_use:N \g_BBLR_slope_B_fp$ \\
 (estimated~with~errors~in~abscissa)&$\fp_use:N \g_BBLR_intercept_B_fp$\\ \hline
  Components~of~unit~vector~along~the~line & $\fp_use:N  \g_BBLR_cos_fp$ \\
  &  $\fp_use:N \g_BBLR_sin_fp$   \\
  Slope~and~intercept~of~optimal~line &$\fp_use:N \g_BBLR_slope_S_fp$ \\
(estimated~with~symmetric~regression) &
 $\fp_use:N \g_BBLR_intercept_S_fp$\\ \hline
\end{tabular}
\end{center}
}
```

8.8 Plot of points and lines

\lrplot The command \lrplot produce a framed plot of the data and of the regression line(s). The size of the plot and its actual content are determined by the arguments.

```
\NewDocumentCommand{\lrplot}{mmmmm}{%
```

The plotting area is divided into a main plotting area for the representation of points and line(s) and a small surrounding free space. The height is computed taking into account the distribution of the points.

```
\fp_gset:Nn \g_BBLR_base_fp {#1}
\fp_gset:Nn \g_BBLR_Xbase_fp {\g_BBLR_base_fp - 0.6}
\fp_gset:Nn \g_BBLR_scale_factor_fp{\g_BBLR_Xbase_fp / \g_BBLR_Dabscissa_fp}
\fp_gset:Nn \g_BBLR_Xheight_fp {\g_BBLR_Dordinate_fp * \g_BBLR_scale_factor_fp}
\fp_gset:Nn \g_BBLR_height_fp {\g_BBLR_Xheight_fp + 0.6}
```

The information about the items to be plotted is in the remaining arguments.

```
\str_gset:Nn \g_BBLR_point_code_str {#2}
\str_gset:Nn \g_BBLR_labelA_str     {#3}
\str_gset:Nn \g_BBLR_labelB_str     {#4}
\str_gset:Nn \g_BBLR_labelS_str     {#5}
\bool_gset:Nn \g_BBLR_plot_points_bool
{!(\str_if_eq_p:NN \g_BBLR_point_code_str \c_BBLR_minus_str)}
\bool_gset:Nn \g_BBLR_plot_lineA_bool
{!(\str_if_eq_p:NN \g_BBLR_labelA_str \c_BBLR_minus_str)}
\bool_gset:Nn \g_BBLR_plot_lineB_bool
{!(\str_if_eq_p:NN \g_BBLR_labelB_str \c_BBLR_minus_str)}
\bool_gset:Nn \g_BBLR_plot_lineS_bool
{!(\str_if_eq_p:NN \g_BBLR_labelS_str \c_BBLR_minus_str)}
```

The unit of length is 1 centimeter. The plotting area is framed.

```
\setlength{\unitlength}{1.0cm}
\fp_gset:Nn \g_tmpa_fp {\g_BBLR_Xbase_fp +0.2}
\fp_gset:Nn \g_tmpb_fp {\g_BBLR_Xheight_fp +0.1}
\begin{picture}(\fp_use:N\g_BBLR_base_fp,\fp_use:N\g_BBLR_height_fp)(-0.3,-0.3)
\put(-0.1,-0.1){\line(1,0){\fp_use:N\g_tmpa_fp}}
\put(-0.1,\fp_use:N\g_tmpb_fp){\line(1,0){\fp_use:N\g_tmpa_fp}}
\fp_gset:Nn \g_tmpa_fp {\g_tmpa_fp -0.1}
\fp_gset:Nn \g_tmpb_fp {\g_tmpb_fp +0.1}
\put(-0.1,-0.1){\line(0,1){\fp_use:N\g_tmpb_fp}}
\put(\fp_use:N\g_tmpa_fp,-0.1){\line(0,1){\fp_use:N\g_tmpb_fp}}
```

The plot of points and line(s) is obtained using auxiliary functions.

```
\thicklines
\bool_if:nT {\g_BBLR_plot_points_bool}{\BBLR_plot_points:}
\bool_if:nT {\g_BBLR_plot_lineA_bool}{
\BBLR_draw_line:NNN \g_BBLR_slope_A_fp\g_BBLR_intercept_A_fp\g_BBLR_labelA_str}
\bool_if:nT {\g_BBLR_plot_lineB_bool}{
\BBLR_draw_line:NNN \g_BBLR_slope_B_fp\g_BBLR_intercept_B_fp\g_BBLR_labelB_str}
\bool_if:nT {\g_BBLR_plot_lineS_bool}{
\BBLR_draw_line:NNN \g_BBLR_slope_S_fp\g_BBLR_intercept_S_fp\g_BBLR_labelS_str}
\end{picture}
}%
```

8.9 Functions for internal use

The functions listed here after are for internal use and are just minimally documented.

\BBLR_decode_data: The function \BBLR_decode_data: extract two numeric values from the string read from the file. Some tricky actions are necessary because a so called csv file sometime do not contains the separating commas.

```
\cs_new_protected:Nn \BBLR_decode_data: {
\tl_trim_spaces:N \g_BBLR_file_line_tl
\int_gzero:N \g_tmpa_int
\int_gzero:N \g_BBLR_first_separator_int
\int_gzero:N \g_BBLR_last_separator_int
\int_gset:Nn \g_BBLR_record_length_int {
\str_count:N \g_BBLR_file_line_tl}
\str_map_variable:NNn \g_BBLR_file_line_tl \g_tmpa_str {
\int_gincr:N \g_tmpa_int
\bool_lazy_or:nnTF
{\str_if_eq_p:NN \g_tmpa_str \c_BBLR_comma_str}
{\str_if_eq_p:NN \g_tmpa_str \c_BBLR_space_str}
{\int_gset_eq:NN \g_BBLR_last_separator_int \g_tmpa_int
\int_if_zero:nTF {\g_BBLR_first_separator_int}
{\int_gset_eq:NN \g_BBLR_first_separator_int \g_tmpa_int
}{\prg_do_nothing:}
}{\prg_do_nothing:}
}
\int_gincr:N \g_BBLR_last_separator_int
\int_gdecr:N \g_BBLR_first_separator_int
\fp_gset:Nn  \g_BBLR_abscissa_fp{
\str_range:Nnn \g_BBLR_file_line_tl{1}{\g_BBLR_first_separator_int}}
\fp_gset:Nn  \g_BBLR_ordinate_fp{
\str_range:Nnn \g_BBLR_file_line_tl
{\g_BBLR_last_separator_int}{\g_BBLR_record_length_int}}
}
```

\BBLR_plot_points: The function \BBLR_plot_points: scans the data file to read the coordinates and it draws a circle for each point.

```
\cs_new_protected:Nn  \BBLR_plot_points: {
\ior_open:Nn   \g_BBLR_file_ior \g_BBLR_file_name_tl
\int_zero:N    \g_BBLR_rec_count_int
\int_do_until:nn
{\g_BBLR_rec_count_int = \g_BBLR_number_of_points_int}
{
    \ior_str_get:NN \g_BBLR_file_ior \g_BBLR_file_line_tl
    \int_incr:N \g_BBLR_rec_count_int
    \BBLR_decode_data:
    \fp_gset:Nn \g_tmpa_fp{(\g_BBLR_abscissa_fp-\g_BBLR_min_abscissa_fp)*
    \g_BBLR_scale_factor_fp}
    \fp_gset:Nn \g_tmpb_fp{(\g_BBLR_ordinate_fp-\g_BBLR_min_ordinate_fp)*
    \g_BBLR_scale_factor_fp}
    \put(\fp_use:N\g_tmpa_fp, \fp_use:N\g_tmpb_fp){
    {\circle*{\fp_use:N\g_BBLR_diameter_fp}}}
}
\ior_close:N \g_BBLR_file_ior
}
```

`\BBLR_draw_line:NNN` The function `\BBLR_draw_line:NNN` draws the line. The first two parameters given as arguments are the slope and the intercept. The third parameter is a label.

The next code finds the intersection of the line with the plotting area.

```
\cs_new_protected:Nn  \BBLR_draw_line:NNN {
\fp_gset:Nn \fp_tmpa_fp {#1 * \g_BBLR_min_abscissa_fp + #2 }
\fp_compare:nTF{\fp_tmpa_fp > \g_BBLR_max_ordinate_fp}{
\fp_gset:Nn \g_BBLR_min_draw_abscissa_fp {(\g_BBLR_max_ordinate_fp -#2) / #1}
}{
\fp_compare:nTF{\fp_tmpa_fp < \g_BBLR_min_ordinate_fp}{
\fp_gset:Nn \g_BBLR_min_draw_abscissa_fp {(\g_BBLR_min_ordinate_fp - #2) / #1}
}{
\fp_gset:Nn \g_BBLR_min_draw_abscissa_fp { \g_BBLR_min_abscissa_fp }
}}
\fp_gset:Nn \fp_tmpa_fp {#1 * \g_BBLR_max_abscissa_fp + #2 }
\fp_compare:nTF{\fp_tmpa_fp > \g_BBLR_max_ordinate_fp}{
\fp_gset:Nn \g_BBLR_max_draw_abscissa_fp {(\g_BBLR_max_ordinate_fp -#2) / #1}
}{
\fp_compare:nTF{\fp_tmpa_fp < \g_BBLR_min_ordinate_fp}{
\fp_gset:Nn \g_BBLR_max_draw_abscissa_fp { (\g_BBLR_min_ordinate_fp - #2) / #1}
}{
\fp_gset:Nn \g_BBLR_max_draw_abscissa_fp {  \g_BBLR_max_abscissa_fp }
}}
```

Some parameters (i.e. starting point and base-length) are computed and the line is drawn.

```
\fp_gset:Nn \fp_tmpa_fp {(\g_BBLR_min_draw_abscissa_fp -
\g_BBLR_min_abscissa_fp)* \g_BBLR_scale_factor_fp}
\fp_gset:Nn \fp_tmpb_fp {(#1 * \g_BBLR_min_draw_abscissa_fp + #2 -
\g_BBLR_min_ordinate_fp)* \g_BBLR_scale_factor_fp}
\fp_gset:Nn \fp_BBLR_line_base_length_fp{(\g_BBLR_max_draw_abscissa_fp -
\g_BBLR_min_draw_abscissa_fp) * \g_BBLR_scale_factor_fp}
\put(\fp_use:N\fp_tmpa_fp, \fp_use:N\fp_tmpb_fp){
\line(1.,\fp_use:N #1){\fp_use:N\fp_BBLR_line_base_length_fp}}
```

The third parameter is used as a label, if it is not a `+`.

```
\bool_if:nF {\str_if_eq_p:NN #3 \c_BBLR_plus_str}{
\fp_gset:Nn \fp_tmpa_fp
{0.08 * \g_BBLR_min_draw_abscissa_fp + 0.92 * \g_BBLR_max_draw_abscissa_fp}
\fp_gset:Nn \fp_tmpb_fp {#1 * \fp_tmpa_fp + #2 }
\fp_gset:Nn \fp_tmpa_fp
{(\fp_tmpa_fp-\g_BBLR_min_abscissa_fp)*\g_BBLR_scale_factor_fp
+ 0.3 * #1 /sqrt(1.+#1*#1)}
\fp_gset:Nn \fp_tmpb_fp
{(\fp_tmpb_fp-\g_BBLR_min_ordinate_fp)* \g_BBLR_scale_factor_fp
- 0.3 /sqrt(1.+#1*#1)}
\put(\fp_use:N\fp_tmpa_fp, \fp_use:N\fp_tmpb_fp){#3}
}
}
```

```
\ExplSyntaxOff
```

9 Acknowledgments

The colleagues Paolo Zatelli, Alfonso Vitti and Giulia Graldi read some preliminary version of this text and suggested several improvements.

10 About the references

Mathematics

The books by Lang [7] and by Strang [14] give all the background on linear algebra. The texts by Sansò [12, 13] (in italian) treat the teory of probability and its application to metrology. See: `http://www.geolab.polimi.it/text-books/`.
The paper by Karl Pearson [11] is the oldest text that I have found on the symmetric regression, or total regression.

Programming

The two documents [9, 10] are the fountamental and official guide for LaTeX3 programming. The books by Donald Knuth [6] and Leslie Lamport [8] are still essential references. The papers by Enrico Gregorio [1, 2, 3, 4, 5] explain some general and some special aspect of LaTeX3 programming.

11 References

[1] Enrico Gregorio, *LaTeX3: un nuovo gioco per i maghi e per diventarlo*, ArsTeXnica **14** (2012), 41–47.

[2] Enrico Gregorio, *Liste, cicli, LaTeX3*, ArsTeXnica **22** (2016), 69–77.

[3] Enrico Gregorio, *Condizionali in LaTeX*, ArsTeXnica **24** (2017), 37–44.

[4] Enrico Gregorio, *Funzioni e* `expl3`, ArsTeXnica **30** (2020), 36–45.

[5] Enrico Gregorio, *Functions and* `expl3`, TUGboat **41** (2020), no. 3, 299–307.

[6] Donald Knuth, *The TeXbook*, American Mathematical Society and Addison-Wesley, 1986.

[7] Serge Lang, *Linear Algebra*, Springer-Verlag, Berlin Heidelberg, 1987.

[8] Leslie Lamport, *LaTeX - A document preparation system (2nd ed.)*, Addison-Wesley, 1994. something interesting in the fist edition, too.

[9] The LaTeX project team, *The* `expl3` *package and LaTeX3 programming* (2024). file: `expl3.pdf` available in CTAN in l3kernel.

[10] The LaTeX project team, *The LaTeX3 interface* (2024). file: `interface3.pdf` available in CTAN in l3kernel.

[11] Karl Pearson, *On lines and planes of closest fit to systems of points in space*, Philosophical Magazine **2** (1901), no. 11, 559–572.

[12] Fernando Sansò, *Elementi di teoria della probabilità*, Città-Studi, Milano, 1996.

[13] Fernando Sansò, *La teoria della stima*, Città-Studi, Milano, 1996.

[14] Gilbert Strang, *Introduction to linear algebra*, Wellesley-Cambridge press, 2009.

***