UDC 631.331.90

FITTING THE MODEL H (A/m) = F(f(Gz))

Kushnirenko A.G., C.Sc.
The Nizhyn Agrotechnical Institute
Tel. (04631) 4-35-89

Annotation – work is devoted the method of regressive analysis of experimental data and construction of model of F (f (Gz))

Keywords – regressive analysis, graph, mathematical model, confidence interval, error.

Introduction. In practice, we can see more problem fitting the model $y_i=\beta_1 f_{1\,i}+\beta_2 f_{2\,i}+\ldots+\beta_p f_{p\,i}+e_i$ where y_i is the i^{th} response, $f_{j\,i}$ is the j^{th} basis function evaluated at the i^{th} observation, and e_i is the i^{th} residual error [1 – 2]. In the Wolfram Mathematica 6.0, the built-in function *Fit* finds a least-squares fit to a list of data as a linearcombination of the specified basis functions. The functions *Regress* and *DesignedRegress* provided in this package augment *Fit* by giving a list of commonly required diagnostics such as the coefficient of determination *RSquared,* the analysis of variance table *ANOVATable,* and the mean squared error *EstimatedVariance.* The output of regression functions can be controlled so that only needed information is produced. The *Linear Regression* provides analogous functionality for nonlinear models.

Material and methods. The basis functions f_j specify the predictors as functions of the independent variables. The resulting model for the response variable is:

$$H = f (b + b_1 f + b f^2) , \qquad (1)$$

Estimates of the coefficients $b_1,\ldots,b_p$ are calculated to minimize:

$$\sum_{i=1}^{n} e_i^2 , \qquad (2)$$

the error or residual sum of squares. For example, simple linear regression is accomplished by defining the basis functions as $f_1=1$ and $f_2=x$, in which case b_1 and b_2 are found to minimize:

$$\sum_{i=1}^{n} [H_i - (b_1 + b_2 f_i)]^2. \qquad (3)$$

The arguments of *Regress* are of the same form as those of *Fit*. The data can be a list of vectors, each vector consisting of the observed values of the independent variables and the associated response. The basis functions f_j must be functions of the symbols given as variables. These symbols correspond to the independent variables represented in the data. By default, a constant function $f_j = 1$ is added to the list of basis functions it not explicitly given in the list of basis functions.

The data can also be a vector of data points. In this case, *Regress* assumes that this vector represents the values of a response variable with the independent variable having values 1, 2, ….

Results of the Measurements. Ways of specifying data in *Regress*.
This loads the package.:

<< LinearRegression`

For example, this data contains ordered pairs of a H(A/m) and a f (Gz):

Data={{100,330},{200,336},{300,380},{400,395},{500,430},{600,490},{700,557}, {800,590},{900,680}};

This is a plot of the data:

ListPlot[{data}

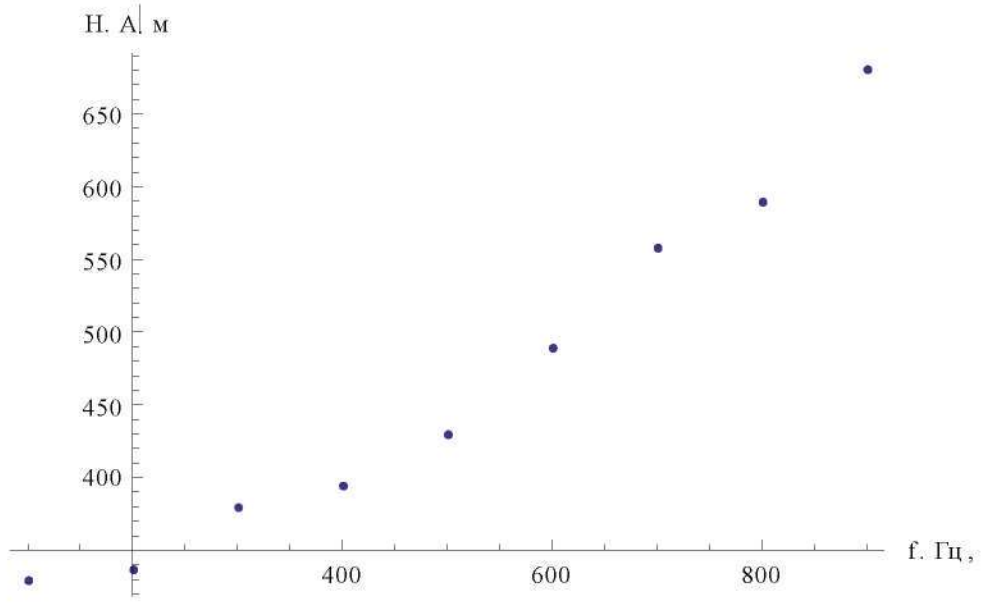

Fig. 1. This is a plot of the data points of the functions H = f (f), А/м, f - Гц

This is the output for fitting the model:

$$y_i = b_0 + b_1 x + b_2 x^2 + e_i. \quad (4)$$

Regress[data,{1,x²},x]

You can use Fit if you want only the fitted function:

Fit[data,{1,x^2},x]

$$H = 328.2 + 0.043f + 0.00043 f^2 \text{ , А / м} \quad (5)$$

Where f – Гц.

Two of the options of *Regress* influence the method of calculation. *IncludeConstant* has a default setting *True,* which causes a constant term to be added to the model even if it is not specified in the basis functions. To fit a model without this constant term, specify *IncludeConstant -> False* and do not include a constant in the basis functions.

The *Weights* option allows you to implement weighted least squares by specifying a list of weights,one for each data point;the default *Weights -> Automatic* implies a weight of unity for each data point.When *Weights -> {w_1,…,w_n}*,the parameter estimates are chosen to minimize the weighted sum of squared residuals:

$$\sum_{i=1}^{n} w_i e_i^2 . \quad (6)$$

Weights can also specify a pure function of the response. For example, to choose parameter estimates to minimize:

$$\sum_{i=1}^{n} \sqrt{y_i \; e_i^2} , \quad (7)$$

set *Weights -> (Sqrt [#] &).*

The options *RegressionReport* and *BasisNames* affect the form and content of the output.If *RegressionReport* is not specified, *Regress* automatically gives a list including values for *ParameterTable, RSquared, AdjustedRSquared, EstimatedVariance* and *ANOVATable.* This set of objects comprises the default *Summary.* The option *RegressionReport* can be used to specify a single object or a list of objects so that more (or less) than the default set of results is included in the output. *RegressionReportValues[Regress]* gives the objects that may be included in the *RegressionReport* list for the *Regress* function.

With the option *BasisNames,* you can label the headings of predictors in tables such as *ParameterTable* and *ParameterCITable.*

The regression functions will also accept any option that can be specified for *SingularValueList* or *StudentTCI.* In particular,the numerical tolerance for the internal singular value decomposition is specified using

Tolerance, and the confidence level for hypothesis testing and confidence intervals is specified using *ConfidenceLevel*.

ANOVATable, a table for analysis of variance, provides a comparison of the given model to a smaller one including only a constant term. If *IncludeConstant->False* is specified, then the smaller model is reduced to the data. The table includes the degrees of freedom, the sum of squares and the mean squares due to the model (in the row labeled Model) and due to the residuals (in the row labeled Error). The residual mean square is also available in *EstimatedVariance*, and is calculated by dividing the residual sum of squares by its degrees of freedom. The *F*-test compares the two models using the ratio of their mean squares. If the value of *F* is large, the null hypothesis supporting the smaller model is rejected.

To evaluate the importance of each basis function,you can get information about the parameter estimates from the parameter table obtained by including *ParameterTable* in the list specified by *RegressionReport*. This table includes the estimates,their standard errors,and*t*-statistics for testing whether each parameter is zero.The*p*-values are calculated by comparing the obtained statistic to the*t*distribution with*n*-*p*degrees of freedom,where*n*is the sample size and*p*is the number of predictors.Confidence intervals for the parameter estimates,also based on the*t*distribution,can be found by specifying *ParameterCITable*. *ParameterConfidenceRegion* specifies the ellipsoidal joint confidence region of all fit parameters. *ParameterConfidenceRegion* [$\{f_{i\,1},f_{i\,2},\ldots\}$] specifies the joint conditional confidence region of the fit parameters associated with basis functions $\{f_{i\,1},f_{i\,2},\ldots\}$,a subset of the complete set of basis functions.

The square of the multiple correlation coefficient is called the coefficient of determinationR^2,and is given by the ratio of the model sum of squares to the total sum of squares.It is a summary statistic that describes the relationship between the predictors and the response variable. *AdjustedRSquared* is defined as:

$$R_2 = 1 - ((n-1)/(n-p))\,(1-R^2), \qquad (8)$$

and gives an adjusted value that you can use to compare subsequent subsets of models.The coefficient of variation is given by the ratio of the residual root mean square to the mean of the response variable.If the response is strictly positive,this is sometimes used to measure the relative magnitude of error variation.

Each row in *MeanPredictionCITable* gives the confidence interval for the mean response at each of the values of the independent variables.Each row in *SinglePredictionCITable* gives the confidence interval for a single observed response at each of the values of the

independent variables. *MeanPredictionCITable* gives a region likely to contain the regression curve,while *SinglePredictionCITable* gives a region likely to contain all possible observations.

The following gives the residuals,the confidence interval table for the predicted response of single observations,and the parameter joint confidence region:

regress=Regress[data,{1,x²},x,RegressionReport → {FitResiduals, SinglePredictionCITable, ParameterConfidenceRegion}]

This is a list of the residuals extracted from the output:

errors=FitResiduals/. regress

The observed response, the predicted response, the standard errors of the predicted response, and the confidence intervals may also be extracted:

{observed,predicted,se,ci}=Transpose[(SinglePredictionCITable/regress) [1]];

This plots the predicted responses against the residuals:

ListPlot[Transpose[{predicted,errors}]]

Here the predicted responses and lower and upper confidence limits are paired with the corresponding *x* values:

```
( xval = data [ [ All,1 ] ];
Predicted = Transpose [ { xval, predicted } ];
lowerCI = Transpose [ { xval, First / @ ci } ];
upperCI = Transpose [ { xval, Last / @ ci} ]);
```

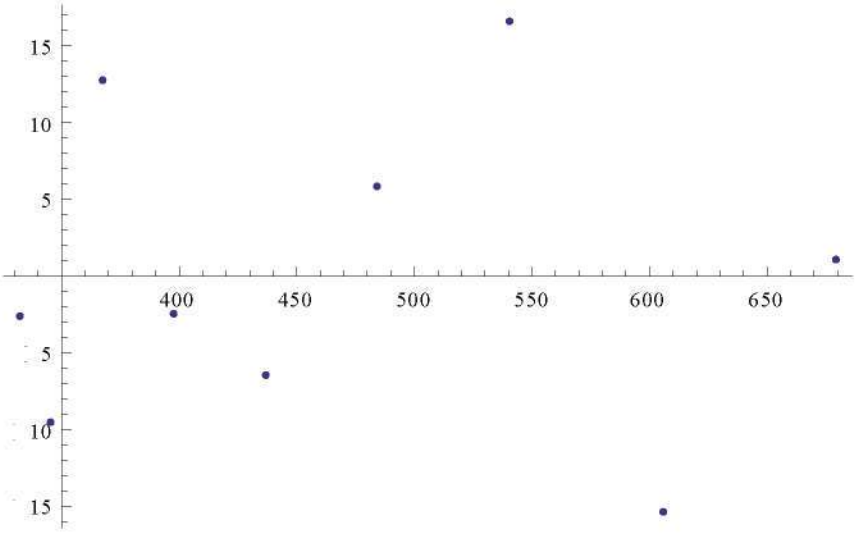

Fig. 2. This plots the predicted responses against the residuals

This displays the raw data, fitted curve, and the 95 % confidence intervals for the predicted responses of single observations.

ListPlot[{data,predicted,lowerCI,upperCI},Joined {False,True,True,True}, PlotStyle {Automatic,Automatic,{Dashing[{.05,.05}],Gray},{Dashing[{.0, .05}],Gray}}]

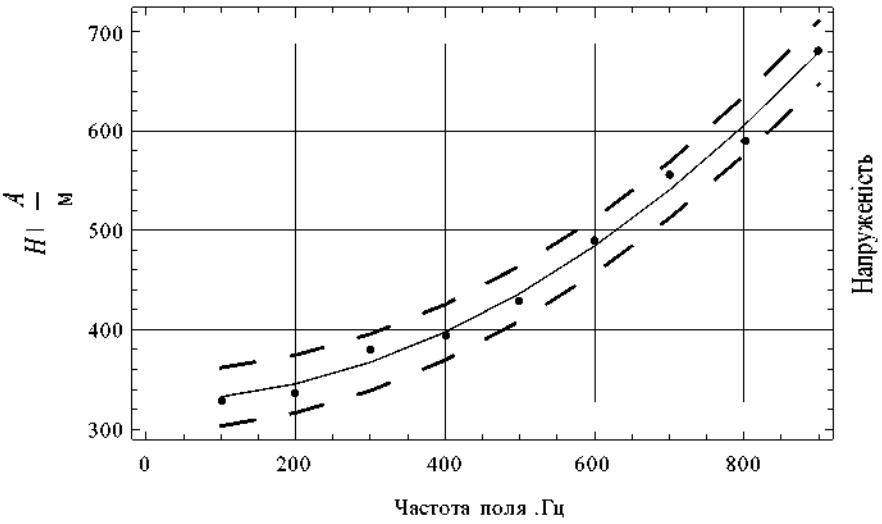

Fig. 3. This displays the raw data, fitted curve, and the 95 % confidence intervals for the predicted responses of single observations

Grapfics may be used to display an *Ellipsoid* object. This is the joint 95 % confidence region for the regression parameters.

Graphics[ParameterConfidenceRegion/.regress
,Axes → True, AxesLabel → { " Constant " , " x Squared " },
AspectRatio → 1]

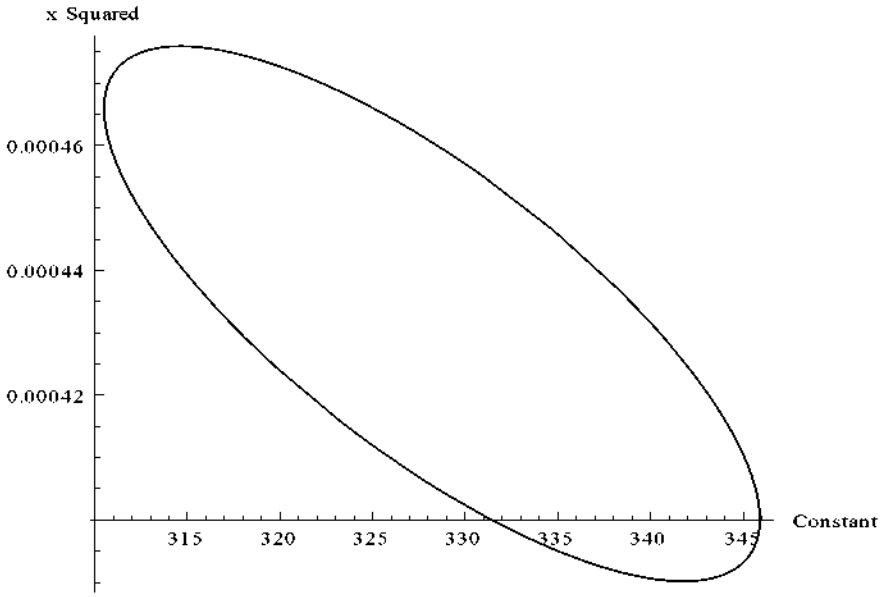

Fig. 4. Graphics may be used to display an *Ellipsoid* object. This is the joint 95 % confidence region for the regression parameters.

This package provides numerous diagnostics for evaluating the data and the fit. The *HatDiagonal* gives the leverage of each point, measuring whether each observation of the independent variables is unusual. *CookD* and *PredictedResponseDelta* are influence diagnostics, simultaneously

measuring whether the independent variables and the response variable are unusual. Unfortunately, these diagnostics are primarily useful in detecting single outliers. In particular, the diagnostics may indicate a single outlier, but deleting that observation and recomputing the diagnostics may indicate others. All these diagnostics are subject to this masking effect.

Some diagnostics indicate the degree to which individual basis functions contribute to the fit, or whether the basis functions are involved in a collinear relationship. The sum of the elements in the *SequentialSumOfSquares* vector gives the model sum of squares listed in the *ANOVATable*. Each element corresponds to the increment in the model sum of squares obtained by sequentially adding each nonconstant basis function to the model. Each element in the *PartialSumOfSquares* vector gives the increase in the model sum of squares due to adding the corresponding nonconstant basis function to a model consisting of all other basis functions. *SequentialSumOfSquares* is useful in determining the degree of a uninvariate polynomial model, while *PartialSumOfSquares* is useful in trimming a large set of predictors. VarianceInflation or *EigenstructureTable* may also be used for predictor set trimming.

The Durbin–Watson *d* statistic is used for testing the existence of a first-order autoregressive process. The statistic takes on values between 0 and 4, with values near the middle of that range indicating uncorrelated errors, an underlying assumption of the regression model. Critical values for the statistic vary with sample size, the number of parameters in the model, and the desired significance. These values can be found in published tables.

Other statistics not mentioned here can be computed with the help of the catcher matrix. This matrix catches all the information the predictors have about the parameter vector. This matrix can be exported from Regress by specifying *CatcherMatrix* with the *RegressionReport* option.

Frequently, linear regression is applied to an existing design matrix rather than the original data. A design matrix is a list containing the basis functions evaluated at the observed values of the independent variable. If your data is already in the form of a design matrix with a corresponding vector of response data, you can use *DesignedRegress* for the same analyses as provided by *Regress*. *DesignMatrix* puts your data in the form of a design matrix.

DesignMatrix takes the same arguments as *Regress*. It can be used to get the necessary arguments for *DesignedRegress*, or to check whether you correctly specified your basis functions. When you use *DesignMatrix*, the constant term is always included in the model unless *IncludeConstant->False* is specified. Every option of *Regress* except *IncludeConstant* is accepted by *DesignedRegress*.

RegressionReportValues[DesignedRegress] gives the values that may be included in the *RegressionReport* list for the *DesignedRegress* function.

This is the design matrix used in the previous regression analysis:
mat=DesignMatrix[data,{1,x²},x]
{{1,10000},{1,40000},{1,90000},{1,160000},{1,250000},{1,360000},
{1,490000},{1,640000},{1,810000}}

Here is the vector of observed responses:
response=data[[All,-1]]
{330,336,380,395,430,490,557,590,680}

Summary. The result of *DesignedRegress* is equivalent to that of *Regress*:
DesignedRegress[mat,response,BasisNames → {"Constant","x Squared"}]

DesignedRegress will also accept the singular value decomposition of the design matrix. If the regression is not weighted, this approach will save recomputing the design matrix decomposition.

This is the singular value decomposition of the design matrix:
svd=SingularValueDecomposition[mat];
When several responses are of interest, this will save recomputing the design matrix decomposition:
DesignedRegress[svd,response,RegressionReport → BestFitParameters]
{330,336,380,395,430,490,557,590,680},RegressionReport→BestFitParameters]

The fitted function: $H = 328.2 + 0.043f + 0.00043 f^2$, (А / м) best fits parameters.

References

1. *Branch, M. A.* Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems /. Branch M.A., Coleman T. F., Li Y. // SIAM Journal on Scientific Computing. – SIAM, 1999.-Vol. 21, Number 1. – P. 1-23.

2. *Cleveland W.S.* Robust Locally Weighted Regression and Smoothing Scatterplots// Journal of the American Statistical Association. – 1979. – Vol. 74. – P. 829-836.

СТВОРЕННЯ МАТЕМАТИЧНОЇ МОДЕЛІ ТИПУ H (A/m) = F (f (Gz))

Кушніренко А.Г.

Анотація

Робота присвячена методиці регресивного аналізу експериментальних даних та побудові моделі F (f (Gz))

СОЗДАНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ТИПА H (A/m) = F (f (Gz))

Кушниренко А.Г.

Аннотация

Работа посвящена методике регрессивного анализа экспериментальных данных и построению модели F (f (Gz))