Available online at www.joac.info

ISSN: 2278-1862

Journal of Applicable Chemistry

2016, 5(5): 908-1033

(International Peer Reviewed Journal)

Research tutorial (ResT)

[Computational/Chemical]TensorLab(CTLab)

Part 2: Linear Least squares in Matlab

K. Ramakrishna[1] and R. Sambasiva Rao[2*]

1. Department of Chemistry, Gitam Institute of Science, Gitam University, Visakhapatnam, 530 017, **INDIA**

2. School of Chemistry, Andhra University, Visakhapatnam 530 003, **INDIA**

Email: karipeddirk@gmail.com, rsr.chem@gmail.com

Accepted on 10th September2016

(Dedicated with profound respects to Dr K V Suryanarayana, former professor of statistics, Andhra University on his seventy fifth birth anniversary)

Conspectus

Background:The models are precise expression of experimental results but not at all a substitute.On the other hand, data driven models do not start with any prefixed model, but at the end a model emerges.The linguistic models or automatic genetic algorithm/genetic programming generate a set of equivalent models and submerge most of earlier category although the mathematical/physical form is different for a naked eye. The regression models, self-organizing models, multiple-(constrained) optimization (with conflicting subgoal) models form major category in the bandwagon of computational tools.

Purpose: The focus of the current research review is to start with simple as possible matrix formulae to estimate regression parameters of linear/polynomial models in one explanatory variable (x) and coding in Matlab illustrating the application for small number of (six to ten) noise free simulated data. The results can be arrived at without any gadget. The perturbation of statistics of model parameters with (homoscedastic) Gaussian noise is dilated. The effects of outliers are exemplified remedial measures viz. least median squares (LMS) and least trimmed squares (LTS) are illustrated. The exhaustive set of models in analyzing data from polynomial models is developed in polyLS2015. The method of least squares is derived for univariate replicate data adhering to mean model perturbed by Gaussian noise. MAD statistic, a robust measure of central tendency is used to detect outliers and probe into central tendency of data in their presence. Linear parametricRegression with Multiple-X variables (MLR) and single response, a hard model is considered. A function of two explanatory variables is coded in MLR2015.m and simulated data sets amply illustrate its utility.In this phase, only mathematical formulae, m-functions, simple-as-possible examples are narrated. Anobject with typical results of each method is invoked and tabular and graphic output programs are available.

In the second phase, the default datasets, autotest_$$$ for all possible testing of program capabilities are discussed. The knowledge-based approach for input checking, validating input data/intermediate results structure to process a mathematical task (set of formulae) are developed in the if-then-else numericalrules. The necessary conditions, failure flags, remedial measures for each type of analysis and textual summary of algorithm flow with Matlab functions used are narrated in the next phase.

Each of these modules run individually and a bunch of them form a combined solution.A template (GUI) mode for selection and transparent flow of the software is under development.

Keywords: Cause-effect relationships, normal distribution, homoscedastic, heteroscedastic, residuals, absolute_residuals, squares of residuals, least_sum, regression parameters, linear, normal population, LLS, LAD, polyLS, LMS, MLR, statistics [residuals, parameters], ANOVA, information, KBs, NCs, failure conditions, Remedial measures, matlab_functions.

Contents

Linear Least squares with Matlab

Supplementary Information (SI)

SI1 Typical statistical packages
SI2 Toolboxes of MATLAB
SI3 Statistics Toolbox of MATLAB

INTRODUCTION

Ever since the human being observed, remembered, expressed to himselfand/or to others, simple relations between repeated happenings were formulated at random. Over a period of time, the accumulated simple links were refined, contradictions were sorted out and evidences were preserved.From this era of dualism of optimism/skepticism, right/wrong, correct/incorrect, true/false, empiricism emerged with a hope of persistence and getting prepared to pessimistic band of surroundings/life processes and so on. The tools developed/evolved were mainly to hunt for food, adaptation for harsh environment and preserving their progeny/musings during stone and iron ages.

1.1 Science i.e. Experiment, Third_eye and Computation (etc.): Ever since the measurement of time and distance started, precision/accuracy increased continuously in twentieth century. The repeatability of any observation in different trials under the same conditions (of experiment) gave birth to theories of determinism, probability, fuzziness and chaos.Now, even one failure in six sigma limits is a challenge in groundbreaking discoveries with experimental outcome. In this backdrop, theword scienceis familiar to everyone now in twenty first century, 'first century of the 3rd millennium', and all have a feel for it. But, what is science? It is mind-blowing enquiry if not impossible to define and dilate what all science is, even leaving aside what is not science. With growing collection of direct observables and indirect observations (response), cause-effect (or response as a function of explanatory variables) relationships evolved. In this decade, the experiments of CERN culminated into an unequivocal evidence for boson, opening a new era to probe into dark matter and dark energy, the light of future. The detection of gravitational waves and mass of neutrino are experiments of concerted efforts of tens of thousands of scientists for over two decades. The belief in the last century was gravitational waves cannot be detected and neutrino has no mass.

But, any discipline starts with empiricism based on raw observations without (experimental, data collection) design and inferences with experience in some other field of their expertise or accumulated knowledge of experts. Theoretical postulates/typical solutions for mathematical formulation of the task is the brain storming activity of (applied) mathematicians with input from experimentalists or from published literature. The numerical methods for reliable solution of mathematical equations and computational details for parallel /high precision (32-bit/64-bit) hardware and scale up (number of data points and variables from tens to thousands) software vary from time to time with global necessity, transportability and interoperability requirements of experts and routine operations in the hands of scientific assistants/technicians/skilled personnel.

The transparency of the method, guidelines of Dos and Don'ts were implicit in the last century, but now explicit passive documentation and preferably integrated software modules for automatic rescheduling the workflow (method choice flow) with outputting along with results is indispensable.It is not an option, but of high priority workflow even at the moment. Moreover, whether the core of necessary conditions are satisfied for the current task and changes if any of preset computational strategy are recorded and vividly displayed.This helps the peer reviewers to endorse or seek alternate flow for the goal as well as sub-goals.

1.2 Data structure, computations and m-D visualization

The progress of measurement, science and/or computational algorithms is interdependent.The physical (gravitational constant, electronic charge), chemical (atomic mass, radius of atom/ion, rate/equilibrium constant), and thermodynamic (G, H, S) constants are all single valued floating point scalar quantities.In matrix approach, each of them is an element of a matrix.In tensor notation a scalar is a zero order tensor. The UV-Vis, IR, spectrum is a vector of values equal to the number of wavelengths of measurements or in modern instrumentation sensors (like in diode array UV-VIS instrument). Considering full spectrum at different HPLC elution times, the measurement data is a matrix for each sample.The time delay-excitation-emission fluorescence spectrum is a third order-tensor measurement of absorbance values.The theoretical quest on one side and processing of measurements on other side smoothly has progressive transition of vector algebra to (extended) matrix and tensor (multi-way) algebra theorems and algorithms implementing standard methods of optimization, solving simultaneous algebraic/differential/integral equations and function approximation etc. The ignored aspect of geometric visualization of computational jargon now occupies a niche and a value added piece of information to further probe into micro details and point of start for newer vision in the discovery domain.Chemical sciences are not an exception to reap the benefits of tensor algebra approach of real/complex/quaternion numerical values in multi- (4-way) response data with first-/second- and third- order advantages.

Most of neural network literature was developed with algebraic notation with a few exceptions.Some of the software packages made use of object oriented programing jargon and cells instead of dimensional arrays. The main focus was to improve training procedures and extend them to NP-hard problems and also with recurrent connections. We used tensor notation (CT.Lab) for Kalman filter, biochemical equilibria, multi_variate-multi_response calibration in our chemometric activity during last two decades.

1.3 Regression (cause-response or effect relationships)

It is a rigorous statistical approach based on sound theoretical basis and consists of several varieties (chart A0-1). The main categories are bivariate, multivariate in explanatory variables/response variables/both, additive/ multiplicative, linear/non-linear in variables/parameters etc.From the structure of data viz. numerical (real) and their distribution category, binary, attributes, logical and so on different heads like binary/logistic/ Poisson/Binomial regression are at the forefront of research tools. Fuzzy regression is coveted if the errors in response are not probabilistic but originate from fuzzy intervals.

Computational TensorLab (CTLab), tensor laboratory for computations (TLC Thin layer chromatography, or laboratory for tensor computations (LTC) all mean transformation of data to knowledge. It is affected through display/ transformation (reduction, expanding) of data, formulae, equations, solution methods, parameters/ knowledge generation/ representation/ manipulationin algebraic, matrix and tensor (3-way, 4-way) modes. The two-way transformation of tensors into structure, objects (including classes) and solution by symbolic mathematics or simply 'evol' function of Matlab is now a trodden path with emerging tool-boxes.

Scope of present review: To start with, linear regression [1-164] comprising of one response and one explanatory vector of real numerical data follows.The necessary conditions of noise in data, parameter behaviour, model equations and constrained approaches are detailed. The failure conditions and remedial measures are described with simulated datasets. Simple as possible (SAP) matlab functions reflecting the formula translation into software is a white box approach. The knowledge bases in the form of passive if-then-rules of first order predicate logic and their implementation in m-programs and output is an expert system approach for numerical computations and generation of knowledge bits for conclusions, advices etc. The auto test modules take care of typical learnable data sets as ready reckoner for advanced training.

02. Linear Least Squares (LLS)

In linear least squares, the model considered is linear in parameters and in also variables.The estimation of parameters was well nurtured in all disciplines of science, engineering and social sciences in early 1970s.The basis of least squares is minimization of sum of squares of Euclidian distance (deviations) between observed (y) and model calculated(ycal) response vectors.With increased information on data precision and accuracy, more and more cases of non-adherence of simple linear model sprouted and non-linear least squares came in to picture. The axioms, limitations and attempts of circumventing hurdles are presented under the heads cited in chart (chart 2-1).The necessary conditions for application of linear least squares are programmed in Matlab, The output in object format (chart 2-2) are edited in chart 2-2(b) for formal browsing.

Chart 2-1:Linear Least squares - essentials

- Model
  - Necessary conditions
  - Assumptions
  - Mathematical form
  - Goal
  - Object Function
- Method
  - Derivation
  - Solution
    - Formulas in matrix notation
    - Matlab code
- SAP output
- Failure conditions
  - Remedial methods
    - Limitations

Chart 2-2: (a) Data structure

Matrix		
Data structure $x=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$; $y=\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$;	$Design\,matrix : X = [one\; x];$ $par \quad : \quad [a0\; a1]^T;$	y : Response/ dependent variable X : Explanatory/ causative /independent variable np : Number of points npar : No of regression parameters

MODEL

Algebraic notation	**Matrix form**
$Model:\; yi + normal\_noise = a0 + a1 * xi$	$Model:\; y + normal\_noise = X * par$ $y + normal\_noise = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix} * \begin{bmatrix} a0 \\ a_1 \end{bmatrix}$

2-2: (b) Method: 'Least Squares'		(c): Matlab function

<pre>   Least Squares Method
 ~~
--- 	
Necessary conditions
noiseX: 'Absent OR noise << x magnitude'
noisey: 'Normal distribution'
        'Homosedastic'
 	
 outliersx: 'Absent'
 outliersy: 'Absent'
SystErrorx: 'Absent'
 	
SystErrory: 'Absent'
 par: 'Adhere to normal distribution'
 	
DontCare: 'Don't careprofiles/
spacing of x or y'
       x: 'non-stochastic or
          deterministic'
       y: 'stochastic'</pre>
		<pre> %
		% NC_LeastSquares.m(R S Rao)20-10-
		201230-5-91
		%
		clean
		Method = 'Least Squares Method';
		st = {'Necessary conditions ';'Failure
		conditions';
		'Remedial Measures'};
		dispst(Method),
		
		 %%
		
		 %%
		
		noiseX = 'Absent OR noise << x
		magnitude';
		noisey = 'Normal distribution
		Homosedastic';
		par='Adhere to normal distribution';
		outliersx = 'Absent';
		outliersy = 'Absent';
		SystErrorx = 'Absent';
		SystErrory = 'Absent';
		MinorProcess = 'Absent';
		DontCare = ['Don''','t
		careprofiles/spacing of x or y'];
		x= 'non-stochastic or deterministic';
		y= 'stochastic' ;
		
		% NC.LS.Method = Method;
		NC.LS.noiseX = noiseX;
		NC.LS.noisey =noisey;
		NC.LS.outliersx = outliersx;
		NC.LS.outliersy = outliersy;
		NC.LS.SystErrorx = SystErrorx;
		NC.LS.SystErrory = SystErrory;
		NC.LS.par = par;
		NC.LS.DontCare= DontCare ;
		NC.LS.x= x ;
		NC.LS.y= y ;
		
		dash = '------------------------------
		------------------------';
		disp(dash),disp(st{1,:});
		
		disp(NC.LS)
		
		disp(dash),disp(' ')</pre>

Estimation of slope and intercept or regression parameters of a straight line: The formulas and corresponding MatLab code for slope and intercept of linear model in matrix notation is in Formulas 2.1. The derivation in algebraic notation and using differentiation of matrices/vectors are given appendix A1 and A2. This method is more precisely called unit weight linear least squares. Formula.lls.1 is a general equation in matrix notation applicable to estimate mean of univariate data, slope and intercept of a straight line, multiple linear model and polynomial regression straight. Of course, the vectors (structure) of design matrix (X) changes with the model. The same solution procedure is used in soft regression (PCR, PLSR) wherein PCs and PLSCs replace x.

Formulas 2.1:Parameters of bi linear least squares model		`%`
		`%Formulas_LS.m (R S Rao) 12/05/91; 10-10-15`
		`%`
		`function [par] = Formulas_LS(X,x,y)`
		`%`

$par = \left(X^T * X\right)^{-1} * X^T * y$	Formula.lls.1	`par = inv(X'*X)*X'*y;% Formula.lls.1`

DataSet 2.1: The simple as possible (SAP) data set simulated using y = 0 + 1 * x for six points is analysed with Formulas_LS.m. The intercept and slope obtained are 0 and 1.0. The residuals obviously are equal to zero. It is a deterministic model and not a stochastic one.

DataSet 2.1:	Command line execution	DataSet 2.2: y = 1 +2 *x +norm(mean,sdt)
Simulated data set for y = 0 + 1*x ;	>>x = [1:6]'; y =x; >>X =[ones(6,1),x]; >> [par] =Formulas_LS(X,x,y)	
>>[x,y] 1 1 2 2 3 3 4 4 5 5 6 6		~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ x y noisey ysimul ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ 1 3.0634 0.063372 3 2 5.1069 0.10693 5 3 6.988 -0.012026 7 4 9.0117 0.011674 9 5 11.031 0.030924 11 6 12.949 -0.050633 13 --------------------------
>>% Calling matlab function Formulas_LS.m >>[par,ycal,resid] = Formulas_LS(X,x,y)		
par = Expected noisey -0.0000 0.0 0.0 1.0000 1.0		Par_LLS Expected noisey 1.1025 1.00 0.05 1.9779 2.00
→ No noise ✓ Consequence: Regression parameters (slope and intercept are exactly equal to those used in model for simulation of data		→ Noise with sd of 0.05 → Consequence: Regression parameters are reasonable

Residuals in y (or response): The residual ($resid_i$) in y at ith point is the difference between measured response (y_i) and that calculated ($ycal_i$) from the model.$ycal_i$ is obtained from the estimated least squares parameters (a_o, a_i) as

$ycal_i = a_0 + a_1 * X_i$;$residy_i = y_i - ycal_i$.

This ordinary residual is a measure of unexplained variations in the response by the regression model. The standard deviation (scale parameter) and variance of sdy are calculated (Formulas-2.2).

Formulas-2.2:	`%`
	`%ordResid.m`
	`%`
	`function [ycal, resid,sdy] = ordResid(X,x,y)`
	`[par] = Formulas_LS(X,x,y)`

$ycal = X * a$
The residuals for all points are calculated.
$\text{residy} = ycal - y$
$if\ np\ is\ small$
$df = NP - Npar + 1$
$else$
$df = NP - Npar$
$\text{vary} = \frac{\sum_{i=1}^{NP}[ycal_i - y_i]^2}{df} = \frac{residy^T * residy}{df}$
$sdy = \sqrt[2]{\text{var}\ y}$
	`ycal = X * par ;`
	`residy = y - ycal ;`
	
	`[NP,Npar] = size(X);`
	`vary = residy'*residy/(NP-Npar)`
	`sdy = sqrt(vary);`
	
	ycal : Model calculated value of y
	vary : Variance in y
	residy : Residual (y-ycal)
	sdy : Standard deviation in y

$CovRsidy = vary * \left(X^T * X\right)^{-1}$	`varCovResidy = vary * inv(X'*X);`

DataSet 2.1(b): for y = 0 + 1*x		DataSet2.2: `y = a0 +a1 *x +norm(mean,sdt)`

`ycal =`
`1.0000`
`2.0000`
`3.0000`
`4.0000`
`5.0000`
`6.0000`
`resid =`
`1.0e-14 *`
`0.0666`
`0`
`-0.0444`
`-0.1776`
`-0.1776`
`-0.1776`
`>>`
		`ycal =`
		`3.0804`
		`5.0582`
		`7.0361`
		`9.014`
		`10.992`
		`12.97`
		`-----------------------------`
		`Residy noise added`
		`------------------------------`
		`-0.016979 0.063372`
		`0.048705 0.10693`
		`-0.048129 -0.012026`
		`-0.0023041 0.011674`
		`0.03907 0.030924`
		`-0.020363 -0.050633`

→ No noise ✓ Consequence: Residuals in y are zero (i.e. order of 10^{-14}. ✓ Is due to high floating point precision of matlab and hardware		→ The noise is homoscedastic Gaussian distribution of low standard deviation compared to range of y. + Consequence: The sd, t values of regression coefficients and sdy are all very low. + Least squares lowered the noise by minimising sum of squares of residuals in y + Passes through statistical tests

Standardized residuals: The ratio of residual in y to standard deviation is standardized residual (Formulas-2.3).

Formulas-2.3	
$\text{standResidy} = \frac{residy}{sdy}$	`standResidy = residy./sdy;`

.... X ---------------------- One x y residystandRes .... 1.0000 1.0000 2.0024 -0.0324 -1.0148 1.0000 2.0000 4.0800 0.0492 1.5402 1.0000 3.0000 6.0163 -0.0105 -0.3284 1.0000 4.0000 8.0312 0.0084 0.2627 1.0000 5.0000 9.9988 -0.0200 -0.6270 1.0000 6.0000 12.0202 0.0053 0.1673 !!	Resid = vary: 0.0010 sdy: 0.0319 scaleEstimate: 0.0319 % % standres.m % Standardized residuals % function [standRes] = standres(X,x,y) if nargin < 2, clean data_xy end [a,ycal,resid] = Formulas_LS(X,x,y); [ycal,residy,sdy] = ordResid(X,x,y,a); % standResidy = residy./sdy;

Parameter statistics: The estimated regression parameters are subjected to statistical tests to infer more about the success of regression for the analyzed dataset. The cumulated information by application of heuristic knowledge for statistics of parameters is of higher order compared to yester years' inspection of number with no recording of finer details.

Standard deviation of regression parameters: In case of bivariate data following a straight line relationship, SD in intercept and slope (Formulas 2.4) reflect the reliability of regression parameters, their co-variation, and confidence intervals.

Formulas-2.4

Matrix form

$$\text{sda} = \left(sqrt\left[\text{diag}\left\{ \left(X^T * X \right)^{-1} \right\} \right] \right) * \text{var}\, y$$ Formula

```
sda=
sqrt(diag(inv(X' *
X))) * vary;
```

If	npar ==2
Then	$SDa_0 = \sqrt{\dfrac{\sum (Y_i - YCAL_i)^2}{NP - NPAR}} * \sqrt{\dfrac{1}{\sum (X_i - \overline{X})^2}} * \sqrt{\dfrac{\sum X_i^2}{NP}}$ $SDa_1 = \sqrt{\dfrac{\sum (Y_i - YCAL_i)^2}{NP - NPAR}} * \sqrt{\dfrac{1}{\sum (X_i - \overline{X})^2}}$ Formula

Algebraic notation

Standard error of regression coefficients: The quotient of standard deviation of regression coefficient to standard deviation in y is called standard error (Formulas-2.5).

Formulas-2.5		
Formula	**Matlab code**	**Knowledge bits**
$\text{StandErra} = \dfrac{sda}{sdy}$	`standErra = sda/sdy`	
$\text{standa} = a * \dfrac{sda}{sdy}$	`standa = a.*sda/sdy;`	If: npar ==2 Then: $sda0 = a0 * sda0 / sdy$ $sda1 = a1 * sda1 / sdy$

Standardized regression coefficient:The numerical magnitudes of regression parameters do not reflect the relative importance of the explainable factors as they are scale dependent. But, the standardized regression coefficientsare scale independent.They are thus useful to interpret the relative importance of regression parameters (especially in multivariate X) in explaining the total variation in y.

t-values of regression parameters: The "t" statistic is computed by dividing the estimated value of regression coefficient by its standard error. It is a likelihood measure that calculated value of the parameter is not zero. Thet-values calculated are used to test null hypothesis i.e. estimated regression parameters are significantly equal to a zero or any expected values at 100*(1-α) % confidence level (for ex. 95% if α=0.05).The testing of null hypothesis for the significance of slope and intercept of the straight line are performed parameter wise.

Chart 2-3: Statistical hypothesis testing for slope and intercept

$$ta = \frac{a}{sda}$$

```
ta = a./sda;
ttable = t_table(alpha,NP-NPAR)
```

Intercept (a0)
H0_a0 :a_0 =a_0 expect
HA_a0:$a_0 \neq a_0$expect

Slope (a1)
H0_a1: a_1 =a_1 expect
HA_a0:$a_1 \neq a_1$ expect

$$\left.\begin{matrix} \dfrac{(a_0 - a_0 expect)}{(SDa0)} \\ \dfrac{(a_1 - a_1 expect)}{SDa1} \end{matrix}\right\} \textit{follows}\ \text{t(df=NP-2) distribution}$$

H0 : Null hypothesis: parameter is same as expected value;
HA: alternate hypothesis: parameter is significantly different from expected value

a0expect	:	Expected value of a0
a1expect	:	Expected value of a1

If	t < t-table(α, DF)
Then	H0 accepted
If	absolute (t_cal) > t-table value larger
Then	it less likely thatestimated value of parameter could be zero i.e. reg_coe > 0 with high probability [(1-alpha)*100%]

DataSet2.2

```
||||||||||||||||  statistics of regress parameters
     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    ---------------------------------------------------------------------------
             a,          sda,         standErra,       standa,      ta       t_prob
    ---------------------------------------------------------------------------
           1.1025       0.03873       0.93095        1.0263       28.465    9.0639e-06
           1.9779       0.009945      0.23905        0.4728       198.88    3.8345e-09
    ```````````````````````````````````````````````````````````````````````````
```

\\\\\\\\\\\\\\\\ t- statistics of regress parameters
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

| a, | ta | t-crit | ta>tcrit | alpha |
|---|---|---|---|---|
| 1.1025 | 28.465 | 4.604 | 1 | 0.05 |
| 1.9779 | 198.88 | 4.604 | 1 | 0.05 |

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

✓ Inferece: ta>tcrit =1 indicates regression

KB for t-table analysis for regression parameters
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

If abs(tvalue) < t_table value
Then H0 (: par = 0) is valid at alpha level
i.e. RegCoef is statistically insignificant

If abs(tvalue) > t_table value
Then HA (: par > 0) is valid at alpha level
i.e. RegCoef is statistically significant
```

coefficients are valid at 0.05 confidence level (for df_t = 4)	

DataSet 2-3:The t-statistic for a dataset of 18 data points shows that slope and intercept are significantly different from zero at 99.5% confidence level. The last column infers the statistical validity of the null hypothesis.

Reg par	Value	SD	t	t-table (α, DF)	t> t-table	Inference (statistical)
a0	1.415	0.218	6.47	1.75	t(6.4) >t-table(1.75) is true → H0 (a0 = zero) is False	a0 (1.41) is significantly >0
a1	0.6987	0.0089	7.78	1.75	t(7.8) >t-table(1.75) is true → H0 (a1 =zero) is False	a1 (0.69) is significantly >0
DF=NP-2=16, α=0.05						

	KB 2.1:Significant Reg_parameters
If	Smaller the value of Prob(t)
Then	Coeficient is more significant i.e.less likely that the actual value is zero
	Probability of acceptance/rejection of Reg_parameters
If	Prob(t) = 0.001
Then	Inference is thatthere is only 1 chance in 1000 forparameter being zero.
If	Prob(t) = 0.92
Then	92% probability that actual value of the parameter could be zero→ Elimination of term in regression function corresponding to this parameter does not significantly affect statistics
	Redundant/correlated explanatory variables
If	Redundant parameters, artefact of correlated variable in x
Then	Prob(t) = 1.00 (or nearer to 1.00)
If	Several parameters have Prob(t) values of(or closer to) 1.00
Then	check X matrix and parameter vector & repeat regression analysis

t-probability: The probability of t-values is computed using a two-sided distribution function (Formulas 2-6). It corresponds to probability of obtaining the estimated value of the coefficient when the actual coefficient is zero (KB 2.1, chart 2-3).

Thus, the derived statistics (t-values), table values/ probability throw light on inadequate/adequate and over ambitious models.

Example 2.1:If estimated value of a parameter is 1.0 and its standard error is 0.7, then the t value is 1.43 (= 1.0/0.7).If the computed Prob(t) value was 0.05, the inference isthat there is only a 0.05 (5%) chance that actual value of the parameter could be zero.

Application:Beer's law is an extensively used univariate calibration model in chemistry, bio-chemistry and many other scientific disciplines. The basic principle is the absorbance of a colored compound with concentration of analyte is a straight line.It passes through the origin when the blank solution has no absorbance or the absorbance is measured against the blank solution. However, in real life tasks the intercept is not exactly 0.000 but of small magnitude.In order to statistically establish that the intercept is not different from zero, point hypothesis testing is used.Similarly, the expected slope of the Hammett equation for variation of log k versus substituent constants is one.A large deviation is explained in terms of ortho-substitution.Here also, a regression parameter is to be tested against a fixed value.Further the regression model is valid only when the parameters are different from zero.

Number of data points and structure: Least squares analysis was practiced in applied sciences in last century with single digit (<=9) and rarely with 30 to 100 points.Numerical analysis, simulation studies with different distributions were carried out with larger number of data. The concern with experimental design (D, A, E etc.) in distribution of data points is of recent concern (Chart 2-4) when the benefit of designed experiments in pure sciences and industry came to light and instrumental and sampling procedures have become cost effective.

Failure conditions & Remedial measures:The variance at each point is generally not known as in many studies replicate measurements are not made in the entire range.Thus,Unit Weighted Linear Least Squares (UW LLS) is in routine practice.But, it is strictly applicable iff (if and only if) the normal noise in y is homoscedastic i.e. same variance for y values of all points in the data set. The non-homogeneous distribution of noise in y, outliers and/oranother process adhering to a linear model but with significantly different intercept and slope lead to unacceptable regression parameters (Chart 2-5).When it is diagnosed that derived data/parameters/sub-space is suffering with a mathematical ailment (artefact of sub-process, outliers, high noise), it is to be detected (diagnosed), reduce its effect by eliminating causes or bypassing the route (use of robust methods).

Chart 2-4: Distribution of x points

Equal interval	
Random	
ED	A, D, ….
D-optimal	FD, FFD, CCD,….
Kateman design	Lin, Quad

Chart 2-5:Failure conditions of UWLLSand remedial methods

LLS model		
Failure conditions	~~~~	**Remedial Measures**
– Heterosedastic noise in y	~~~~	Weighted LLS
– Outliers in y	~~~~	Least Median squares
– Non-normal noise in y	~~~~	MLE
– Noise both x and y	~~~~	Orthogonal-LMS
– Fuzzy errors	~~~~	Fuzzy Regression

```
%%
FC= {'Heterosedastic noise in y';
'Outliers in y';
'Non-normal noise in y';
'Noise both x and y ';
'Fuzzy errors'};


 RM= {'Weighted LLS';
'Least Median squares (LMS)';
' MLE';
'Orthogonal-LMS';
'Fuzzy Regression';
 } ;
 dash = '-------------------------------------------------------';
disp(' '),disp(dash)
```

```
disp('Failure conditions Remedial Measure')
disp(dash)
[nFC,col] = size(FC);
for i = 1:nFC
 zFC= FC{i,:};zRM = RM{i,:};
 x=[zFC,' ',zRM];
 disp(x)
end
disp(dash)
```

Errors in x and y: Sarabia et. al. proposed orthogonal least median squares regression (chart 2-6) to account for errors in both axes (noise-in-x and noise-in-y) and also in presence of outliers.It results in better sdy in prediction compared to orthogonal least squares (LSOrtho).

Chart 2-6: MODEL		
Algebraic notation	**Matrix form**	
$Model:$ $yi + noiseY_i = a0 + a1*(xi + noiseX_i)$	$Model: \; y + normal\_noise = X * par$ $$\begin{bmatrix} y_1 + ny_1 \\ y_2 + ny_2 \\ y_3 + ny_3 \end{bmatrix} = \begin{bmatrix} 1 & x_1 + nx_1 \\ 1 & x_2 + nx_2 \\ 1 & x_3 + nx_3 \end{bmatrix} * \begin{bmatrix} a0 \\ a_1 \end{bmatrix}$$	L A Sarabia, M C Ortiz, X Thomas; Anal. Chim. Acta.348 (2001)11-18; Performance of the orthogonalLeast median squares regression; J Riu, F XRius; J Chemomet., 9(1995) 343; Univariate regression models

lspar2015.m: This m-function (chart 2-7) calculates the regression parameters, simple residuals and their statistics. ccangsvd.m outputs correlation, angles between each of vectors (X and y) and singular values/ percentage explainablity of X matrix. CondMat calculates determinant, various scalar condition numbers of X. It shows matrix conditions like near-singularity, singularity etc. and guides for the choice of adequate inversion procedures. The listings of tabular display and graphics routines are not given for paucity of space.

Chart 2-7: Method flow and listings of m-files

```
``````````````````````````````````````````````````````````````````````````
 MethodFlow --lls2015 m file
~~~~~~
 Calculation ofregression parameters by least Squareslls2015

> Formulae for regression parametersFormulas_LS
> Ordinary residualsordResid(X,x,y,a_LS)
> Advanced residualsresidstat
> regression parameter statistics regcoefstat
> ANOVA Formulas_anova
 %
>> output: Tabular summary
Graphic display

```

```
%
%lspar2015.m (R S Rao)4/13/93, 10/27/1997,10/21/2011
%
```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

```
 ccangsvd% correlationCoef; angles; SVD;

[par] = Formulas_LS(X,x,y);% Reg parmaters
[ycal, resid,sdy] = ordResid(X,x,y,par);% Residual &sd in y

[sda,ta,standa] = regcoefstat(X,x,y);% Standar
deviation,
% t-value and
% standardized
reg parameters
```

```
%
%ordResid.m
%
function [ycal,residy,sdy] = ordResid(X,x,y,par)
if nargin < 3
clean
 usage('[ycal, resid,sdy]', 'ordResid','(X,x,y,par');
data_xy
y(6,1) = 10.;
 [par,ycal,resid] = Formulas_LS(X,x,y );
 par
end
 [par,ycal,resid] = Formulas_LS(X,x,y);
 ycal = X * par ;
 residy = y - ycal ;
 [NP,Npar] = size(X);
 vary = residy'*residy/(NP-Npar)
 sdy = sqrt(vary);


 Resid.residy = residy;
 Resid.vary = vary;
 Resid.sdy = sdy;

 Resid.varCovResid = vary * inv(X'*X);
 Resid.scaleEstimate=sdy;
 Resid
 Resid.varCovResid
```

```
%
% regcoefstat.m (R S Rao) 30-8-1993, 10/21/2011
%

function [sda,ta,standa] = regcoefstat(X,x,y)
%
if nargin < 3,
 clean
 data_xy
end
zzz= [];
[a,ycal,resid] = Formulas_LS(X,x,y );
[np,npar] = size(X);
vary = resid'*resid/(np-npar);
sdy = sqrt(vary);

%
```

```
%Statistics of regression coefficients
%
sda= sqrt(diag(inv(X' * X))* vary) ; % Standard deviation,
ta = a./sda;% tvalues,
standa = a.*sda/sdy;% standardized
standErra = sda/sdy% Standardised error

format shortg

 oo_regcoefstat
```

```
%
% oo_regpar.m
%
 disp('corrcoef([X,y])')
 zcc{n,:}= corrcoef([X,y]);
 zpar{n,:} = par;
 zsda{n,:} = sda;
 zta{n,:} = ta;
 zstanda{n,:} = standa;
 zresid{:,n} = resid;
 zsdy{:,n} = sdy;
 zycal{:,n}= ycal;
```

Constrained Regression -straight line through origin

In tasks like calibration, there is an apriori information that the intercept is zero.In linear least squares model of a straight line, the constraint (a0= 0) is implemented (chart 2-8).The point to be noted here is that the regression parameter is biased and is not BLUE (best linear unbiased estimate) in statistical sense, as the plot is forced through pass through zero on y axis.

Chart 2-8: Constrained linear univariate model		**Constrained Regression**
Matrix		
Data structure $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}; \; y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix};$	$Design\,matrix : X = x;$ $par \quad : \quad [a0];$	☞ Non-negative least squares ☞ Regression passing through origin ☞ Linear Regression with prefixed slope

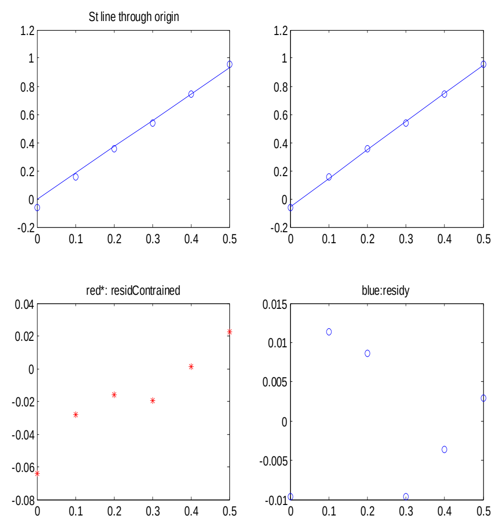
MODEL	
Algebraic notation	**Matrix form**
$Model: \; yi + normal\_noise = a1 * xi$	$Model: \; y + normal\_noise = X * par$ $y + normal\_noise = [x_1 \quad x_2 \quad x_3] * [a_1]$

	Y = a0 + a1*x + noise_n(mean,std) x; NP : 6

Constrained LS

x	y	resid
0	-0.063814	-0.0096803
0.1	0.15809	0.011325
0.2	0.3563	0.0086491
0.3	0.53891	-0.0096393
0.4	0.74588	-0.0035655
0.5	0.95325	0.0029115
	sdy =	0.031637

Caution:
sdy of contrained LS (0.031) >Unconstrained unit weighted (0.009)
It is valid if data acquisition system is more accurate than 0.002



LS through origin	Unconstrained LS	
Par sda a11.86130.0016195	Parsda a0-0.0541347.4341e-05 a12.00890.00024554	mean(noise) std(noise) ------------------------ -0.00189690.0092182 Obtained 0.00.01Desired
	→ Mean and std of noise desired and obtained are exact Since small number of noise points (NP =6) are simulated ✓ Sdy of LS (0.0091) = sdy of added noise (0.0092) i.e. LS almost extracts noise from data after modelling	

03.Univariate data

A vector of numerical values of duplicate/repeated measurements of response or an explanatory variable is the simplest univariate real numbers in one dimension (Chart 3-1).If the number of values are very small (<9), sample wise inspection serves the purpose to understand the variation. If the number of exceeds two digits (>99), a mathematical parameter (average) or statistical (arithmetic/geometric/harmonic) mean throws light on central tendency property of data. If the data set is in thousands to millions (simulation studies), visual inspection (graph/image) on different ranges/scales is the first step of exploratory analysis. The titbits of classical statistics viz. mean, standard deviation, their breakdown point and robust category (median) follow.

Mean

Mean is calculated as the quotient of sum of numerical values and number of observations (Formulas 3-1). Within the matrix algebra frame, it is the least squares estimator.

Chart 3-1:	Process
Deterministic process	$y = a0$
Parametric models of univariate data from processes	
NC	
Random process(Normal, lognormal,	$y = a0 + noise(distribution)$

exponential)	
If Random process is normal	$y = a0 + normal\_noise$
$x_i = x_{MEAN} + r_i + S_i$	Where r_i and S_i are normal and systematic errors.
If S_i is negligible, $X_i = X_{MEAN} + r_i$ If r_i follow normal distribution & Homosedastic Then $par = \frac{1}{NP} * \sum_{i=1}^{NP} y_i = mean\ (or\ average)$	

Least squares solution of $(y = a0 + noise)$**:** It is a parametric model for estimation of mean of samples perturbed by only random noise of much lower magnitude than the measured value. The design matrix is column vector of ones and the response is y vector. The mean being a least squares solution, it is an unbiased estimator and confidence levels are calculable.The formulas in matrix and algebraic notations is in Formulae 3.1.

Formulas 3-1 Mean of univariate data			
Process $y = a0$		Model $X * par = y + normal\_noise$	
Input data	Design matrix	Unknown perturbation	To be estimated
$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix};$	$X = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix};$	$noise = \begin{bmatrix} noise_1 \\ noise_2 \\ noise_3 \end{bmatrix};$	$par = [a0];$
The least squares solution is			
$par = (X^T * X)^{-1} * X^T * y$		$par = \frac{1}{NP} * \sum_{i=1}^{NP} y_i = mean\ (or\ average)$	

Scratch pad	
$X^T * X = [1 \quad 1 \quad 1] * \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = [3] \; = NP$ $(X^T * X)^{-1} = (NP)^{-1} = \frac{1}{NP}$	So, $(X^T * X)^{-1} * X^T * y = \frac{1}{NP} * \sum_{i=1}^{NP} y_i = average$
$X^T * y = [1 \quad 1 \quad 1] * \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = [y_1 + y_2 + y_3] = \sum_{i=1}^{NP} y_i$	

The basics of classical statistics postulates that population mean (μ)is obtained with infinite number of measurements.But, in practice, it means very large number.Coming to limits of experimental science,such a large number of experiments are cost and time prohibitive and small sample (NP<30) are in practice.

Standard deviation (SD): The SD of univariate numerical data set is calculated as the moment of mean about mean. It is a measure of dispersion and thus reflects spread of (process in time/replicate experimental) measurements. In the case of univariate data, SD throws light on dispersion from central tendency.Mean is subtracted from each observation, squared and summed for over entire data set. It is divided by degrees of freedom.Since mean calculated is used, one degree of freedom is lost. Thus NP-1 corresponds to DF of standard deviation. It inherits the positive features like confidence limits and at the same time the limitations viz. one outlier is the breakdown point of this statistic (KB 3-1, MatLabProg 3-1).

KB 3-1: Failure of mean → robust statistics

If	Noise is non-normal	Or
	Heteroscedastic	Or
	Outlier is present	
Then	Classical statistics fails	
	Remedy: Robust statistical methods	

If	SEDA	&
	Non-parametric method	
Then	Median Inter quartile range Spread	

Median

+	50% range of values are depleted of low/high numerical outliers
+	Not inflated like mean
+	Insensitiveto obliqueness of distribution
+	Extreme values

–	Break down point (failure)
–	Outliers > 50% of observations
	For mean > = one point

Central tendency and dispersion of univariate data

$$y = a0 + Normal\_noise\,(\text{homose}dastic)$$

Models --- Statistics

Parametric

If	Noise follows normal distribution	&
	Ei independent(not autocorrelated)	&
	Homoscedastic (equal variance)	&
	No trend	
	No outliers	
Then	Mean and standard deviation are unbiased estimators of central tendency	

non-parametric

If	High quality data acquisition	&
	Normal distribution not verified	
Then	Biweight method	

If	SEDA	&
	parametric method	&
Then	Mean SD	

If	Normal distribution confirmed	
Then	Mean is BLUE	
Else	Analyse with otherdistributions Chaotic profiles Discipline & process specific known signal profiles	
If	High quality data acquisition	&
	Process & sub-process knowledge	
Then	Six-sigma limits Ex: CERN experiments & results & hypotheses	

MatLabProg 3-1

```
%
% stats_univariate.m (R S Rao)10-1-16;
 19/12/05; 08/06/91
%
% Mean and SD
%

statsV.meanx = mean(x);
statsV.stdx= std(x);

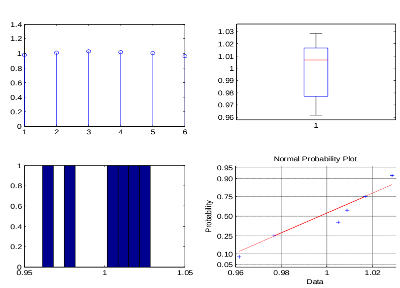
% MAD
statsV.medianx = median(x);

%Range
statsV.minx= min(x);
statsV.maxx= max(x);
statsV.rangex = (statsV.maxx-statsV.minx);
statsV
%
figure,plot(111),subplot(221),stem(x),subplot(222),bo
xplot(x), subplot(223),hist(x),
subplot(224),normplot(x)
```

Example 3.1: A simulated data set of six points with random noise of unit mean and standard deviation of0.02 is generated. The output of stats_univariate (chart 3-2) shows mean and median are very nearer (1.0018, 1.002) and with a very low sd (0.018).This is what is expected in absence of very large noise or outliers.

Chart 3-2: univariate data without outliers and low noise

```
x =
1.0259
0.9868
0.9967
1.0175
0.9762
1.0077
```

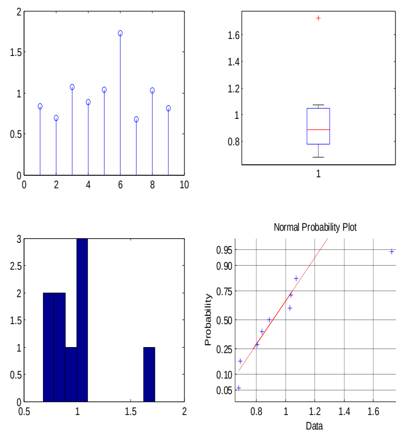



```
ans =
meanx: 1.0018
 stdx: 0.0188
medianx: 1.0022
 minx: 0.9762
 maxx: 1.0259
 rangex: 0.0497
```

Example 3.2: Here, nine points with mean 1 and high standard deviation (0.2) is analyzed. Box plot shows one of the points (1.72) different from others. The mean is 0.98 with sd of 0.32 (chart 3-3).

Chart 3-3: univariate data without outliers and high noise

```
Input is row vector!
 It is transformed
into column vector
~~~~~~~~
x =
0.8398
0.6906
1.0738
0.8890
1.0369
1.7262
0.6796
1.0312
0.8077
```




```
statsV =
meanx: 0.9750
 stdx: 0.3168
medianx: 0.8890
 minx: 0.6796
 maxx: 1.7262
 rangex: 1.0465
```

Outlier:An outlier is a datum very different in numerical magnitude from all other data points.If it is an artefact of transcription errors, it can be corrected.However many a time the outlier is the correct observation and the reasons are physicochemical in nature or instrumental spikes.A few instances in chemical science are logarithm of rate or equilibrium constants of ortho-substituted benzoic acids do not follow Hammett's straight line behaviour and exhibit an extremum in water and alcohol mixtures. The latter is due to predominant specific solute-solvent interactions.

Breakdown point of mean:Even one outlier inflates the mean which breaks down statistical character.The break down point (BDP) corresponds to percentage of outliers in the data that will not vitiate trend significantly.So, the BDP is zero for mean.

Remedy:Robust estimates of central tendency viz. median, S and Q measures have been put forward.

Median

Median is the second quartile or 50th percentile **(Chart 3-4)**.For a (univariate) dataset is in ascending order, median is the middle value if number of points is odd while it is the mean of the middle two values for even number of observations.Thus, approximately 50% of elements lie below and the other 50% lie above the median.MAD is aimed at symmetric distributions.

Chart 3-4: characteristics of median → Median detects outliers → MLE of centraltendency for Laplace distribution.	(b) Positive features and limitations + Robust to outliers – A biased estimator for normal distribution
Failure of median If Area of the tail is large(or) 50% or more of the observations are outliers Then Median fails	Remedy: Median absolute deviation
– Low (37%) Gaussian efficiency	Remedy / Gaussian Efficienty $S_r = 1.1926$ med / 5870

	Q: 0.25 quantile of distances | 8290
Median absolute deviation If Res/Med(res) > 5.0 Then Outlier	+ Breakdown point is 50% outliers

Median absolute deviation (MAD): It is the median of all absolute deviations from the sample median. It is a powerful tool to detect outlying observations. In chemical analysis any method with BDP > 20% is adequate and thus MAD and its derived parameters are sought after exploratory analysis.Here (1.0/0.6745) is correction factor consistent with usual scale of Gaussian distribution.

The program stats_univar.m (MatLabProg 3-2, chart 3-5) is useful for exploratory statistical analysis of univariate data.It calls many of the built in Matlab functions and robust method for central tendency in presence of outliers are incorporated.

Chart 3-5: matlab calling sequence MAD2015.m

```
.........Median statistic for detection of outliers in Univariate data with normal noise
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

AaAaAaAaAaAaAaAaAaAaAa Calling MAD_stats.m AuAuAuAu

``````````````````````````````````````````````````````````````````````
 Algorithmm file
~~~~~~
Repeat until no_outliers MAD_stats
 Cal parametric univariate statistics stats_univariate
 Sorts x into ascending order vector sortz
 calculation of MAD statistic om_MAD
 Detection of outliers through MAD
 Deleting outliers from x vectoroutlierRemoval
 New x vector generated excluding detected outliers
End repeat

MmMmMmMmMmMmMmMmMmMmMmMmMmexit from MAD_stats.m AuAu
```

MatLabProg 3-2: stats_univariate and MAD2015

```
%
% om_MAD.m(R S Rao 19/08/1991)
%
function [MedAbsDev_FromMed,ind] = om_MAD(x)
[np,c]=size(x); one = ones(np,1);
MAD_Limit = 5;
%
% calculation of MAD statistic
xmed = median(x);
DevFromMed = x-xmed*one ;
absDevFromMed= abs(DevFromMed);
Med_DevFromMed = median(absDevFromMed);
MedAbsDev_FromMed = [abs(DevFromMed)]/Med_DevFromMed;
%
% Detection of outliers through MAD
%
index_outlier = (MedAbsDev_FromMed >MAD_Limit) ;
```
~~~~~~

```
ind = (index_outlier)';


%
% calculation of
%
mean_absOfresidFromMean = mean(abs(x-mean(x)));
med_absOfresidFromMed = median(abs(x-median(x)));
% For mean absolute deviation
NormalScalePar_sigmaMean = 1.253*mean_absOfresidFromMean
% For median absolute deviation
NormalScalePar_sigmaMedian = 1.4826*med_absOfresidFromMed ;
```

```
%
% outlierRemoval.m
%(R S Rao 19/08/1991)
%
function [xNoOutlier] =
outlierRemoval(x,ind)
%
%Outlier removal
%
[np,c] = size(x);
for i =1:np
if ind(i) == 0
ind2(i) = 1;
elseif ind(i)==1
ind2(i) = 0;
end
end
x2 = x.*ind2';
k = 0;
for i = 1: np
if x2(i,1) < 1e-10
else
 k = k+1;
 xNoOutlier(k,1)= x2(i,1);
end
end
```

```
%
% MAD2015.m(R S Rao 10-1-16; 19/12/05; 08/06/91)
%
function MAD2015(x0)

if nargin ==0
clean
x0= [4.60 4.62 5.01 6.99 4.65 4.63 7.22]';
end
StepByStep_MAD2015
%


%
% column vector sorted in asceding order
xsorted=sortz(x0); x = xsorted;
disp(' ')
center02('..........Median statistic for detection
of outliers in Univariate data with normal noise');
%
StepByStep_MAD2015
%
outlier = 1;

% removal of outliers
whileoutlier
%
 [statsV ] = stats_univariate( x )
%detection of outliers by MAD statistic
[MedAbsDev_FromMed,ind] = om_MAD(x);
if (any(ind))
 outlier = 1;
 [xNoOutlier] = outlierRemoval(x,ind);
disp([' >>>>>>>>>>>>>>>> Outliers removed; ',
'&&&&&&&&&&&&&&& Median analysis repeated'])

clear x
x = xNoOutlier;
% x vector (excluding detected outliers is again
tested
% to make sure of absence of masked outliers)
else
outlier = 0;
disp([' >>>>>>>>>>>>>>>> No Outliers based MAD
statistic'])
center02(['!!!!!!! Advice: Inspect precision and
```

```
accuracy of data acquisition in the experiment' ]);
return
end
end
```

Example 3.3: The first four points are replicate values without any type of error, an ideal data set. Two high influencing outliers are added at 5th and 6th position of column vector. Obviously, mean and median differ (output 3-1).The deviation from mean show high values for fifth and sixth data points. But deviations from median are zero except for outliers. The MAD statistic clearly indicates them to be outliers.The LS solution gives a value of 2.3 for mean which is high.

In the next step,the function (removeOutliers.m) deletes these outliers. The program reanalyses for univariate statistics until no outliers (even masked ones) remain in the data vector. Now, the mean, median are exactly equal and deviations/standard deviation is zero. But, MAD is not a number (NaN) as the denominator of formula is zero. Here, box plot collapses to a straight line.

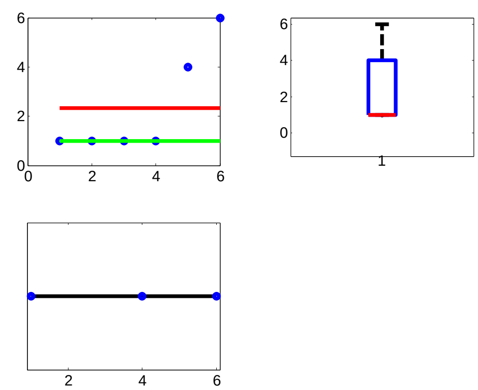
Output 3-1:Results of x0.m

Order : 0; NP :6

```
 |||||||||||||||||||||| Statistics-univariate data
```````````````````````````````````````````````````````````
 Residuals from
 ----
x MeanMedian Asc(abs(devMed))MADIndex
```````````````````````````````````````````````````````````
1.0000 -1.3333 0 0 NaN 0
1.0000 -1.3333 0 0 NaN 0
1.0000 -1.3333 0 0 NaN 0
1.0000 -1.3333 0 0 NaN 0
4.00001.66673.0000  3.00  Inf1.0000
6.00003.66675.00005.00Inf1.0000
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
meanx: 2.3333
medianx: 1
 stdx: 2.1602
 minx: 1
 maxx: 6
 rangex: 5
```

```
          Med_DevFromMed: 0
      indOutlier: [0 0 0 0 1 1]
   mean_absOfresidFromMean: 1.7778
    med_absOfresidFromMed: 1.7778
  NormalScalePar_sigmaMean: 2.2276
    NormalScalePar_sigmaMedian: 0
```




Red: Mean line : Green: Median; o: Data points
Box plot
Data on a horizontal vector

```
par =
2.3333

x xcalx-meanx

1 -1.33332.3333
1 -1.33332.3333
1 -1.33332.3333
1 -1.33332.3333
41.66672.3333
63.66672.3333
```

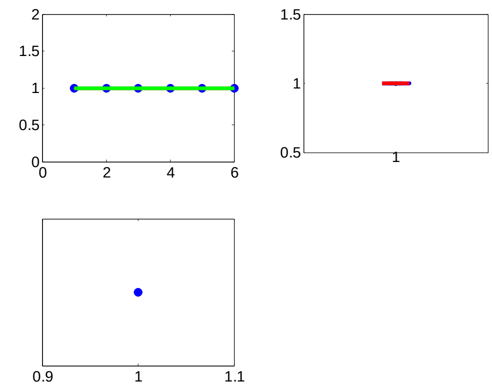
~~~~ xOutliers detected with MAD statistic

| Xoutlier | | | | | 4 | 6 |
|---|---|---|---|---|---|---|
| 1:outlier 0:not outlier | 0 | 0 | 0 | 0 | 1 | 1 |
| # | 1 | 2 | 3 | 4 | 5 | 6 |
| MAD | | | | | | |

```
>>>>>>>>>>>>>>>> Outliers removed; &&&&&&&&&&&&&&& Median
analysis repeated
```
~~~~

```
|||||||||||||||||||||||||||||||||||||Statistics-univariate
`````````````````````````````````````````````````````````
 Residuals from
 ----------------
 x MeanMedian Asc(abs(devMed))MADIndex
````````````````````````````````````````````````````````````
 1 0 0 0 NaN 0
 1 0 0 0 NaN 0
 1 0 0 0 NaN 0
 1 0 0 0 NaN 0
~~~~~~

MAD : [abs(deviations from median)]/median Of(abs(dev from
median)]
```



Green: Mean & Median;all data points are at dot
Box plot collapses to a mean line

```
meanx: 1
medianx: 1
 stdx: 0
 minx: 1
 maxx: 1
 rangex: 0
```

```
                        Med_DevFromMed: 0
 indOutlier: [0 0 0 0]
 mean_absOfresidFromMean: 0
 med_absOfresidFromMed: 0
NormalScalePar_sigmaMean: 0
NormalScalePar_sigmaMedian: 0
```

```
>>>>>>>>>>>> No Outliers based MAD
statistic

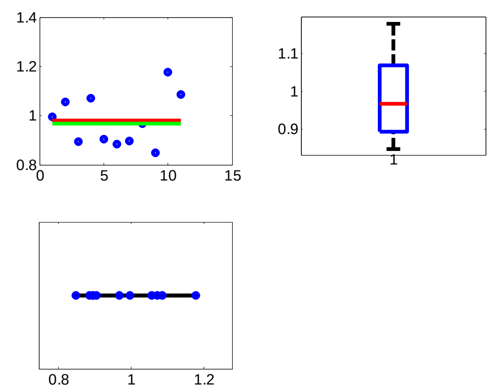
!!!!!! Advice: Inspect precision and
accuracy of data acquisition in the
experiment
```

Example 3.4: It is a small sample (NP: 11) data adhering to normal distribution (mean = 1.0 and sd = 0.01). The simulated data has a sample mean of 1.008 and sd of 0.0108 (output 3-2).

```
Output 3-2
|||||||||||||||||||||||||||||||||||||Statistics-
univariate data
 # x x-meanx x-medianXMAD
1.00000.99650.01610.0285
2.00001.05630.07590.0883
3.00000.8943 -0.0861 -0.0737
4.00001.07180.09140.1038
5.00000.9038 -0.0766 -0.0642
6.00000.8854 -0.0950 -0.0826
7.00000.8967 -0.0837 -0.0713
8.00000.9680 -0.0124 0
9.00000.8480 -0.1324 -0.1200
  10.000  1.17810.19770.2101
 11.0001.08540.10500.1174

Simulated noise_mean: 1; Std: 0.01; np:11
Obtained

 0.98040.10560.96800.0884
~~~No Outliers detected with MAD statistic
```

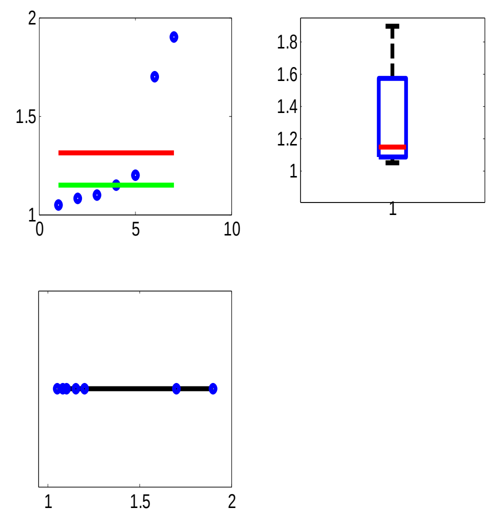


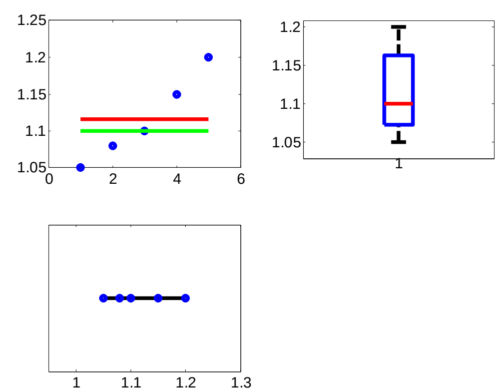

```
par =
0.9804
ans =
0.99650.01610.9804
1.05630.07590.9804
0.8943-0.08610.9804
1.07180.09140.9804
0.9038-0.07660.9804
0.8854-0.09500.9804
0.8967-0.08370.9804
0.9680-0.01240.9804
0.8480-0.13240.9804
1.17810.19770.9804
1.08540.10500.9804
```

Example 3.5: It is a real life measured dataset, but 6th and 7th points being outliers and MAD statistics detected them (output 3-3). The program removed and reanalyzed the remaining points.The ordinary statistics now coincide with robust parameters.

Output 3-3

```
Input is real numeric data
statsV =
NP: 7
meanx: 1.3114
medianx: 1.1500
 stdx: 0.3422
 minx: 1.0500
 maxx: 1.9000
 rangex: 0.8500
`````````````````````````````````````````````````
Residuals from
----
x Mean Median Asc
 (abs(devMed)) MAD Index
``````````````````````````````````````````````````
1.0500 -0.2614 -0.100001.4286 0
1.0800 -0.2314 -0.07000.05001.0000 0
1.1000 -0.2114 -0.05000.05000.7143 0
1.1500 -0.161400.07000 0
1.2000 -0.11140.05000.10000.7143 0
1.70000.38860.55000.55007.85711.0000
1.90000.58860.75000.7500 10.71431.0000
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
0.0700
 MAD : [abs(deviations from median)]/median Of
(abs(dev from median)]
statsV =
Med_DevFromMed: 0.0700
indOutlier: [0 0 0 0 0 1 1]
 mean_absOfresidFromMean: 0.2792
 med_absOfresidFromMed: 0.2792
NormalScalePar_sigmaMean: 0.3498
NormalScalePar_sigmaMedian: 0.1038
>>>>>>>>>>>>>>>> Outliers removed; &&&&&&&&&&&&&&&
 Median analysis repeated
 Input is real numeric data
statsV =
NP: 5
meanx: 1.1160
medianx: 1.1000
 stdx: 0.0594
 minx: 1.0500
 maxx: 1.2000
 rangex: 0.1500
`````````````````````````````````````````````````````````
Residuals from
----
x MeanMedianAsc(abs(devMed)) MADIndex
`````````````````````````````````````````````````````````````
ans =
1.0500 -0.0660 -0.050001.0000 0
1.0800 -0.0360 -0.02000.02000.4000 0
1.1000 -0.016000.050000
1.15000.03400.05000.05001.0000 0
1.20000.08400.10000.10002.0000 0
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
MAD : [abs(deviations from median)]/
 median Of(abs(dev from median)]
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```





```
statsV =
Med_DevFromMed: 0.0500
indOutlier: [0 0 0 0 0]
 mean_absOfresidFromMean: 0.0472
 med_absOfresidFromMed: 0.0472
NormalScalePar_sigmaMean: 0.0591
NormalScalePar_sigmaMedian: 0.0741
>>>>>>>>>>>>>>>> No Outliers based MAD
statistic
!!!!!! Advice: Inspect precision and accuracy
of data
acquisition in the experiment
```

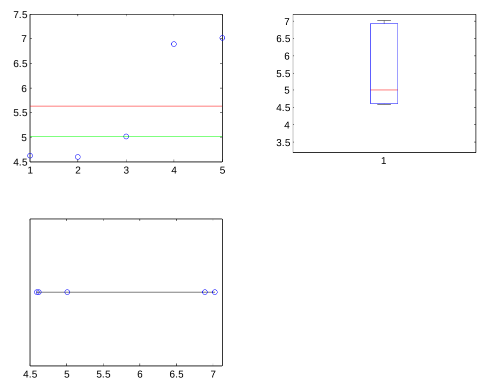
Applications:The outlier detection, adherence to normal noise, distribution free noise in chemical applications, graphics in hard/ soft regressions including inter-laboratory comparison studies will be the theme of a separate publication [164].

**Outliers (>50%NP)**

Example 3.6:A data set of five points with three outliers is an example of breakdown of MAD statistic(output 3-4). In the process of elimination of outliers, the analysis reaches a stage of two points.

Output 3-4: Results of data with outliers

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 Median statistic for detection of outliers in
Univariate data with normal noise
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Input is real numeric data
                              Phase I; NP =5
```




```
statsV =
NP: 5
meanx: 5.6480
medianx: 5.0100
 stdx: 1.2521
 minx: 4.6000
 maxx: 7.1200
 rangex: 2.5200
` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` `
Residuals from
-------------------
x MeanMedianAsc(abs(devMed)) MADIndex
` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` `
ans =
4.6000 -1.0480 -0.410001.0000 0
4.6200 -1.0280 -0.39000.39000.9512 0
5.0100 -0.638000.41000 0
6.89001.24201.88001.88004.5854 0
7.12001.47202.11002.11005.14631.0000
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ans =
0.4100
MAD : [abs(deviations from median)]/
median Of(abs(dev from median)]
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
statsV =
Med_DevFromMed: 0.4100
indOutlier: [0 0 0 0 1]
 mean_absOfresidFromMean: 1.0856
 med_absOfresidFromMed: 1.0856
NormalScalePar_sigmaMean: 1.3603
NormalScalePar_sigmaMedian: 0.6079
>>>>>>>>>>>>>>>> Outliers removed;
&&&&&&&&&&&&&&& Median analysis repeated
```

```
par =
5.6280
ans =
4.6200 -1.00805.6280
4.6000 -1.02805.6280
5.0100 -0.61805.6280
6.89001.26205.6280
7.02001.39205.6280

~~~~~~~~~~~ Outliers removed & analysis repeated
```

```
Phase II; NP =4

Input is real numeric data

statsV =
NP: 4
meanx: 5.2800
medianx: 4.8150
 stdx: 1.0898
 minx: 4.6000
 maxx: 6.8900
 rangex: 2.2900
``````````````````````````````````````````````````````````
Residuals from
-------------------
x MeanMedianAsc(abs(devMed)) MADIndex
```````````````````````````````````````````````````````````
ans =
4.6000 -0.6800 -0.21500.19501.0488 0
4.6200 -0.6600 -0.19500.19500.9512 0
5.0100 -0.27000.19500.21500.9512 0
6.89001.61002.07502.0750 10.12201.0000
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ans =
0.2050
MAD : [abs(deviations from median)]/
median Of(abs(dev from median)]
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
statsV =
Med_DevFromMed: 0.2050
indOutlier: [0 0 0 1]
 mean_absOfresidFromMean: 0.8050
 med_absOfresidFromMed: 0.8050
NormalScalePar_sigmaMean: 1.0087
NormalScalePar_sigmaMedian: 0.3039

>>>>>>>>>>Outliers removed; & Median analysis repeated
```

```
Phase III; NP =3
Input is real numeric data

statsV =
NP: 3
meanx: 4.7433
medianx: 4.6200
 stdx: 0.2312
 minx: 4.6000
```

```
 maxx: 5.0100
 rangex: 0.4100
```````````````````````````````````````````````
Residuals from
-------------------
x MeanMedianAsc(abs(devMed)) MAD   Index
`````````````````````````````````````````````````````
4.6000 -0.1433 -0.0200 01.0000 0
4.6200 -0.123300.020000
5.01000.26670.39000.3900 19.50001.0000
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ans =
0.0200
MAD : [abs(deviations from median)]/
median Of(abs(dev from median)]
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
statsV =
Med_DevFromMed: 0.0200
indOutlier: [0 0 1]
 mean_absOfresidFromMean: 0.1778
 med_absOfresidFromMed: 0.1778
NormalScalePar_sigmaMean: 0.2228
NormalScalePar_sigmaMedian: 0.0297

>>>>>>>>>Outliers removed; &
Median analysis repeated
```

```
                              Phase IV; NP =2
Input is real numeric data

statsV =
 NP: 2
meanx: 4.6100
medianx: 4.6100
 stdx: 0.0141
 minx: 4.6000
 maxx: 4.6200
 rangex: 0.0200
````````````````````````````````````````````````````````````````
Residuals from
----
x MeanMedianAsc(abs(devMed)) MADIndex
````````````````````````````````````````````````````````````````````
4.6000 -0.0100 -0.01000.01001.0000 0
4.62000.01000.01000.01001.0000 0
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ans =
0.0100
MAD : [abs(deviations from median)]/
median Of(abs(dev from median)]
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
statsV =
Med_DevFromMed: 0.0100
indOutlier: [0 0]
 mean_absOfresidFromMean: 0.0100
 med_absOfresidFromMed: 0.0100
NormalScalePar_sigmaMean: 0.0125
NormalScalePar_sigmaMedian: 0.0148

>>>>>>>>>>>>>>>>No Outliers based MAD statistic
```

```
!!!!!! Advice: Inspect precision and accuracy
 of data acquisition in the experiment
          ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

## 4. Regression robust to outliers

The representation of large number of points in a bivariate dataset by a few numbers of parameters is the focus of regression. This enables the reproduction of dataset within noise limits with the estimated regression parameters.Thus, output information of lls2015.m reflects the trends of majority of data points.The statistical reliability of parameters are ensured, iff (if and only if) the data adheres to necessary conditions. But, in many real life tasks outliers though in small number (10-20%) vitiate the estimation of slope and intercept of even straight line, mean of univariate data or multi(x1,x2) variate vs y models.

### Outliers

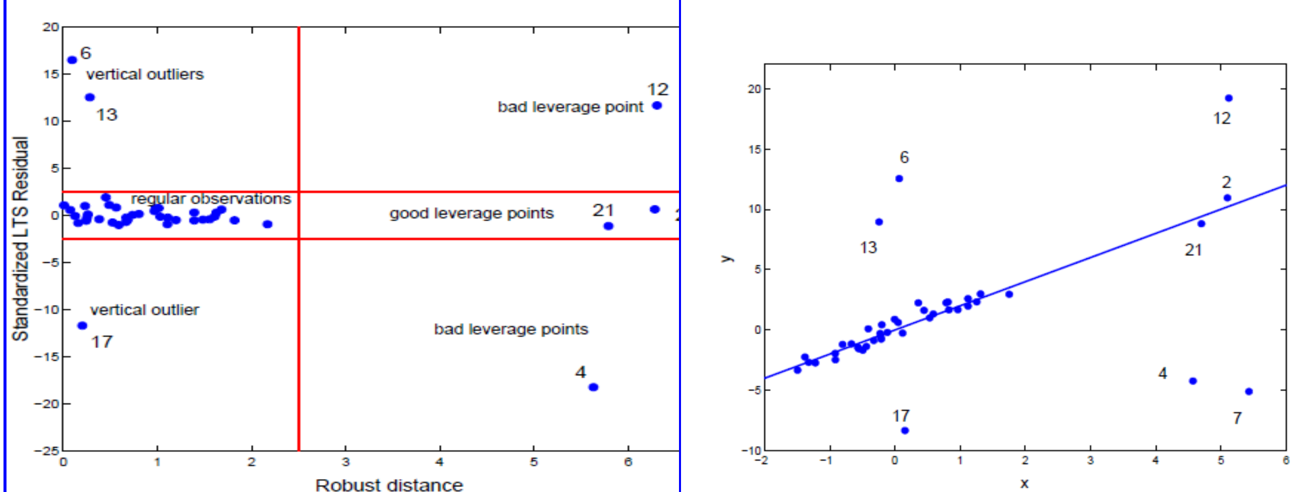
x-outliers:They are also called leverage points. X-outliers are those whose xi values are outlying i.e. The point (xi,yi) deviates from majority of x space covered by the data set(fig. 4-1).

Good-x-outliers: A good leverage point is one that follows the linear pattern of majority (points 2,21).

Bad-x-outliers: Bad leverage points are those which do not adhere to linear pattern of majority of points (points 4,7,12).

y-outlier or vertical outlier: The observation whose xi belongs to majority of x-space but the point deviating from linear pattern in the vertical direction is called vertical outlier (points 6,13,17)

Fig. 4-1: Leverage points and outliers




Courtesy of Ref :

- Observations with normal trend
- Leverage points
  - Good
  - Bad
- Outlier
  - Vertical
  - Horizontal

Failure of standardized residuals:The trend line (or plane or hyperplane) is attracted more towards the outliers and thus unit-weighted LS analysis does not represent the majority of data points. With ordinary least squares procedure, dataset even with outliers, the most of data points fall within -3 SD to +3SD horizontal cut off lines. The reason is SD is inflated by outliers.

Remedial Measure:Thus, robust methods to the presence of outliers (chart 4-1, KB. 4-1)) circumvent this hurdle.Median, robust (up to 50%) for outliers in central tendency has been in use and procedures based on this statistic have become pivotal in arsenal of cause-effect relationship analysis

| **Chart 4-1: Robust cause-effect methods** | |
|---|---|
| 🔔 | Single median method |
| 🔔 | Repeated median method |
| 🔔 | Least squares |
| 🔔 | Least median squares |
| 🔔 | Trimmed least squares |

| **KB. 4-1: Effect of multiple outliers on function of residuals in y** | |
|---|---|
| | |
| If | Multiple outliers in y direction |
| Then | Standardized residuals &<br>MD do not detect outlers<br>Expl: s and s(i) explode in presence of outlier |
| | |
| If | Multiple outliers in x direction |
| Then | Standardized residuals &<br>Transformed residuals do not detect outliers<br>Expl: ordinary Least squares pull LS fit more towards them |
| If | |
| Then | |

4.1 Least Median Squares (LMS):The method flow and algorithm of LMS2015 are given chart 4-2.

```
``````````````````````````````````  ``````````
Chart 4-2a: MethodFlow lms2015.m    m file
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~   ~~~~~~~~~~~
Regression parameters by            lms2015
least Median Squares

> Formulae for regression           Formulas_lms
 parameters
> Ordinary residuals                ordResid
> Scaled LMS residuals              scal_resid
%
>> output: Tabular summary          tab_lms1
2D-Graphic display                  gr_lms
---------------------------------   -------------
```

MatLabProg 4-1

```
lms2015.m (R S Rao 11/8/97,
09/06/94)
%
function [a_LMS] = lms2015(X,x,y)
 prin = 0
%
if nargin<4
 clean
 x = [1:4]'; y = 2*x;
 X = [ones(length(x),1) x];
 y(3,1) = 3;prin = 0
end
%
StepByStep_lms2015
%
[a_LMS] =
Formulas_lms(X,x,y,prin);
[ycal_LMS, resid_LMS,sdy_LMS] =
ordResid(X,x,y,a_LMS);
%
[np,npar] =size(X);
sc_LMSs = 1.4826*(1+5/(np-
npar))*sqrt(median(resid_LMS.^2));
sda_LMS = [];
%
 oo_lms2015
 tab_lms1,gr_lms1
%
```

| **Chart 4-2b: Algorithm and m program of LMS2015.m** | MatLabProg 4-2<br>`Formulas_lms2015.m` |
|---|---|
| | |
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

| | |
|---|---|
| For each set of points (NP_set = Npar) , regression parameters are calculated by solving deterministic equations | `cx = npar-1;`<br>`for i = 1: np-cx`<br>`for j = i+1 : np-cx`<br>`if j> np`<br>`else`<br>`if x(i) ~= x(j)`<br>` X1 = [x1]; set = set +1;` |
| NPar =1 $[a0]=\begin{bmatrix}1\end{bmatrix}^{-1}*[y1]$ | `if npar-1 == 0`<br>`       x1 = [1];`<br>`       y1 = [y(i);y(j)];`<br>`       zijk = [zijk;i ];`<br>`end` |
| | |
| NPar=2 $\begin{bmatrix}a0\\a1\end{bmatrix}=\begin{bmatrix}1\ x11\\1\ x21\end{bmatrix}^{-1}*\begin{bmatrix}y1\\y2\end{bmatrix}$ | `if npar-1 == 1`<br>`x1 = [X(i,:);X(j,:)];`<br>` y1 = [y(i);y(j)];`<br>` zijk = [zijk;i j ];`<br>`end` |
| | |
| In the case of bivariate-linear LS, slope and intercept are obtained from a pair points.<br>NPar = 3 $\begin{bmatrix}a0\\a1\\a2\end{bmatrix}=\begin{bmatrix}1\ x11\ x12\\1\ x21\ x22\\1\ x31\ x32\end{bmatrix}*\begin{bmatrix}y1\\y2\\y3\end{bmatrix}$ | `if npar-1 == 2`<br>`x1 = [X(i,:);X(j,:); X(j+1,:)];`<br>` y1 = [y(i);y(j); y(j+1)];`<br>` zijk = [zijk;i j ];`<br><br>`end` |
| | |
| NPar =4 $\begin{bmatrix}a0\\a1\\a2\\a3\end{bmatrix}=\begin{bmatrix}1\ x11\ x12\\1\ x21\ x22\\1\ x31\ x32\\1\ x41\ x42\end{bmatrix}*\begin{bmatrix}y1\\y2\\y3\\y4\end{bmatrix}$ | `if npar-1 == 3`<br>` x1 = [X(i,:);X(j,:); X(j+1,:);X(j+2,:)];`<br>` y1 = [y(i);y(j); y(j+1,:);y(j+2,:)];`<br>` zijk = [zijk;i j j+1];`<br>`end` |
| | |
| $ycal = X * a$ Formula. Lms.2<br>The residuals for all points are calculated.<br>$Residy = ycal - y$ Formula. Lms.3<br>$RES2 = RES.*RES$ Formula. Lms.4<br>The median of the squares of residuals is calculated.<br>Med_Res2 (:,1)= Med(Res2) Formula. Lms.5 | `a = X1\y1; %Formula lms.1`<br>` ycal = X * a; %Formula lms.2`<br>` resid = X* a - y; %Formula lms.3`<br>` res2 = resid.^2; %Formula lms.4`<br><br>` med_res2= median(res2);`<br>`              %Formula lms.5` |
| | `ij = [i j];`<br>` zij = [zij,[i;j]];`<br>` za = [za;a'];`<br>`%`<br>`zssa = [zssa;med_res2 a'];`<br><br>`end`<br>`end`<br>`end%j`<br>`end% i` |

| | |
|---|---|
| The minimum of medians of squares of residuals is found<br><br>Parameters of LMS are those corresponding to Minimum(Med_Res2) | `zz2 = sortz(zssa);  %first row is minimum of med_res2`<br>` [r,c] = size(zz2);`<br><br>`a_LMS = zz2(1,2:c)';  %Formula.lms.6` |

Applications: LMS has been extensively used in chemistry, electrical engineering, process control, computer vision and finance over the last three decades.

Example 4.1:A three point simulated data set of model y = 2*x with one y outlier is analyzed with lms2015.m. The estimated parameters (a0 = 0; a1=1) are exactly same as model ones even in presence of outlier. For the same data ordinary least squares (lls2015.m) outputs (a0 = 2; a1=0.5) which are wrong (output 4-1). It is consequence that least squares drag the regression line to minimize squares of Euclidian distances. But the parameters are unreliable as is evident from their standard deviations (sda0 = 2.2 and sda1=1.06). But sdy indicates LLS model (sdy_LLS: 1.2) is far less than that for LMS (3). A close examination shows that residual is (-3) for outlier (y= 3 for x = 3), while the residuals for the other two points are zero. It means that the procedure not only detects outlier, but also prevents its effect on slope and intercept of best straight line without eliminating it from dataset. It is all in considering median which is robust to 50% of outliers. The details of LMS calculation shown in Table 4-1.

| Output 4-1:Example 4.1 | Outliers : 1; NP :3 | y = 2*x;NP:3;<br>#outliers :1; no noise ; |
|---|---|---|
| ---------------------------------------<br>a_lms a_lls sda_lls<br>---------------------------------------<br>02 2.2913<br>20.5 1.0607<br>sdy_lms :3 sdy_lls : 1.2247 | k x y res_lms res__lls<br>.................................<br>1120 -0.5<br>22401<br>333-3 -0.5<br>................................ | − 33% outliers<br>+ LMS finds correct solution |

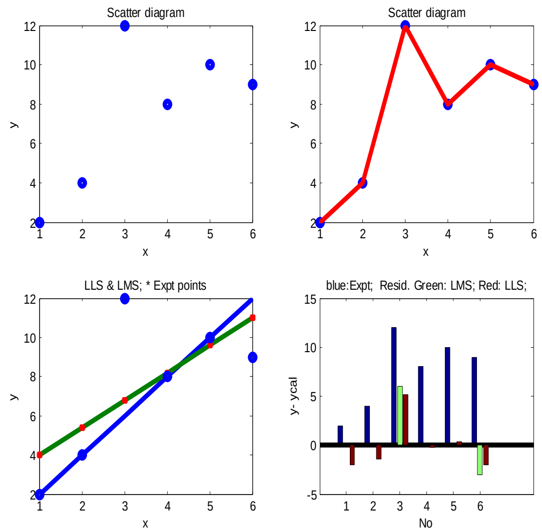
Table 4-1:Details of parameter estimation with LMS

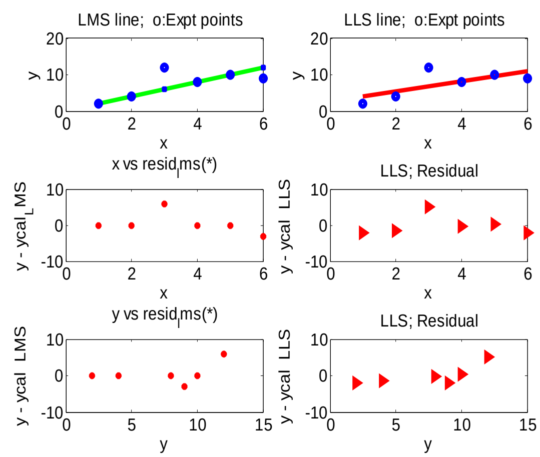
| i ,j | X | a0 | a1 | res2 | zmed | a_lms |
|---|---|---|---|---|---|---|
| 1 2 | 1 1<br>1 2 | 0 | 2 | 0<br>0<br>9<br>0 | 0 | 0<br>2 |
| 1 3 | 1 1<br>1 3 | 1.5 | 0.5 | 0<br>2.25<br>0<br>22.25 | 1.125 | |
| 2 3 | 1 2<br>1 3 | 6 | -1 | 9<br>0<br>0<br>36 | 4.5 | |

Example4.2:It is similar to example 4.1, but with six data points (output 4-2). The y_outliers (also called vertical outliers) are in positions 3 and 6.The outliers are visually clear from scatter diagram. The bar diagram of experimental points, residuals by LMS and LLS represent functioning of two methods in presence of outliers. The residuals in y versus x and residuals versus y adds information of model fit.

| Output 4-2: Example 4.2 | Outliers : 2; NP :6 |
|---|---|
| ---------------------------------------<br>a_lms a_lls sda_lls<br>---------------------------------------<br>0 -0.13333 1.8929 | ..............................................<br>kxy res_lmsres_lls<br>..............................................<br>11200.33333 |

| | |
|---|---|
| 21.80.48606<br>sdy_lms :1.8028<br>sdy_lls : 1.4259 | 22400.53333<br>333-3-2.2667<br>44800.93333<br>551001.1333<br>6610 -2 -0.66667<br>......................... |
| | y = 2*x; NP:6;<br>Noutliers :2; no noise: |




Features of LMS:The presence of outliers increases the magnitude of residuals of LLS model, but they are within 3SD limits. This is an artefact of increased standard deviation of residuals for entire data set.With LMS model, the residuals of outlying points are very high, but residuals for all other data are very low compared to LLS. But, when outliers are deleted, it is obvious that parameters and statistics are same or almost same for LMS and LLS.

The regression parameters of LLS adhering to necessary conditions are BLUE (best linear unbiased estimators). This combination of LMS to detect outliers and LLS to calculate parameters is a popular hybrid method (chart 4-3).

**Chart 4-3: LMS algorithm progress and implementation in commercial software packages**

| | | |
|---|---|---|
| 1984 | LMS algorithm | Rouaawwuw |
| 1987 | Resampling alg (PROGRESS) | Rouaawwuw |
| 1986 | Cal of regression coefficients | Steele and Steiger |
| 1993 | Cal of regression coefficients | Stromberg |
| 1997 | Branch and bond alg in selection of sub-sets of points | Agykki |

| LMS |
|---|
| h = NP+npar+1/2 |
| Highest possible breakdown = {(NP- npar)/2 +1}/NP |

| Software | Program |
|---|---|
| S-Plus | lmsreg |
| SAS/IML | LMS |

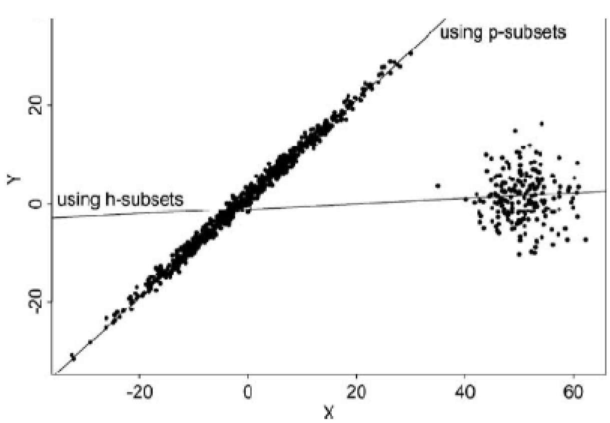
Failure of LMS

Example 4.3:The data for model in example 4.1 (y = 2*x) is simulated but with outliers in different positions. LMS failed to find arrive at correct parameter values(output 4-3). This is an artefact of very small number of points (NP=3) although number outliers are 33%.

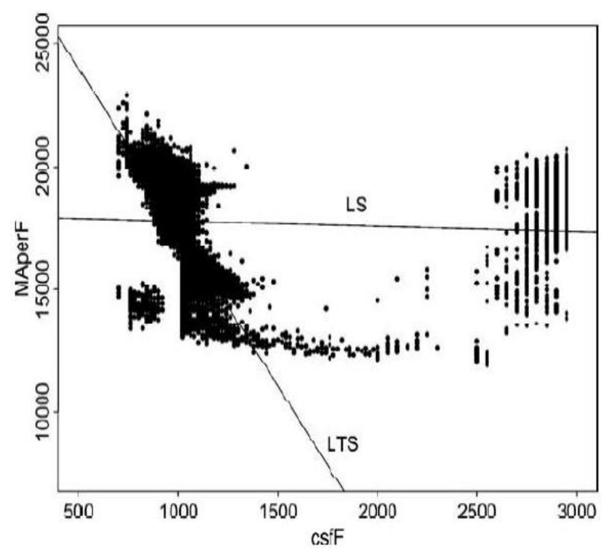
| Output 4-3: Example 4.3 | | NP : 3; #outliers :1; no noise: |
|---|---|---|
| ---------------------------------------<br>a_lms a_lls sda_lls<br>---------------------------------------<br>1 -0.33333 1.0184<br>12 0.4714<br>---------------------------------------<br>sdy_lms :2<br>sdy_lls : 0.8165 | .........................<br>kxy res_lmsres_lls<br>.........................<br>11200.33333<br>2230 -0.66667<br>33620.33333<br>.........................<br>sdy | LMS fails to find correct solution<br><br>Position of outlier has a role |
| | | |
| ---------------------------------------<br>a_lms a_lls sda_lls<br>---------------------------------------<br>2 1.33330.25459<br>11.50.11785<br>---------------------------------------<br>sdy_lms :1<br>sdy_lls : 0.40825 | ..................................................<br>kxyres_lmsres_lls<br>…...............................................<br>11300.16667<br>2240 -0.33333<br>33610.16667<br>.................................................. | LMS fails to find correct solution<br>Position of outlier has a role |

## 4.2 Least Trimmed Squares (LTS)

The dataset of NP points is divided into h subsets with coverage between NP/2 and NP. The LS parameters and residuals for each subset are calculated. The LTS parameters correspond to those of a subset (h) whose sum of squared residuals is minimum. However, CPU time grows with data size and number of subclasses.The sub-classification of outliers due to Rousseeuw consists of vertical/horizontal and leverage points. The leverage point is further divided into good and bad ones. In reality or even simulated datasets, all these types are not present in every case-study.The new algorithm is faster than other methods even for tens of thousands of points. Chart 4-4 describes the results of LTS for typical simulated and real life data sets.The algorithms, Matlab functions and KBs from pedagogic stand point will be reported [164] separately in hot-ice series.

| Chart 4-4: output of LTS analysis<br><br>Simulated Data set<br><br>→ First 800 points are simulated randomly (Eqn. 3.2.1). The added noise is from standard normal variate.<br>→ The second set (801 to 1000) is from bivariate normal distribution<br>→ Now 1000 data points are divided into 501 subsets.<br>→ LTS (Data) | <br> |
|---|---|
| Inference<br>– With initial h subsets bad leverage points influence regression. Wrong LTS line<br>+ p-subsets yield correct LTS line with same reg coefficients | $y_i = 1 + x_i + nr_i$ ; $NP = 1000$; Eqn. 3.2.1<br>$h = \frac{[NP + npar + 1]}{2}$<br>$= \frac{1000+2+1}{2} = 501$ |

| | |
|---|---|
| Courtesy ofP J Roussseeuw,K V Driessen, Data mining and knowledge discovery, 12(2006)29-45 | $\left.\begin{array}{l} x_i: \; norm(0,100); \\ nr_i: \; norm(0,1)\;; \end{array}\right\} i=1:800$ <br> $\left.\begin{array}{l} \text{bivariate}\; normal\; distribution \\ mean=(50,0) \\ sd \quad =25*\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \end{array}\right\} i=801:1000$ |

| | |
|---|---|
| (b) Spectroscopic data on F band characteristics of 56, 744 stars |  |
| y: MAperF (Aperture magnitude, a brightness measure in F-band) <br> x: scfF <br> npar = 2 | |
| Inference <br> – ordinary LS fails - x-outliers makes LSline into parallel to x-axis <br> + LTS-line correctly shows decreasing trend of majority of starts -- The bunch of x-otliers appear as they are far away from LTS-fitted down ward trend line | |



| FAST_LTS algorithm | | |
|---|---|---|
| standard datasets | large data sets | x-outliers |

| | NP | Npar |
|---|---|---|
| Heart | 12 | 3 |
| phosphor | 18 | 3 |
| Coleman | 20 | 6 |
| wood | 20 | 6 |
| salinity | 28 | 4 |
| aircraft | 23 | 5 |
| delivery | 25 | 3 |

large data sets
#outliers (40%)

| NP | Npar |
|---|---|
| 100 | [2,3,5] |
| 500 | [2,3,5] |
| 1000* | [2,5,10] |
| 10,000 | [2,5,10] |
| 50,000 | [2,5] |

*: 35% outliers

x-outliers

$$\left.\begin{array}{l} x_i: \; norm(0,100); \\ nr_i: \; norm(0,1)\;; \end{array}\right\} i=1:800$$

$$x\_outliers$$

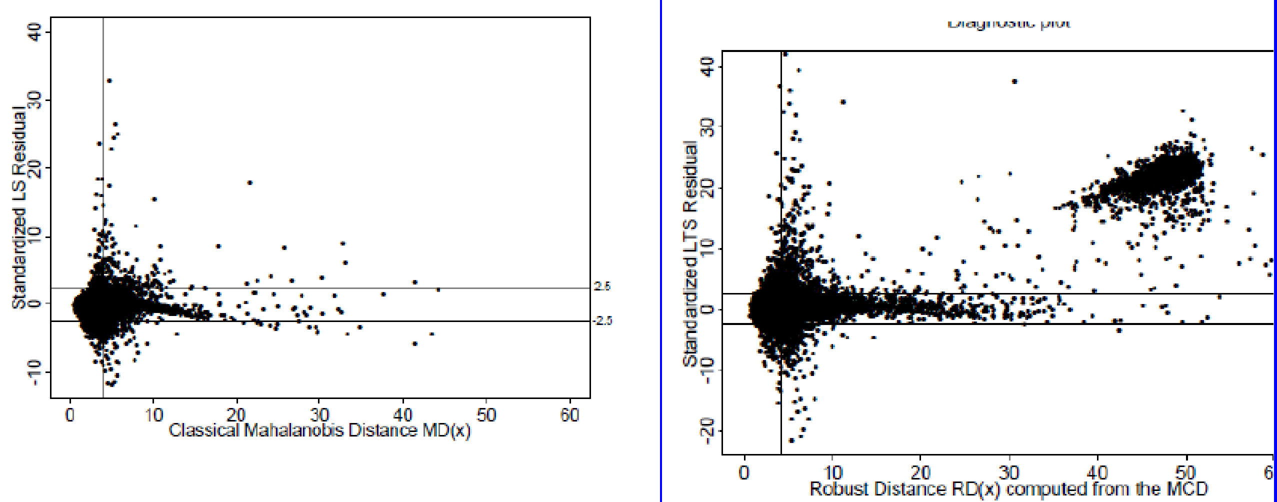
**Residuals from LS and Robust regression**

Robust residuals versus robust distances:Van Zomeren (1990) proposed a graphic display of ratio of residuals to standard deviation versus robust distances (Fig. 4-2).The vertical and horizontal cutoff lines discriminate outliers as different categories

Standardized LTS distance versus robust distance: The second cluster corresponds to larger subset of observations with large robust residuals and also with large robust distances.In the accepted terminology, they are bad leverage points. But, the ground truth is that they correspond to giant stars which have altogether different behaviour from others.

Thus, here the outliers correspond to another process/phenomenon and the lacuna lies in combing data sets belonging to different clusters, each of which are homogeneous with linear trend. The combination resulted in a heterogeneous outcome.

Fig. 4-2: Residual analysis




## Least Absolute deviations (LAD)

The minimization of sum of absolute of residuals (Eqn. 04.1) is referred as LAD. It is also called least absolute errors (LAE), least absolute value (LAV), least absolute residual (LAR), or sum of absolute deviations.In order words, it is finding L1-norm, remembering that least squares solution uses a L2-norm. The necessary conditions, data structure and model are same as that of LLS. There is no analytical solution for object function and thus no straight forward way to obtain optimum parameters of model.So, it is transformed into a linear programming format and solved with the iterative methods (table 4-2). The algorithm consists of addition of a pair of unknown (so called slack) variables. The features of LAD are compared with LLS in chart 4-5.Alternate ways of solving LAD are considering it as quantile regression and FMINUNC (Optimizaation toolbox) or ROBUSTFIT (statistics Toolbox).

Chart 4-5: Object function and goal in LAD

$$objFn\_LAD = \sum_{i=1}^{NP} |y_i - Fn(x_i)| = \sum_{i=1}^{NP} |residy_i|$$

$$Goal = \min(objFn\_LAD) \qquad \text{Eqn.04.1}$$

$$= \min\left(\sum_{i=1}^{NP} |y_i - Fn(x_i)|\right) = \min\left(\sum_{i=1}^{NP} |residy_i|\right)$$

$$= min\left[sum\{abs(residy)\}\right]$$

This had no direct solution

Constrained solution for LAD

$$y = X * par + u - v$$

For single x variable

$$y_i = a0 + a1 * x_i + u_i - v_i$$

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \geq 0 \qquad v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \geq 0$$

With these equality constraints , the object function becomes, the goal is

$$Goal\_LAD = \min(objFn\_LAD\_Transfored = \min\{sum(u) + su(v)\} \qquad \text{Eqn.04.2}$$

Total numbers of unknowns = regression parameters + u and v vectors
= 2 + 2*NP

Ref: Ref: Errico

**Table 4-2: Iterative methods for LAD**

- Simplex-based methods
  - Barrodale-Roberts algorithm
- Iteratively re-weighted least squares
- Wesolowsky's direct descent method
- Li-Arce's maximum likelihood approach

**Chart 4.5b: Features of LAD vs LLS**

| LAD regression | LLS |
|---|---|
| + Robust | − Not very robust |
| − Unstable solution | + Stable |
| − Possibly multiple solutions | + Always one solution |

| LAD2015.m calls matlab built-in function linprog.m | | |
|---|---|---|
| LAD2015 | ← | linprog |

MatlabProg 4-3

```
%
%LAD2015
%
%%
function [coef_LAD,stats]= LAD2015(X,x,y)
 %%


if nargin < 3
clean

x = sort(rand(100,1));
nr2 = rand(size(x))-.5;
y = 1+2*x + nr2;
X= x;

x = [1:6]';[np,c] = size(x);

nr2 = noise_n(np,0.,0.02);
y = 2*x + nr2;X=x;[X,y,nr2]
end
[X,y]
```

```
%%
(n,nvar) = size(x)

% our objective sums both u and v, ignores the
regression
% coefficients themselves.
[np,col] = size(x);
nr2 = zeros(np,1);
objFnLAD =[0;0; ones(2*np,1)];%f = [0 0 ones(1,2*n)]';

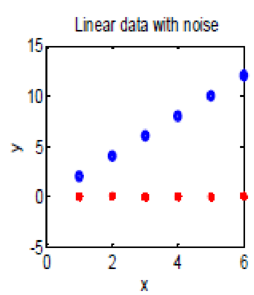
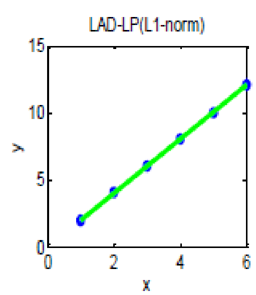
% a and b are unconstrained, u and v vectors must be
positive.
LowerBound = [-inf; -inf ; zeros(2*np,1)]'; %LB = [-inf
-inf , zeros(1,2*n)];
% no upper bounds at all.
UpperBound = [];

% Build the regression problem as EQUALITY constraints,
when
% the slack variables are included in the problem.
Aeqn = [ones(np,1), x, eye(np,np), -eye(np,np)];% Aeq =
[ones(n,1), x, eye(n,n), -eye(n,n)];
beqn = y;

% estimation using linprog
par_LP =
linprog(objFnLAD,[],[],Aeqn,beqn,LowerBound,UpperBound);

% we can now drop the slack variables
coef_LAD = par_LP(1:2);
%out99
gr_lad2015
%
oo_lad2015
```

Dataset 4-1: A simulated linear data with added noise is used to calculate intercept and slope of the straight line with LAD (output 4-4).

| Output 4-4: DataSet 4-1<br><br>-------------------------<br>a_LAD<br>-------------------------<br>-0.0217 intercept<br>2.0111 Slope |   |   | ..................................<br>x y res_LAD<br>..................................<br>1.00001.9893 -0.0000<br>2.00004.05360.0531<br>3.00006.02460.0130<br>4.00008.0134 -0.0092<br>5.0000 10.007 -0.0259<br>6.0000 12.044 -0.0000<br><br>Sdy_LAD 0.0268 |
|---|---|---|---|
| | Blue: simulated data;<br>Red:normal noise | Green: LAD line;<br>Blue: data with noise | |

Typical literatue reports in development of robust regression methods and their applications are described in table 4-1.

Table 4-1: Recent advances and applications of LAD

| Data with | Applied in |
|---|---|
| – Heteroscedastic interval<br>– Censoring left truncation | |
| – For longer-tailed error distributions and outliers | |
| | +Blind arma |
| **Data with** | **Applied in** |
| – outliers such as deep valleys | Robust against outliers |
| – large heterogeneous noise | Fundamental matrix: algebraic representation of epipolar geometry<br>Epipolar geometryis the intrinsic projective geometry between two views |
| – Multiple change points occurring at unknown times | Estimation of multiple-regime regressions |
| | semiparametric model with longitudinal data |
| | Robust Binary regr |
| | Linear inequalities<br>Inconsistent systems |
| | linear and mixture linear errors-in-variables regression models |
| – Autoregressive time series | |
| | Minimizing the maximum of a weighted sum of absolute deviations |
| – Serial correlation<br>– Nonnormal<br>– Outliers<br>– Autocorrelation | time series regression |
| | |
| – Fuzzy input<br>– Fuzzy output | Fuzzy multivariate regression models |
| – outliers in the response variable | Inverse least absolute deviations regression |

| | |
|---|---|
| | MLR |
| – Number of upper or lower outliers in normal sample | Parameters estimation in real time |
| | |
| – Single time series | Box-Jenkins models<br>multiplicative seasonal moving average model<br>monthly rice sales data and to US airline passenger data |
| | Limiting behavior of least absolute deviation estimators for threshold autoregressive models |
| – Heavy-tailed innovation | ARCH-type model |
| | Huber loss<br>Iteratively reweighted least squares algorithm<br>Tukey loss |
| – Outliers | SVM model |
| | Object tracking<br>Corruption modelled as a Laplacian distribution<br>LAD–Lasso optimisation model proposed based on Bayesian Maximum A Posteriori (MAP) estimation theory |
| – Trapezoidal fuzzy number | Fuzzy regression model |
| | Short term forecasting |
| | Robust variable selection procedure |

Table 4-1b: Recent advances and applications of hybrid_LAD, LMS and LTS

| | Method | Data with | Applied in |
|---|---|---|---|
| Fuzzy | LAD | | Linear problem |
| Moving | LAD | – outliers | Weighted median problem<br>Global data approximation |

| | | | |
|---|---|---|---|
| Non-linear | LAD | – Unevenly distributed data errors about the function | ✈ Non-linear least absolute deviation |
| Non-parametric | LAD | – Regressor and error term are contemporaneously correlated | ✈ Nonparametric estimation in a nonlinear cointegration model |
| Penalized | LAD | – Cauchy noise distributions | ✈ High dimensional sparse regression<br>✈ Npar>Np<br>✈ Does not need any knowledge of standard deviation of the noises or any moment assumptions of the noises. |
| Stepwise penalized | LAD | – Outliers in the response variables<br>– Heavy-tailed distributederror | ✈ asymptotic normality of the index parametric estimator<br>✈ oracle property of the linear parametric estimator |
| Orthogonal | LMS | – Outliers are at random | ✈ LS ortho_LS LMS ortho_LMS are compared |
| Clipped | LASSO | Lasso<br>– Selects too many noisy variables. | ✈ Moderatly clipped (MC) LASSO<br>✈ Deletes noisy variables successively without sacrificing prediction accuracy much |
| Weighted | WLAD | – Asymptotic normality<br>– Stationarity<br>– Non-stationary | ✈ Wlad<br>✈ Arfima |
| Weighted | WLAD | – Heavy-tailed errors<br>– Outliers in x | ✈ Adaptive least absolute shrinkage and selection operator (LASSO)<br>✈ Simultaneous robust parameter estimation and variable selection in regression |

Table 4-1c: Recent advances and applications of LASSO, LMS

| | | |
|---|---|---|
| LMS | – Outliers with respect to the set of independent variables | ✈ \{SYSTAT\} |
| LMS | – non-Gaussian | ✈ Fractionally integrated autoregressive moving average |
| LMS | – Outliers<br>– highly skewed or heavy tailed distributions | ✈ Robust fuzzy linear regression model based on the Least Median Squares |
| LTS | – Multicollinearity<br>– Outlier<br>– HeterosedasticNoise | |
| LMS | | ✈ Outlier-free major region of the shape is extracted |

| | | |
|---|---|---|
| LMS | | Probabilistic algorithms for LMS |
| LMS | – Left-truncated<br>– Right-censored | |
| LMS | | A Microsoft Excel workbook developed<br>bivariateRegression through the origin |
| LARE | | Least absolute relative error (LARE) |
| LASSO | | Hydrophilic interaction liquid chromatography (HILIC)<br>QSRR<br>Nucleosides |
| LASSO | – Multiple-regimes<br>– Unknown number of thresholds | Threshold autoregressive models (TAR)<br>Consistent location of the thresholds |
| LASSO | | Fault isolation → quadratic programming problem with a sparsity constraint → solved with LASSO |

## 05. Polynomial regression

Polynomial models are invoked if the magnitudes of residuals in y for a linear model are far greater than the accuracy of the measurement/ reproducibility of the data and/or exhibit a trend at least for four to five successive points. A distinct case is when the scatter diagram of x vs y shows a parabolic trend or even residuals are not random. Such non-linear trends are common in univariate/multi-component calibration, variation of chemical parameters with dielectric constant, ionic strength or temperature.The procedure of moving towards cubic and quartic terms along with binary and higher order cross product terms is continued as per the need and prior literature reports for similar tasks.A full quadratic model is sought after in experimental design in many fields of research under the name 'Response surface methodology' and was discussed extensively in our earlier reviews [17-44 and references therein]. The advantage with RSM in full factorial, central composite designs is that design matrix is orthogonal and ordinary (unit weighted) multiple least squares method is adequate. In all other cases, values of x, $x^2$, $x^3$ etc. are correlated and thus MLR becomes unstable with number of terms of design matrix. However, its prevalence in yester years in applied sciences was with an implicit plea that it is used for finding trend (curve fitting) with minimum residuals and not as a technique for arriving at parameters of physico-bio-chemical parameters.In mathematical literature the more appropriate methods available are orthogonal-/ collocation/ rational polynomials with of course a few constraints on x-scale.The data input and least squares procedure are same as that for LLS except for the developed design matrix.

Method flow of polyLS2015: This program uses design matrix function to calculate linear, quadratic, cubic, quartic vectors and also binary and ternary product terms for all x variables.Different models are generated with polyModels program. For each model, exploratory analysis including angle and correlation between all pairs of vectors and singular values of X matrix are inspected. The regression coefficient, their statics and ordinary residuals are outputted in tabular form and in graphic mode. The flow of m-functions is in chart 5-1.

**Chart 5-1: Program flow of polynomial regression**

```
```````````````````````````````````````````````````````````````````````````````
 Method flow polyLS2015m file
```
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Cal linear, quadratic,cubic,quartic, binary/ternary cross products desmat2015
Generate models with different combinationsPolyModels

Repeat foreach model in the set
 %
 CalCorrelation coefficient, angle between vectorsccangsvd
SVD analysis for each term in model
 cal parameters, ycal and ordinary residualsFormulas_LS
 cal standard deviation, t balues of regression coefficientsregcoefstat

 Storing output in object modeoo_polyLS
 Graphic output
End repeat
 %
 Summary table
--------------------------------------------------------------------------------
```

Design matrices for polynomials of second to fourth order: The m-functions for design matrix and models up to fourth degree polynomial are described inMatLabProg. 5-1.

```
MatLabProg. 5-1:
```

```
%
%polyLS2015.m (R S Rao) 4/13/93, 10/27/1997,10/21/2011
%

%
%Flow of Method base
%
StepByStep_polyLS2015


%%Terms in Polynomial model

[Models_] = polyModels ;
% Design matrix
[one,lin,quad,cube,quartic] = desmat2015(x);
%

%%
 [Nmodels,columns] = size(Models_);
M1 = 1; M2 = Nmodels;
%% LLop for select polynomial models

for n = M1:M2
%
clear X

z = Models_{n,:}; dispst(['^^^^^^^^^^^^^^^^^^^^ Model : ', z])
X = [one eval(z)];
[X,y]
%
%%
% Reg parameters, ord_residuals, par statitics
%
lspar2015
%
 %%
 oo_regpar
%gr_polLS99
```

DataSet 5.1: The response (y) data for a third order polynomial (y = **1** + fn(x),a: [1,1,0.2,0.8]) is simulated in the x range of -1 to +1 with 0.2 increments.The Gaussian noise generated with zero mean and 0.05 standard deviation is added for the eleven y points.

Results and knowledge bits: The contributions of individual components to fn(x) in numerical and graphical form follows (output 5-1).No noise is added to the data.

Output 5-1: DataSet 5-1	

Simulated data of third order polynomial	f1 x; f2 0.2 * x.^2; f3 .8*x.^3; f4 ones(rx,1);
	
	fn = f1+f2+f3+f4;
	Order : 3; NP : 11

```
  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      x,      y       noiseRN,    fn,      f1,       f2,      f3,       f4
 -------------------------------------------------------------------------
      -1     -0.6      0-0.6       -1      0.2      -0.8       1
     -0.8    -0.0816       0    -0.0816     -0.8     0.128    -0.4096       1
     -0.6     0.2992       0     0.2992     -0.6     0.072    -0.1728       1
     -0.4     0.5808       0     0.5808     -0.4     0.032    -0.0512       1
     -0.2     0.8016       0     0.8016     -0.2     0.008    -0.0064       1
      0       1       01       0       0       0       1
     0.2     1.2144      0     1.2144      0.2     0.008    0.0064       1
     0.4     1.4832      0     1.4832      0.4     0.032    0.0512       1
     0.6     1.8448      0     1.8448      0.6     0.072    0.1728       1
     0.8     2.3376      0     2.3376      0.8     0.128    0.4096       1
      1       3       03       1      0.2      0.8       1
```

```
[Desired meanYnoise, stdYnoise]

      0      0.00
mean(nr),std(nr)obtained for NP: 11

  0.0    0.0
```

Fig 5-1shows individual contributions of fn, x, $0.2*x^2$, $0.8*x^3$. Fig.5-2(b) depicts function, noise and that with noise. Fig. 5-2 pictures the same information but on a single scale (-1 to +1). It enables visual picture of individual trends and on relative scale.

Fig. 5-1(a): Linear, quadratic and cubic Components of fn

a) f1 :x;

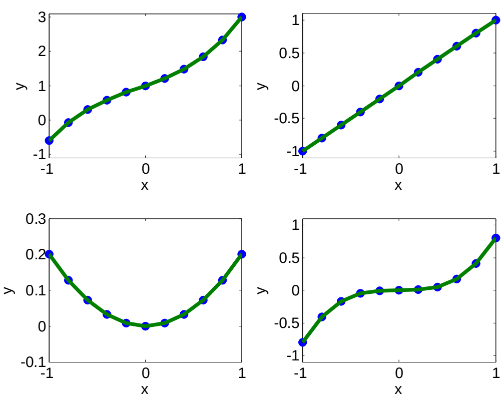
b) f2: 0.2 * x.^2;

c) f3: .8*x.^3;

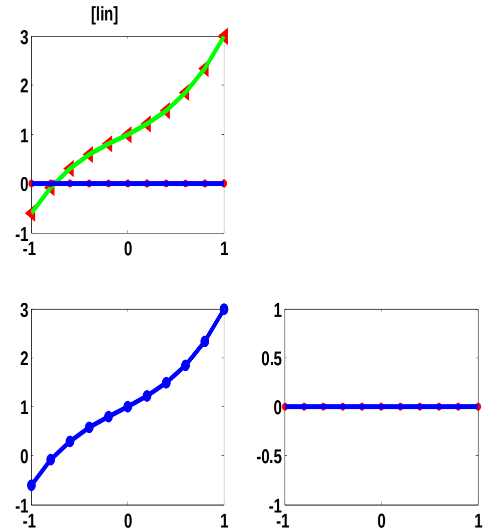
d) f4: ones(rx,1);

e) fn: f1+f2+f3+f4;

(a) (b)



(c) (d)



(b)(c)

Noise : 0

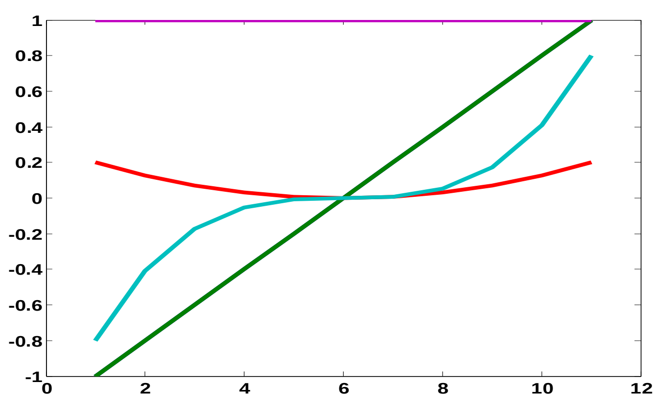


Fig. 5.2

[Desired meanYnoise, stdYnoise]

0 0.05

mean(nr),std(nr)obtained for NP: 11

-0.012372 0.032935

Analysis with third order model

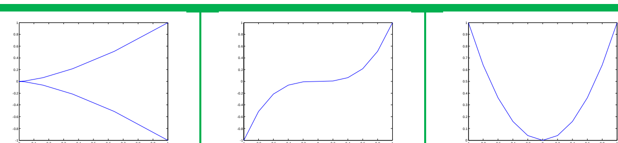
Correlation coefficient, angles and singular values of design matrix: The pairs of vectors (x2,x3; x3 x4) are highly linearly correlated.From the profiles (Fig.5-3), it bears no meaning.The vector angles are also low for pairs (c1 c3;), but orthogonal for the pairs (c1 c2; c3 c2; c3 c4;c4 c1).The singular values and percent explainability show all the four functions significantly contribute to y response.

Table 5-1: Correlation matrix, angles between vectors, singular values and their explainability

Correlation matrix	Angles between column vectors	Singular values	
~~	---------------------------------------		
x1x2x3x4y1	c1c2c3c4c5	s	% expl
------------------------------------	---------------------------------------	----------------------	
x1NaN	c10.00	3.6012	46.234
x2NaN1.00	c290.000.00	2.6017	33.403
x3NaN0.001.00	c341.4590.000.00	1.0791	13.854
x4NaN0.920.001.00	c490.0022.8390.000.00	0.5069	6.5086
y1NaN0.990.070.971.00	c542.9947.7554.5548.730.00		
	c1c2c3c4c5		

Explanation: First column of correlation matrix is NaN, since the first column of X is colum vector of ones

>>one =ones(6,1) one = 1 1 1 1 1 1	>>corrcoef([one]) ans = NaN >>corrcoef([one one]) ans = NaN NaN NaN NaN	>>x = [1:6]' x = 1 2 3 4 5 6	>> corrcoef([x]) ans = 1 >> corrcoef([x x]) ans = 1 1 1 1	>>corrcoef([one x]) ans = NaN NaN NaN 1 >>corrcoef([one x x one]) ans = NaN NaN NaN NaN NaN 1 1 NaN NaN 1 1 NaN NaN NaN NaN NaN



x: quad ; y: cube x: lin ; y: cube x: lin ; y: quad

Fig. 5-3: Scatter profiles of pair wise component

Regression coefficients and statistics for third order polynomial: The estimated regression coefficient from least squares analysis coincides with the values with which the function is generated. The standard deviations in par are of the order 10^{-30}, as it is simulated data without noise. Standardized regression coefficients and t-values bear no relevance as no stochastic component is present in response.

Table 5-2: Model No: 7

y = Fn{[lin quad cube]}

[zpar_poly{n,:} zsda{n,:} zstanda{n,:} zta{n,:}]

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Parsda        standa
`````````````````````````````````````````````
      1   5.151e-31   4.8437e-16
11.3894e-30 1.3066e-15
0.2 9.6521e-31 1.8153e-16
0.81.7986e-30 1.353e-15
~~~~~~
```

Chart 5-2:

ModelPoly =
'[lin]'
'[quad] '
'[cube] '
'[lin quad] '
'[lin cube]'
'[quad cube]'
'[lin quad cube]'
'[quartic]'
~~~~~~

This is an analysis of choice for curve fitting and not parametrization design. Another situation is when the SDs of regression coefficients are high and sdy is low which serves in curve fitting and interpolation.

Statistical analysis of choice of best set of models: The program automatically tests eight models and outputs regression parameters and statistics.From a perusal of sdy, it is obvious, quad, quartic models are ruled out based on sdy (>1) as the y-data in the range of -1 to +1. Cubic as well asquad and qubic models are the next set to be investigated. The sdys are similar. Finally the two models viz. (i) linear & cubic (ii)lin quad & cubic are prospecting. At a glance model 7 gives an impression of over ambitious from the sdy magnitude of 1e-16. The further analysis will help to choose the correct model.

Table 5-2(b): Model No: 7							
Sdy							
1	2	3	4	5	6	7	8
[lin]	[quad]	[cube]	[lin quad]	[lin cube]	[quad cube]	[lin quad cube]	[quatric]
0.17548	1.0532	0.26784	0.15907	0.074103	0.25739	8.7024e-16	1.0534
Valid if Sdy in y > 0.2	invalid	invalid	Valid if Sdy in y > 0.2	Acceptable	invalid	Overambitious	invalid

Sd in parameters: quad model is invalid as sda >500%. In the quatric model, sda of quatric term rules out its validity (sda>500%) and mean term also has sda >50% ruling out the model.All other models appear reasonable and cannot be rejected based on sda and obviously on t-values.

Table 5-2(c): Model No: 7							
Sda							
[lin]	[quad]	[cube]	[lin quad]	[lin cube]	[quad cube]	[lin quad cube]	[quatric]
OK	Invalid	OK	OK	OK	OK	OK	invalid

Table 5-2(d): Model No: 7				
#		**Parsd_par Standardized par**	**Parameterof**	
1	[lin]	1.080.010317 0.060234 1.5696 0.0163120.13841	Linearinflated	Quad, cube missing
2	[quad]	10.56140.50567 0.21.05200.18951	quad exact	Lin and cube missing
3	[cube]	1.08 0.024033 0.091936 1.993 0.0491890.34724	Cubic inflated	Lin quad missing
4	[lin quad]	1 0.014406 0.08100 1 0.014406 0.08100 0.2 0.026995 0.030358	Linear inflated Quad exact	cube missing
5	[lin cube]	1.080.0020696 0.026978 10.00843320.10179 0.80.0109160.10541	X exact cube exact	Quad missing

6	[quad cube]	10.0377180.13107 0.20.070677 0.049121 1.9930.0511020.35393	Quad exact Cube inflated	Linear missing
7	[lin quad cube]	1 5.151 e-31 4.8437e-16 1 1.3894e-30 1.3066e-15 0.2 9.6521e-31 1.8153e-16 0.8 1.7986e-30 1.353e-15	All exact	All terms upto cubic
8	[Quatric]	1.02710.470770.43545 0.185741.01410.16963	High standard deviation	Model invalid

Pointwise residuals: Ordinary residuals speak of the model if the precision and accuracy of y response is known apriori. Otherwise local heuristics based on discipline govern. A bird's eye view of residuals (numeric magnitude (table 5-2e, graphs to detect trends Fig. 5-4) yield the information bits vide infra.

The magnitudes of residuals of models 1 and 3 infer that they are acceptable if data reproducibility is greater than 0.15. The trends in plot of residual vs x values, all models show a significant trend indicating inadequacy of models. Looking into magnitudes on plots indicate model 5 miss contribution to an extent of 0. 2 and similarly model 6. Models 1 and 8 miss a larger contribution of around 1.0.

It can be reconciled as model 1 does not contain most significant quadratic and cubic terms, while quartric model does contain any of the terns of simulated data. Another point to be noted is the range of x is -1 to +1 and hence the relative magnitudes of quadratic and cubic terns are less than linear one. Further, the coefficients 0.2 and 0.8 have diminishing effect. A perusal of data shows their relative significance.

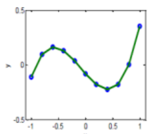
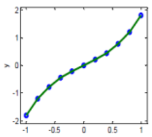
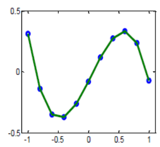
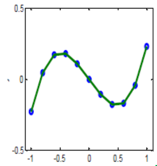
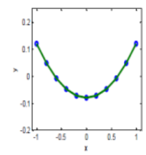
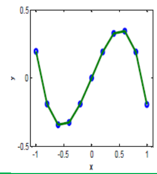
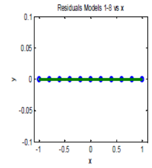
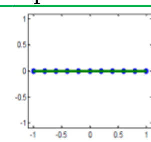
Table 5-2(e): Model No: 7

```
Columns 1 through 9
      1     -1      0-0.1104    -1.8    0.31303-0.2304      0.12    0.19303
      2     -0.8      0    0.09408   -1.2096    -0.14117    0.04608     0.048    -0.18917
      3     -0.6      0    0.16096   -0.7728    -0.35031    0.16896     -0.008    -0.34231
      4     -0.4      0    0.12864   -0.4512    -0.37165    0.17664     -0.048    -0.32365
      5     -0.2      0    0.03552   -0.2064    -0.26246    0.10752     -0.072    -0.19046
      6      0      0-0.08      0      -0.08       0     -0.080
      7      0.2     0   -0.17952     0.2064    0.11846   -0.10752     -0.072    0.19046
      8      0.4     0   -0.22464     0.4512    0.27565   -0.17664     -0.048    0.32365
      9      0.6     0   -0.17696     0.7728    0.33431   -0.16896     -0.008    0.34231
      10      0.8     0   0.00192     1.2096    0.23717   -0.04608     0.048    0.18917
      111      0   0.3504      1.8   -0.0730290.2304      0.12    -0.19303
              x      nr      1            2      3            4       5            6

 Columns 10 through 12
  6.6613e-16   -1.8128     -0.6
  1.2212e-15   -1.1848   -0.0816
  1.2768e-15   -0.75197    0.2992
  9.992e-16   -0.45106    0.5808
  6.6613e-16   -0.2258    0.8016
  2.2204e-16  -0.0271031
 -2.2204e-16    0.187    1.2144
 -8.8818e-16   0.45134    1.4832
 -1.1102e-15   0.79363    1.8448
 -8.8818e-16    1.2344    2.3376
0    1.7872      3

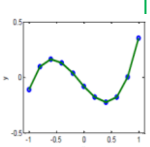
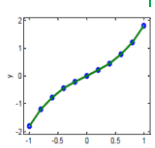
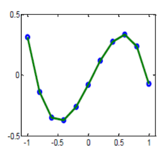
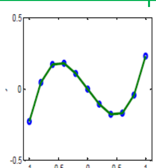
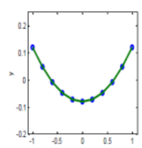
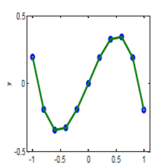
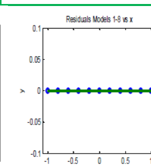
   7          8             y
```

Figure 5-4: Plots of residual for exhaustive model set[Models No: 1 to 8]

Residuals							
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
[lin]	**[quad]**	**[cube]**	**[lin quad]**	**[lin cube]**	**[quad cube]**	**[lin quad cube]**	**[quatric]**
							
Cubic	Linear& cubic	Quadratic + ??	Cubic	Quadratic	cubic	Adequate or overambitious	
			Input noise				
							

Consolidation of model results: Table 5-3 incorporate picking up inadequate, adequate and overambitious models from the exhaustive model set.

Table 5-3: Knowledge bits for picking up best model

	Residuals							
	1	2	3	4	5	6	7	8
	[lin]	[quad]	[cube]	[lin quad]	[lin cube]	[quad cube]	[lin quad cube]	[quatric]
Resid								
Sdy	Valid if Sdy in y > 0.2	invalid	invalid	Valid if Sdy in y > 0.2	Acceptable	invalid	Overambitious	invalid
sda	OK	Invalid	OK	OK	OK	OK	OK	invalid

→

	Residuals							
	1		3	4	5	6	7	
	[lin]		[cube]	[lin quad]	[lin cube]	[quad cube]	[lin quad cube]	
Resid	Cubic		Quadratic + ??	Cubic	Quadratic	cubic	Adequate or overambitious	
Sdy	Valid if Sdy in y > 0.2		invalid	Valid if Sdy in y > 0.2	Acceptable	invalid	Overambitious	
sda	OK		OK	OK	OK	OK	OK	

→

	Residuals							
	1			4	5		7	
	[lin]			[lin quad]	[lin cube]		[lin quad cube]	
Resid	Cubic			Cubic	Quadratic		Adequate or overambitious	
Sdy	Valid if Sdy in y > 0.2			Valid if Sdy in y > 0.2	Acceptable		Overambitious	
sda	OK			OK	OK		OK	

→

	Since there is no noise →			→since simulated data with zero noise even 0.07 in sdy is intolerable. So,			
Model No →	**5**	**7**		**5**	**7**		**7**
Statistic	[lin cube]	[lin quad cube]		[lin cube]	[lin quad cube]		[lin quad cube]
Resid	Quadratic	Adequate or overambitious		Quadratic	Adequate or overambitious		Adequate or overambitious
Sdy	Acceptable	Overambitious		Acceptable	Overambitious		Overambitious
sda	OK	OK		OK	OK		OK

The acceptable one is model 7. It appears to be overambitious from analysis of noisy data.But it is fact of modeling that it is the acceptable model. The reproduction of regression coefficients and degree of polynomial with zero residuals is a worthy knowledge bit.

6. Multi linear LEAST SQUARES (MLR)

The variation of a response vector on more than one explanatory variable is modelled as

$y=a_0 + a_1 * x_1 + a_2 * x_2 + \ldots + a_j * xj + \varepsilon$

Here, the model is linear in variables as well as in parameters which are estimated by a regression procedure. The necessary conditions and consequences are same as for linear least squares with one explanatory variable.The additional constraint is that the two variables (x_1 andx_2) are orthogonal to each other.The holds in the experimental design task with factorial designs. In all other instances, the x variables are to be chosen such that their source is not only independent but also their numerical magnitudes are not statistically linearly correlated.Here, the number of variables is restricted to two in simulated sets while larger number of xs is considered for real life datasets.

Linear model with two explanatory variables: The variation of rate constant or equilibrium constant of reactions of homologous organic moieties with macroscopic properties of the compounds (substituent effect, steric factor) is explained by linear model with two explainable parameters

$y = a_0 + a_1 * x_1 + a_2 * x_2 + \varepsilon$

Except additional tests for the relationship between x1 and x2, the functions developed for lls2015m are used in parameter estimation, residual analysis and regression coefficient statistics. Chart 6-1 incorporates the data structure,additional necessary conditions for multiple linear regression.

Chart 6-:1 Data structure		
$x = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ x_{31} & x_{32} \end{bmatrix};$	$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix};$	$Design\, matrix : X = [one\ x];$ $par \quad : \ [a_0\ a_1\, a_2]^T;$
	x : Explanatory/ independent variable	

MODEL	
Matrix form	**Algebraic notation**
$Model:\ y + normal\_noise = X * par$	**Noise** : iidnoise of normal distribution $\varepsilon_y=[\varepsilon_1\varepsilon_2\varepsilon_3......\varepsilon_{NP}]^T$
$y + normal\_noise = \begin{bmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ 1 & x_{31} & x_{32} \end{bmatrix} * \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$	$Model:\ yi + normal\_noise = a_0 + a_1 * x_{i,1} + a_1 * x_{i,2}$

Chart 6-1b: Multiple linear regression (MLR)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Necessary Conditions (NC)	(single) Object Fn of MLR	
~~ 'Indepedent i.e. ang(x1,x2)= 90°' ' cc(x1,x2)= 0' ' + NCs of LLS'	~~~~~~~~~~~~~~~~~~~~~~~~~ objFn: 'Sum of squares of residuals' Goal: 'Min(objFn)' SolutionMethod: 'Unit weighted Least Squares'	$objFn = residy^T * residy$ $Goal: \min(objFn)$ $par = \left(X^T * X\right)^{-1} * X^T * y$

```
---------------------------------------------------------
Failure conditions        Remedial Measure
---------------------------------------------------------
'x1 and x2 correlated'   'Ridge Regression ;PCR'
'Mixture constraints'    'PLSR'
---------------------------------------------------------
```

Chart 6-1c: ```````````````````````````````````` MLR2015.m MethodBase m file ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ Calculation of % > Cal polynomial, crossproducts vectors desmat2015 > Models developement mlrModels	% % mlr2015.m (R S Rao) 8/10/92 9-11-15 % function [LLS_stats] = mlr2015(X,x,y,prin) % % if nargin < 3,
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

```
  For each model
> X matrix
>  LLS procedure         lls2015
  endFor

>> output: Tabular summary
```

```
     data_x1x2y_01
end
diary off
!del mlr2015.txt
diary mlr2015.txt
%
if nargin <3 | nargin ==3
    prin = 0;
end
%
stats_LLS = [];
```

```
%
%    Estimation of Slope,inter, ycal
%
StepByStep_MLR2015

anova2015(X,x,y);
[a_LLS,ycal,res2] = Formulas_LS(X,x,y);
[sda_LLS,ta_LLS,standa_LLS] = regcoefstat(X,x,y);
[ycal,residy,sdy] = ordResid(X,x,y,a_LLS);
```

```
diary off
edit mlr2015.txt
```

```
Analysis of Variance for regression
`~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
          sum of    degrees of   mean F_Reg
          squares   freedom    squares
`~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Model      52.2337    2     26.1169    198303.6776
Residualy   0.0001317 1    0.0001317
Totaly     52.2338    4     17.4113
`--------------------------------------------------
x =
    1.983e+05
Information: RegMod  is acceptable at 0.05
significant level
              since,Fcal:198303.6776 > F_table_value
= 4999.5(with df.Residy=1,df.Model 2):
Inference_ANOVA :Chance  occurance of RegMod <
0.05(or <5%) probability
              KB: F is scale independent
```

```
np: 4
                 npar: 3
             ss_Model: 52.234
            ss_Residy: 0.0001317
              ss_Toty: 52.234
             df_Model: 2
            df_Residy: 1
              df_Toty: 4
          df_TotyCorr: 3
          Meanss_Model: 26.117
         Meanss_Residy: 0.0001317
           Meanss_Toty: 17.411
            F_RegModel: 1.983e+05
    probFvalueRegModel: 5.0427e-
06
             R_squared: 1
      R_squared_adjsted: 0.99999
            replicates: 'No'
                   LOF: ''
                    PE: ''
```

```
||||||||||||||||  statistics of regress parameters
     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
----------------------------------------------------------------------------
         a,          sda,         standErra,       standa,      ta   ttable prob
----------------------------------------------------------------------------
      0.99986    0.0057381          0.5       0.49993       174.25     0.0036534
       1.9961    0.0057381          0.5       0.99806       347.88     0.00183
       3.0123    0.0057381          0.5       1.5061       524.97     0.0012127
````````````````````````````````````````````````````````````````````````````
```

```
\\\\\\\\\\\\\\ t- statistics of regress parameters
     ~~~~~~
....
```
     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

```
          a,           ta    t-crit      ta>tcrit       alpha
................................................................................
 0.99986      174.25       63.657            1          0.05
       1.9961        347.88       63.657            1          0.05
       3.0123        524.97       63.657            1          0.05
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

Resid = vary: 0.0001317 sdy: 0.011476 scaleEstimate: 0.011476 R_squared: 1 R_squared_adjsted: 0.99999	varCovResid: 3.2925e-05 0 0 0 3.2925e-05 0 0 0 3.2925e-05

	MatLabProg 6-1 % % mlrLS2015.m (R S Rao) 8/10/92 9-11-15 % % StepByStep_MLR2015 %%Terms MLR model [Models_] = mlrModels ; % Design matrix [one,lin,quad,cube,quartic,cpb,cpt] = desmat2015(x); % %% [Nmodels,columns] = size(Models_); M1 = 1; M2 = Nmodels;
For each model > X matrix >LLS procedure lls2015 endFor	%% Loop for select MLR models for n = M1:M2 % clear X z = Models_{n,:}; st = 'Cal Design matrix developement'; center02(st); dispst(['^^^^^^^^^^^^^^^^^^^^ Model__ : ', z]) X = [one eval(z)]; [X,y] % %%
MLR2015: InpCheck_LS; Formulas_LS	% Reg parameters, ord_residuals, par statitics % st = 'Cal of regression coe, sda, resid, sdy..'; dispst(st); lspar2015

ordResid
 	
Resid Analysis
 	
Autotest_mlr2015
	`%`
	`%%`
	`st = 'object form of regression coe, sda, resid, sdy..';`
	`dispst(st);`
	`oo_regpar`
	`end`
	`%%`
	`%`

`>> output: Tabular summary`
`-----------------------------------`
	`st= ' Display of summary of models';`
	`dispst(st)`
	`disp_regpar`
	`%%`

6.1 Orthogonal (x1 and x2) & No noise in y

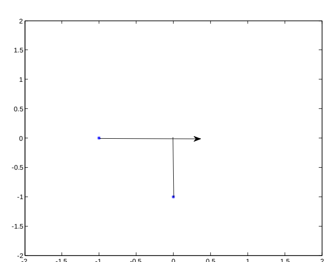
Dataset.sim 6.1:The response data (y) is simulated from bilinear function (x1 + 2*x2) with orthogonal x1 and x2 column vectors (chart 6-2).No Gaussian noise is added and it is devoid of outliers or even minor processes. This dataset is designed to demonstrate the steps in MLR and can be implemented even with simple memory or at best with paper and pencil.

Chart 6-2: Data.Sim 6.1

$$x1 = \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix}; \qquad x2 = \begin{bmatrix} +1 \\ +1 \\ -1 \\ -1 \end{bmatrix}$$

Correlation coefficient, angles and singular values of design matrix: The angle between columns vectors (x1 and x2) or row vectors ($x1^T$, $x2^T$) are 90^o. The linear correlation coefficient of x1 and x2 is 0. Thus, the dataset adheres to necessary conditions of MLR (output 6-1). The magnitude of singular values show equal contribution of the two vectors each to an extent of 50%. The columns of v matrix are represented in the figure 6-1. The response is correlated with explanatory factors to an extent of 0.45and 0.89.

Output 6-1: Dataset.sim6.1

Ysimul = x1 + 2*x2	x1,x2 ; NP : 4		
~~ x1x2ysimul ----------------- 1 1 3 -1 1 1 1-1-1 -1-1-3	Angles between x1 and x2 x1x2 --------------------- x10.00 x290.000.00 x1x2 -----------------------	Angles between Row vectors r1r2 --------------------- r10.00 r290.000.00 r1r2 ---------------------	correlation matrix x1x2y1 --------------------------- **x11.00** **x20.001.00** y10.450.891.00
	s Var totVar x1 3 50% 50% x2 3 50% 100%	V = [0 -1 -1 0]	 Fig. 6-1: V-matarix representation

Regression coefficients and statistics: The estimated regression coefficients from MLR $[0\ 1\ 2]^T$ are exactly equal to the values used in the simulation. The standard deviations in parameters are zero.

Output 6-1b

	Par	sda	sta	ta
0	0	NaN	0	0
1	0	NaN	0	0
2	0	NaN	0	0

Residuals in y: The y_{cal} values reproduce the simulated ones with zero residuals. The model is adequate. As it is a simulated one without noise, obviously no further statistical analysis.

Output 6-1c

```
X1x2yycal residy
1 1 3 3 0
-1 1 1 1 0
 1-1-1-1 0
-1-1-3-3 0

Sdy0
```

Expert' Inferences

Ground truth; Data is noise free
No need of statistical analysis

- ✓ **Cc (x1,x2) =0; angle(x1,x2) =90**
- ✓ **MLR is apt**
- ✓ **ResidY zero,**
- ✓ **Sda in a is zero**

Model acceptable

- **Robust method Redundant**
  - o **As it outputs same result**
- Orthogonalisation techniques for x is not necessary
- No need of stochastic approach

Deterministic mathematical model

- Data is**noise free/of extremely small noise**

6.2 Non-orthogonal variables-- Failure of MLR

If x matrix is significantly correlated (or nearly singular) the regression coefficients are of wrong sign/ with high standard deviation. The presence of outliers in y, or x or both attract the regression plane towards outliers and thus not reliable. The typical failure conditions and remedial measures are described in chart 6-3

Chart 6-3:**Failure Conditions FC**	**Remedial Measures RM**
'heterosedastic noise'	**'WMLR'**
'Outliers in y '	**'LMS'**
x1 and x2 correlated	Ridge Regression PCR
Mixture constraints	PLSR

Outlier in y → (wrong regression coefficients)

Dataset.simul.6.2: The dataset with 8 data points includes one y outlier marked in red. To the simulated function values, normal noise of 0.0 is added.

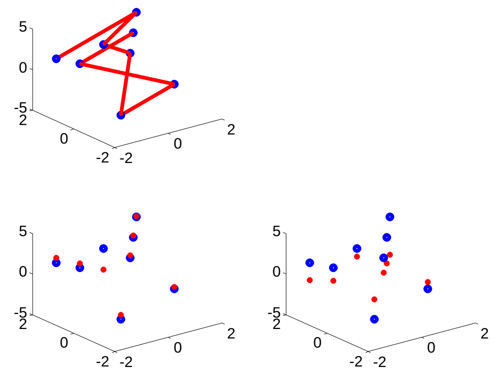
Output 6-2: Dataset.sim 6.2

One outlier in y `

```
````````````````````````````````````````````````````
````
     x1      x2       y      ycal     resid        nor
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~
```

```
      1       1       3     3.0701  -0.070139      0
     -1       1       1     1.1824   -0.18236      0
      1      -1      -1   -0.92986  -0.070139      0
     -1      -1      -3    -2.8176   -0.18236      0
0.5     0.51.5     1.5982   -0.098194       0
-0.5     0.5     1.51    0.65431    0.85569       0
1.5     1.54.5     4.5421   -0.042083       0
     -1.5     1.51.5     1.7104    -0.21042      0
```

sdy =
0.37953



```
parsdastandaexpected (par)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
0.12625 0.069294 0.0210420
0.94389 0.0576170.130811
20.0653310.314282
```

Dataset.simul.6.3:This is also a data set of eight points but with two outliers (output 6-3).

Output 6-3: Dataset.sim 6-3

Two outliers in y

```
````````````````````````````````````````````````````````````````````
     x1   x2   y   ycal   resid   nor
  ~~~~~~
       1   1   3   2.9399   0.060119   0
      -1   1   1   1.2765   -0.27655   high 0
1   -1   -2.01-1.493   -0.51702 Large   0
      -1   -1   -3   -3.1563   0.15631   high0
0.5   0.5   1.5   1.4158   0.084167   0
-0.5   0.5   1.51   0.58417   0.92583 Large   0
 1.5   1.5   4.5   4.4639   0.036071   0
-1.5   1.5   1.5   1.9689   -0.46893   high   0
```

sdy = 0.49283 Not acceptable Expected :0.0

```
parsdastanda expected (par)

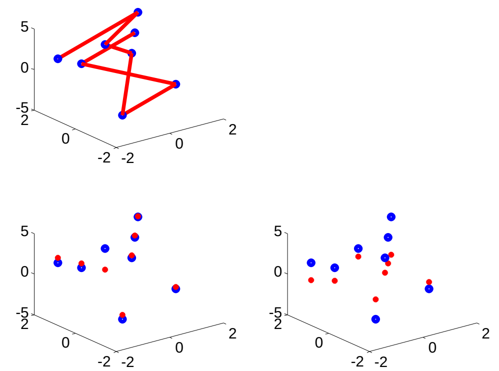
-0.10821 decreased  0.11684 ok -0.023421         0
0.83167 decreased  0.097152 ok  0.14966         1
 2.2164increased  0.11016  ok 0.45226         2
```

Parameters acceptable from Sda values

Two very high residuals in y … suspicious for trend

Sdy is very large
Model fails

Remedy : Remove two outliers rerun
Robust method (LMS) to know trend without removing outliers in data
  ~~~~~~



Output 6-3b:

Two outliers in y

a_LMS =
0
1
2

a_LLS
0.28571
1.1765
2.4286
ans =
0
0
0
0
-2
-1

✓ LMS **Parameters are correct**

Residuals detected outliers (points: 5 and 6)

Rerun deleting outliers

[X]

one x1x2y

1 1 1 3
1-1 1 1
1 1-1-1
1-1-1-3

a_LLS =	a_LMS =
0 1 2	0 1 2
Resid_LLS =	**Resid_LLS**
0 0 0 0	0 0 0 0

Correlated & non-orthogonal (x1 and x2) & No noise in y

Dataset 6.3: This is a 11 point dataset with x1 and x2 vectors of explanatory variables correlated and angle between them is zero (output 6-4).

Output 6-4: **Dataset 7.3** x1,x2 ; NP : 11

x1	x2		yycal	resid
1	1	2	5.7578	-3.7578
2	2	4	11.516	-7.5156
3	3	6	17.273	-11.273
4	4	8	23.031	-15.031
5	5	10	28.789	-18.789
6	6	12	34.547	-22.547
7	7	14	40.305	-26.305
8	8	16	46.063	-30.063
9	9	18	51.82	-33.82
10	10	20	57.578	-37.578

```
11      11       22    63.336    -41.336
sdy =
    28.177
```

```
-----------------------------------------------------------------
      Par       sda           sta
` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` ` `
 -1.7764e-15    577.58   -3.433e-14
2.9414   3.7462e+09   3.6871e+08
2.8164   3.7462e+09   3.5304e+08
```

```
correlation matrix
x1x2y1
--------------------------
x11.00
x21.001.00
y11.001.001.00
```

- X1 and x2 highly correlated
- Inverse should not be used, if so large variance
- Parameters not acceptable ; sda suggests failure of inverse of X'X
- Residy are extremely large

→ Model fails

Remedy: PCA for X; regression of PCs with y

```
Angles between x1 and x2
x1x2
--------------------
x10.00
x20.000.00
x1x2
----------------------

Angles between Row vectors
r1r2
--------------------
r10.00
r290.000.00
r1r2
--------------------
```

Dataset 6.4: In this dataset, x1 and x2 correlated (0.99) andnon-orthgonal (angx = 11.3). It is one instance of failure of the model (output 6-5).

output 6-5: Dataset 7.4

x1,x2 ; NP : 4

```
Angles between x1 and
x2
x1x2
--------------------
x10.00
x2   11.380.00
x1x2
----------------------
```

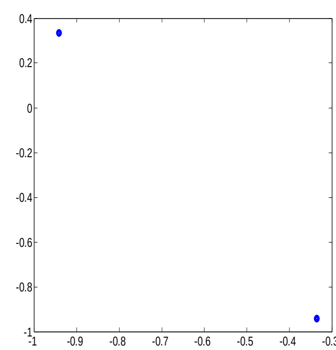
```
Angles between Row
vectors
r1r2
--------------------
r10.00
r29.74   0.00
r1r2
--------------------
```

```
correlation matrix
x1x2y1
--------------------------
x11.00
x20.991.00
y11.00 0.991.00
```

```
     s      Var  totVar
x1  23.869  94.0594.05
x2  1.510   5.949   99.999
```

Model fails
Remedy:PCA for X;
regression of PCs with y

```
V = [-0.94218    0.3351
-0.3351  -0.94218
```



6.3 Inadequate models: The response (y) of dataset 7.4 is simulated as ysimul = par1*x1 +par2*x2. But, it analysed neglecting x2 variable i.e. as ysimul3= par3*x1. The output is inoutput 6.6.

output 6-6:Dataset 7.4

x1,x2 ; NP : 4

```
Model analyzed as ysimul = par* x1;
   i.e. x2 is ignored
```

```
correlation matrix
```

<table>
<tr><td><pre>
``````````````````````````````````
x y ycalresidnor
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 1 3 2 1 0
 1 1 2-1 0
-1-1-2 1 0
-1-3-2-1 0

 sdy = 1.1547
</pre></td><td><pre>
x1y1
--------------------
x11.00
y10.891.00
</pre></td></tr>
<tr><td><pre>
-----------------------------
par sdaexpected (par)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
010
211.41421
</pre></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td><pre>
Model analyzed as ysimul = par*x2;
 i.e. x1 is ignored
</pre></td><td></td></tr>
<tr><td><pre>
``````````````````````````````````
x1 y ycalresidnor
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ans =
 1 3 1 2 0
-1 1-1 2 0
 1-1 1-2 0
-1-3-1-2 0
sdy =
 2.3094
</pre></td><td><pre>
correlation matrix
x1y1
----
x11.00
y10.451.00
</pre></td></tr>
<tr><td><pre>
par sda standaexpected (par)

0400
141.41421
sda =
 4
 4
standa =
0
 1.4142
</pre></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>

6.4 Realistic data: A four point data set with two orthogonal x1 and x2 and adding random normal noise. The statistics alomg with parameters are in output 6-7.

<table>
<tr><td><pre>
Output 6-7: Dataset 7.5
Ysimul = x1 + 2*x2 +noise_n(4,0.0,0.1)
</pre></td><td></td></tr>
<tr><td><pre>
Model analyzed as ysimul = par0+ par1*x1+
par*x2;
</pre></td><td></td></tr>
<tr><td><pre>
`````````````````````````````````````````````
x1,x2,y,nor,      ysimul
---------------------------------------------
      1      1    3.0875   0.087541      3
     -1      1    1.109    0.10902      1
      1     -1   -1.0218   -0.021825     -1
     -1     -1   -3.0348   -0.034768     -3

Mean(nor) = 0.0350
Std(nor)  = 0.0738
</pre></td><td><pre>
--------------------
correlation matrix
x1x2y1
--------------------
x11.00
x20.001.00
y10.440.901.00
------------------------
</pre></td></tr>
<tr><td><pre>
```````````````````````````````````````````
par      sda   standa   expected
(par)
````````````````````````````````````````````

 0.034991 0.00014808  0.00030108   0.0
 0.99787  0.00014808  0.0085863 1.0
 2.0633  0.00014808  0.017754  2.0
</pre></td><td></td></tr>
</table>
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

| | |
|---|---|
| ---------------------------------------------- | |
| | |

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

x1,x2,y,       ycal,    resid      noise added
------------------------------------------------------------
11 3.0875 3.0961 -0.0086046 0.087541
-111.1091.10040.00860460.10902
1-1-1.0218-1.03040.0086046-0.021825
-1-1-3.0348-3.0262 -0.0086046-0.034768
sdy =
 0.012169
```

| | | |
|---|---|---|
| Model: y = Fn{[x1 x2];par} + noise_n(mean,std) | | |
| Explanatory variable (x) | Response (y) | |
| | | |
| | | |
| [x1 x2] | y | |
| No noise | Noise [no, small, large] | |
| No outlier | [Outlier[no, few, many] | |
| | Subprocess [no, minor, major] | Non overlaping |
| | | Overlapping [partial, complete[magnitude [small, large] |
| Cc [0, 0.5,1] | | |
| Angle [90,45,0] | | |

## 7. Analysis of Variance (ANOVA) for regression

In applied sciences, based on number of influential factors considered, ANOVA is popular under different names like one way (ANOVA I), two way (ANOVA II) and multiway (MANOVA) types. The variation in response can sometimes be ignored by inspecting the numbers at a glance.When it is difficult to decide that the variation in y is just due to ignorable random (normal distribution) noise or is a result of model, a fool proof and unbiased approach for accreditation purpose is ANOVA, a sound statistical procedure.It separates variation in y into explainable factors and random effects.

KB for regression and parameter statistics: The first condition to be satisfied is number of data points is equal or more than number of regression parameters to obtain unique least squares solution (chart 7-1; MatLabProg 7-1). Otherwise, Simplex method in linear algebra is the choice. Even then uniqueness is sacrificed. The statistics for regression parameters and residual spread in y are calculable when NP > Npar. The kb_reg.m implements these heuristics rending a pure numerical algorithm into knowledge based one for proper choice of method, appropriate use of statistical procedures and avoiding software failurefor rare, but possible data sets.Similar add-ons of KBs at various levels of algorithm enhances power of software and also heart of fault-tracking, explanation of why it happened and why not that did not happen etc.

| Chart 7-1:KB of solution of regression equation | MatLabProg 7-1: |
|---|---|
| >>dem_kb_RegSoln<br>X =<br>1.00002.00003.0000<br>1.00001.41421.7321<br>ANOVA or Regression analysis not possible; No unique solution since np < npar | function dem_kb_RegSoln<br><br>v = [1 2 3];<br>X = [v; sqrt(v)],kb_RegSoln(X)<br>X = [1 2; 3 5],kb_RegSoln(X)<br>X = [v' v'.^2],kb_RegSoln(X) |

```
NP =2; Npar = 3
*************
X =
 1 2
 3 5
Deterministic task ; Solution of
simultaneous equation or Regression; Cal
ofstatistics (sdy, sda, t,.. not possible
since np == npar
NP =2; Npar = 2
*************
X =
 1 1
 2 4
 3 9
Over-determined task; Regression analysis
since np > npar
NP =3; Npar = 2
*************
XTX =
1436
3698
```

```
% KB_RegSoln.m18/3/1997 ; 9/11/15
%

function kb_RegSoln(X)
[np,npar] = size(X);

conseq{:,1} = 'ANOVA or Regression analysis
not possible; No unique solution';
conseq{:,2} = ['Deterministic task ;
Solution of simultaneous equation or
Regression'...
'Calculation ofstatistics (sdy, sda, t,..)
not possible'];
conseq{:,3} = 'Over-determined task;
Regression analysis';

 Ant{:,1} = 'np < npar';
 Ant{:,2} = 'np == npar ';
 Ant{:,3} = 'np > npar';

for i = 1:3

if eval(Ant{:,i})
disp(conseq{:,i})
disp(['since ',Ant{:,i}])
disp(['NP =',num2str(np), '; Npar = ',
num2str(npar)])
end
 end
```

ANOVA for regression model: The partitioning of sum of squares of response (y) into explainable regression, residuals and mean sum of squares is in table 7-1. If replicate measurements are available, residual sum of squares can further be decomposed into SS due to pure error (PE) and lack of fit (LOF). The Matlab program for ANOVA is given in MatLabProg 7-2.

```
Table 7-1: Decomposition of Total Sum of Squares of variance (SST)
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

SST (np)
                                   |
                                   |
                  |---------------------------------------|
                  |                                       |
                  |                                       |
            SS mean(1)                              SS corr (np-1)
                                                          |
                                                          |
                                           |-----------------------|
                                           |                       |
                                        SS Fact                 SS res
                                       (npar-1)              (np - npar)
                                                                 |
                                                                 |
                                                 |---------------------|
                                                 |                     |
                                              SS LOF                 SS PE
                                              (f -p)                (np - f)
```

$$\begin{aligned} &Total\ Sum\ of\ squares\ in\ y &&= Model\ SS &&+ \text{Re}sidual(in\ y)\ SS \\ &y^T * y &&= ycal^T * ycal &&+ (ycal - y)^T * (ycal - y) \end{aligned}$$

Table 7-1b: Formulae of ANOVA and Matlab code

| | | |
|---|---|---|
| **Source** | **Sum of squares** | `%%`<br>`% ------------- ANOVA FORMULAE --------`<br>`%` |
| | | `ybar = one' * y/(one' * one);`<br>`ymean= one * ybar;` |
| Mean Sum of Squares(SS): mean of sum of squares of response (y) | $\frac{\left[y^T * y\right]}{NP}$ | `ssmean = ymean' * ymean;`<br>`sscorr = (y-ymean)' * (y-ymean);` |
| Regression SS: SS explained by model minus meanSS | $\left[A^T * X^T * y - \frac{y^T * y}{NP}\right]$ | `ssfact = (ycal -ymean)' * (ycal-ymean);` |
| ResidualSS: SS of residuals in y | $\left[y^T * y - A^T * X^T * y\right]$ | `ssr= resid' * resid;` |
| | ---------------------- | |
| Total SS: SS ofy | $\left[y^T * y\right]$ | `sst= y' * y;` |
| Degrees of freedom: calculated from NP, npar and number of replicate measurements | df_Model = npar-1;<br>df_Residy = np-npar;<br>df_Toty = np;<br>df_TotyCorr = np-1; | `df = [np;1;np-1; npar-1;np-npar;];` |
| | | |
| Mean sum of squares | | |
| | $\frac{\text{RegSS}}{Npar - 1}$ | `ssz= [sst;ssmean; sscorr; ssfact; ssr];`<br>`mss = ssz./df;` |

| | | |
|---|---|---|
| | $\frac{\text{ResidSS}}{NP - Npar}$ | |
| | $\frac{\text{TSS}}{NP}$ | |
| F | $\frac{\frac{\text{RegSS}}{Npar - 1}}{\frac{\text{ResidSS}}{NP - Npar}}$ | `df1= npar;df2 = np-npar-1;` |
| | | `varExplained = ssfact/sscorr*100;` |
| | | anova =<br>sst: 364.2808<br>ssmean: 293.6949<br>sscorr: 70.5859<br>ssfact: 70.5836<br>ssr: 0.0023<br>varExplained: 99.9968<br>ybar: 6.9964<br>df1: 2<br>df2: 3 |

Table 7-2: R-square and corrected R-square

| Statistic | Formula | Matlab code | |
|---|---|---|---|
| | | | **KB.1:**<br>If $R^2 \rightarrow 1$<br>Then Larger variance in y explained by regression<br>If $R^2 \rightarrow 0$<br>Then X does not explain variation in y |
| Coefficient of determination (R-squared): It is equal to the proportion of variance in response (y) explained by independent variables (of model) in linear regression. It is also referred ordinary (or unadjusted) R-square | $R\_Sq = R^2 = \left[1 - \frac{SSModel}{SSTotal}\right]$<br>$Range: [0 < R^2 < 1]$ | `R_squared = 1-ssr/sst;` | – $R^2$ increases with increase in number of x variables<br>Remedy: $R^2_{adj}$ |

| | | | |
|---|---|---|---|
| | | | |

| | | | |
|---|---|---|---|
| Adjusted_R-square: R-Square is adjusted considering number of parameters | $R\_Sq\_adj = R^2_{adj}$ <br> $= 1 - \left[ \dfrac{\dfrac{SSModel}{(np - npar)}}{\dfrac{SSTotal}{(np-1)}} \right]$ <br> $= 1 - \dfrac{Meanss\_Residy}{Meanss\_Toty}$ | dft = np-1; <br> dfr = np-npar; <br> R_squared_adjsted = 1-(ssr/sst)*(dft/dfr); | + Models with different number of explanatory variables can be compared |
| | | | + |

Anova2015: The methodFlow of m-function of MatLab software of anova2015 (MatLabProg 7-2) is briefed in chart 7-2.

Chart 7-2:

```
``````````````````````````````````````````````````````````````````````````

 MethodFlow m file
~~~~~~

 Anova_regressionanova2015
 Model definitionpolyModels
 Development of design matrix (X)X2015
 Characterstics of X Xcond

 Knowledge bits Regression Feasibility kb_RegSoln
> Formulae for ANOVAFormulas_anova2015
 F testFtest

 if eXpert systemInference_anova
 if replicate measurements Inference_LOF
 if novice inform_anova
 if intelligent system Advice_anova
 %
>> output: Tabular summary

```

Data(x,y) → ModelDef→ Design matrix→ Condition of X → Feasibility of ANOVA → ANOVA.Reg [F-test, LOF-test]

MatLabProg 7-2:

```
 %
% anova2015.m(30-7-97) 22-5-15
%
function [sig,Fcal,Ftable] = anova2015(X,x,y)

%%
%Called functions : R-Squared2015.mftest2015.m ;
 %%

%%
```
~~~~~~

```
if nargin < 3
 clean
 data_xy
end
```

```
H0 = 'RegMod';alpha = 0.05;
[a_LLS,ycal,residy] = Formulas_LS(X,x,y)
%%
 [np,npar]=size(X); one = ones(np,1);
[np,npar] = size(X);
sst= y' * y;
ybar = one' * y/(one' * one);
ymean= one * ybar;
ssmean = ymean' * ymean;
sscorr = (y-ymean)' * (y-ymean);
%
ss_Model = (ycal -ymean)' * (ycal-ymean);
ss_Residy= residy' * residy;
ss_Toty= (y-ymean)' * (y-ymean);
%
df_Model = npar-1;
df_Residy = np-npar;
df_Toty = np;
df_TotyCorr = np-1;

%
Meanss_Model =ss_Model/df_Model;
Meanss_Residy=ss_Residy/df_Residy;
Meanss_Toty=ss_Toty/df_TotyCorr;
%

F_RegModel = Meanss_Model/Meanss_Residy;
Fcal =F_RegModel;
```

```
%
%%
R_Squared2015
%%
%%

%%
tab_anova2015
ftest2015
oo_anova2015
```

```
MatLabProg 7-3:

%
%  R_squared2015.m  (R S Rao)   25/3/2K 16/3/97 ;
10/04/93;
%%
  R_squared = 1- ss_Residy/ss_Toty;
  R_squared_adjsted = 1-
(Meanss_Residy/Meanss_Toty)
  %%
```

F test: It is named in honor of Sir Ronald A. Fisher. He introduced in 1920 a new statistic as the ratio of two variances.F-statistic follows F-distribution under null hypothesis. In the context of regression, the gross statistical validity of a model (functional relationship between y and x) is assessed from comparison of the calculated value with table value and also from probability value.

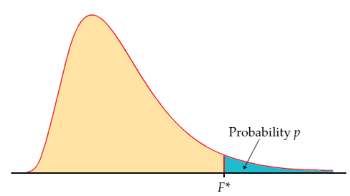
F_statistic tests the null hypothesis that each of regression coefficients are equal to zero. In other words, the model is with only one independent variable which is the mean of values of dependent variable (y). If the null hypothesis (H_o) fails, the alternate one (H_A) is true that the model with independent variables explain the variation in y (chart 7-3, output 7-1)).

Chart 7-3: 'F_ratio test'

Necessary conditions

sums of squares:	'statistically independent' 'chi-squared distribution

H_o :All of the regression coefficients are zero
H_A :All of the regression coefficients are not equal to zero



F_table value (also called critical value of F)
with probability p lying to its right

If	data values are independent normally distributed common variance
Then	Sum of squares follow chi-square distribution

Example 7.1:critical value of F is 3.40
F_cal from ANOVA > F_critical_Value
Inference : F_Cal is acceptable as its chance occurance < (p = 0.05)

Output 7-1

X,yysimul,y-ysimul

1.00001.00002.04382.00000.0438
1.00002.00004.05454.00000.0545
1.00003.00005.98916.0000 -0.0109
1.00004.00007.98268.0000 -0.0174
1.00005.00009.975410.000 -0.0246
1.00006.0000 12.011 12.00000.0113
----
a_LLS =
0.0501
1.9884

```
anova =
np: 6
npar: 2
ss_Model: 69.19
 ss_Residy: 0.0031464
 ss_Toty: 69.193
df_Model: 1
 df_Residy: 4
 df_Toty: 6
 df_TotyCorr: 5
Meanss_Model: 69.19
 Meanss_Residy: 0.00078659
 Meanss_Toty: 13.839
F_RegModel: 87961
probFvalueRegModel: 0.0025288
 R_squared: 0.99995
 R_squared_adjsted: 0.99994
replicates: 'No'
 LOF: ''
PE: ''
```

`~~~~~~
sum ofdegrees of mean F_Reg
squares freedomsquares
`~~~~~~
Model69.18981 69.189887961.4062
Residualy0.003146 4 40.00078659
Totaly 69.1929 6 13.8386
`--

Information: RegModis acceptable at 0.05 significant level since,Fcal:87961.4062 > F_table_value = 21.2(with df.Residy=4,df.Model 1):
Inference_ANOVA :Chanceoccurance of RegMod < 0.05(or <5%) probability
KB: F is scale independent

Probability (F):It is calculated from CDF (cumulative distribution function) and the value corresponds to probability that Ho is true to an extent to (1-prob(F)) * 100 percent.

Example 7.2: Ifprob(F) = 0.010, it means that there is 1chance in 100 that all regression coefficients are equal to zero. In other words, that at least some of regression parameters are non-zero and regression equation does have validity in explaining variation of y (chart 7-4, output 7-2). In statistical sense, independent variables are not pure random with respect to y.

Chart 7-4

	Coefficient	SE	t_cal	Prob(t)	95% CI	
a1	0.583884	0.016	36.40	<00001	0.5508	0.6169
a0	-0.845346	1.106	-0.76	0.45203	-3.124	1.434

✓	95% CI of a1(0.58) is in the range of 0.55 to 0.62 is reasonable
–	95% CI of a0(-0.84) is in the range of -3.1 to 1.4 is less reliable
	Data is to be acquired with ED and with more number of points near origin.

```
MatLabProg 7-3b
%
% ftest2015.m(30-7-97) 22-5-15
%
%function ftest2015(Fcal)

%% F probability
%
 x = F_RegModel
%

xunder = 1./max(0,F_RegModel);
 xunder(isnan(F_RegModel)) = NaN;
 probF = fcdf(xunder,df_Model,df_Residy);
 [probFvalueRegModel]=probF;

%%
Ft_table;
Ftable = F_TABLE01(df_Residy,df_Model);

%%
chr=' ';no =14;
b10 = setstr(ones(1,no)*eval('chr'));
alpha = 0.05;
 atsiglevel = [' at ' num2str(alpha),' significant
level'];
 a1a =['Fcal:',num2str(Fcal)];
 a1b = ['F_table_value = ',num2str(Ftable),'(with
df.Residy=',num2str(df_Residy),',df.Model
',num2str(df_Model),'):',];
 inf2= ['Chanceoccurance of ',H0, ' < ',
num2str(alpha),'(or <', num2str(alpha*100),'%)
probability '];
if Fcal >Ftable
 sig = 1;
 disp(['Information: ',H0, 'is acceptable',
atsiglevel])
 disp([b10,'since,',a1a, ' > ' , a1b])
else
 sig = 0;
 disp(['Information:', H0, 'is not
acceptable',atsiglevel])
 disp([b10,'since,'a1a, ' < ' , a1b])
```

```
 inf2= ['Chanceoccurance of ', H0, '>',
num2str(alpha),'(or >', num2str(alpha*100),'%)
probability '];
end
disp(['Inference_ANOVA :' , inf2])
disp([b10,'KB: F is scale independent '])
```

Output 7-2	
>> autotest_ftest	
H0 = Equal Variance SS1 = 0.0019 SS2 = 0.0014 df1 = 15 df2 = 15 H0 = Equal Variance Ftable = 2.4000 H0 : Equal Varianceis acceptable since,Fcal:2.7632 > F_table_value (df1=15,df2= 4):2.4 sig = 1 Fcal = 2.7632 Ftable = 2.4000	SS1 = 5.1860e-06 SS2 = 9.0600e-06 df1 = 2 df2 = 3 H0 = Equal Variance H0 : Equal Varianceis not acceptable since,Fcal:1.1647 < F_table_value (df1=2,df2= 3):19.16 sig = 0 Fcal = 1.1647 Ftable = 19.1600

Lack of fit (LOF):The necessary conditions are same as those for LLS. Replicate response (y) values at one or more X values are needed. The Error sum of squares is decomposed into two components viz. Pure error and LOF. Then, F test is performed for inference (output 7-3).

Output 7-3:	

LOF is insignificant as
flof(6.5) < table value (8)
Advice : Accept the model
 	
LOF is highly significant as
flof(14.14) > table value (8)
Model is not adequate
Remedy : Useanother model with more terms
	MatLabProg 7-4a
	%
	%dem_inference_LOF.m(R S Rao)1-11-96
	%
	table_lof=8;
	flof = 6.5;
	inference_LOF
	
	flof = 14.14;
	inference_LOF

MatLabProg 7-4b

```
%
%inference_LOF.m(R S Rao)1-11-96
%
 b20 = blanks(20);b40=blanks(40);b10=blanks(10);b5=blanks(5);
```

```
if flof > table_lof
disp(['LOF is highly significant as '])
disp([b10,'flof(',num2str(flof),') > table value (',num2str(table_lof),')'])
disp([b10,' Model is not adequate'])
disp([b5,'Remedy : Useanother model with more terms '])
zlof = 1;
end
if flof <table_lof
disp(['LOF is insignificant as'])
disp([b10,'flof(',num2str(flof),') < table value (', ..
num2str(table_lof),')'])
 disp([b5,'Advice : Accept the model '])
 zlof = 0;
end
```

MatLabProg 7-4c

```
%
% LOF2015.m (R S Rao)25/3/2K16/3/97 ; 10/04/93;
%
function [zlof,sslof,sspe,unique] = LOF2015(X,x,y)
%
% LOF and PE
%
%jy:Mean replicate response
%structured as y

 %%
%Called functions : jlof.m ;F_TABLE0.m ; oo_LOF.m
%
 %%
if nargin == 0
x = [1:6]';
 x = [1:6 2 4 5 6 ]';
% x = [1:6]';
 [np,~] = size(x);one=ones(np,1);
 y = one +2*x+1.01*randn(np,1);X= [one x];
end
zlof = [];sslof = []; sspe=[]; unique=[];
%
 [a,sda,r] =Formulas_LS(X,x,y);
 [npar,ca] = size(a);
 [np,cx] = size(x);
 ycal = y - r;
 [z] = sortz([x,y,ycal]);
 x = z (:,1); y = z(:,2); ycal= z(:,3);
%
%

 [np,npar] = size(X);
 [jx,jy,unique] = jlof2015(x,y);
 f = unique;
%


%
% -------------------- KB(LOF) --------------
 disp(['np : ', sprintf('%2.2g',np)])
 disp(['unique : ', sprintf('%2.2g',unique)])
if unique == np
disp(' ')
disp(' No replicates __ LOF & PE calc. not possible')
```

MatLabProg 7-4d

```

%
%jlof2015.m (R S
Rao)25/3/2K16/3/97 ; 1-11-96
; 10/04/93;
%
function [jx,jy,unique] =
jlof(x,y)
if nargin == 0
x = [1:6 2 4 ]';
% x = [1:6]';
y = 2*x;
end

 tol = 1e-12 ;
zx=x;zy=y;
%
%%
 [x,y]= xysort([zx,zy]);
 [x y]
jy = [];jx = [];
%
%check unique
 tocontinue = 1; unique = 1;
 %%
%
 %%
while tocontinue
unique = unique + 1;
 xrep=[];yrep=[];
 [rx,cx] = size(x);
xrep = [xrep;x(1,:)];
yrep = [yrep;y(1)];
n = 1;
next = 1; j = 1; z =1;
%
while next
%
j = j +1; [rx1,cx1] =
size(x);
if j<= rx
z= abs(x(j)-x(j-1)) ;
else
unique = unique -1;
```

```
return
elseif np > unique
 disp([sprintf('%2.2g',np-unique),' Replicates___ LOF &
Pure ErroR calculated'])
end
% -----------------------------------------------

if unique < np
%
if any(abs(jy-y) > eps)
 sslof = (jy-ycal)' * (jy-ycal);
 sspe= (y-jy)' * (y-jy);
 flof = (sslof/(f-npar))/(sspe/(np-f));
%
 df_lof = unique-npar;
 df_pe = np-unique;
end
end
 xunder = 1./max(0,flof);
 xunder(isnan(flof)) = NaN;
 probF = fcdf(xunder,df_lof,df_pe);
 [probFvalueLOF]=probF;
%
Ft_table;
table_lof = F_TABLE01(df_lof,df_pe);
%
disp(['probF_LOF = ',num2str(probFvalueLOF)])
%%
%----------KB_LOF -------------------
 inference_LOF
%%
 oo_LOF
 %%
% In Designed Expts df_PE is nearly =df_LOF
% IfF_LOF > F_Table value
% ThenReject regression model
% IfF_LOF < F_Table value
% ThenAccept regression model
```

```
z = -1;
end
 rep = 1;
if z < tol
if z ~= -1
n = n+1;
xrep = [xrep;x(j,:)];
yrep= [yrep ;y(j,:)];
end
else
rj = j;
 rep = 0 ;
 next = 0;
end
if z == -1
 rep = 0;
 next= 0;tocontinue = 0;
end

if rep ==0
 xtemp = x(rj:rx);
 ytemp = y(rj:rx);
end
end% whilenext
%
 [m,n] = size(yrep);
avey = sum(yrep)/m;
%
for i = 1:m
jy = [jy;avey];
jx = [jx;xrep(i)];
end% for i
%
x = xtemp;
y = ytemp;
 [mx,nx]=size(x);
%
if mx == 1& z ~= -1
 jx = [jx;xtemp];
 jy = [jy;ytemp];
 tocontinue = 0;
end
%
end% while continue
%%
```

```
>>  [zlof,sslof,sspe,unique]  =
LOF2015

1 3.9283
2 5.4098
2 5.2019
```

```
 probF_LOF = 0.19365
LOF is insignificant as
flof(2.5468) < table value (15.98)
 Advice : Accept the model
```

3 8.9353 4 6.5198 4 7.9513 5 10.908 5 11.535 6 12.975 6 10.761 np : 10; unique :6 4 Replicates___LOF & Pure ErroR calculated	LOF = flof: 2.5468 probFvalueLOF: 0.19365 LOF: 0 Ftablevalue: 15.98 sslof: 9.4086 sspe: 3.6943 df_lof: 4 df_pe: 4 sslof : 9.4086sspe: 3.6943 LOF : variation of group means about the line PE: Variation within the groups zlof = 0
ans = 1 3.1085 2 5.4433 3 6.9586 49.368 5 10.229 6 13.555 np :6 ;unique :6 No replicates __(np-unique) =0 LOF & PE calc. not possible	zlof = [] sslof = [] sspe = [] unique = 6

8. Advanced residuals and regression coefficients

The standard deviation and t-values of regression coefficient, and standardized regression coefficient are used in least squares analysis. The advanced statistics like confidence intervals, joint confidence contours follow now (chart 8-1). Theslopeandintercept in linear regression are estimated simultaneously satisfying the condition of minimization of sum of squares of residuals in y.The magnitude of correlation coefficient of regression parameters throws light on the elliptical contour of any two values.

Chart 8-1: parameter statistics of regression coefficients		
sda	:	Standard deviation
ta	:	Student t values
standa	:	Standardized
ccpar	:	Correlation coefficient
CIpar	:	Confidence Interval
JCIpar	:	Joint CI

Formulae 8-1:	MatLabProg 8-1
$H = inv(X^T * X)^T$ $h_{NPARx1} = diag\{inv(X^T * X)^T\}$ $r\_par = \left(\sqrt{h}\right)^T * inv(X^T * X)^T * \frac{1}{\left(\sqrt{h}\right)^T}$ Formula	`h = diag(inv(X'*X));` `rab = h.^(1/2)' * inv(X'*X) * h.^(-1/2)`

Example 8.1	
~~|~~~~~~~~|~~~~~~~~~|~~~~~~~~~~~~~~~~ **X,|ysimul |randNoise| y(=ysimul+** **one x | | | randNoise)** -----------------|--------|---------|---------------- 1.00001.00005.0000 -0.02284.9772 1.00002.00009.00000.24689.2468	rab = 0.0936 infMat = 621 2191 invInfMat =

1.00003.000013.00000.003413.0034 1.00004.000017.00000.151017.1510 1.00005.0000 21.0000 -0.111120.8889 1.00006.0000 25.00000.127625.1276	0.8667 -0.2000 -0.20000.0571

Confidence Intervals (CI) ofslope and intercept: If slope and intercept are not linearly correlated based on Pearson correlation coefficient, their confidence contours adhere to t- and z-distributions for small and large samples respectively (KB.8-1). This also holds good when two successive regression parameters are not significantly correlated, individual confidence intervals are calculated pairwise.

KB. 8-1: confidence interval (CI)of slope and intercept of a straight line

If	Corrcoef(a0,a1) < small
Then	Confidence interval (CI) of parameter
If	Confidence interval & NP > 30
Then	LU_a_0 = $a_0 \pm$ Z (CL) * SDa_0LU_a_1= $a_1 \pm$ Z (CL) * SDa_1
If	Confidence interval & NP <= 30
Then	LU_a_0=$a_0 \pm$ t (CL,Z) * SDa_0 LU a_1 =$a_1 \pm$ t (CL,Z) * SDa_1

If	Corrcoef(a0,a1) > significant
Then	Joint Confidence interval (JCIP)

Joint confidence (JC) contours ofparameters (CP): If two successive regression parameters are significantly correlated, joint confidence contours/surfaces are appropriate (chart zz). The JCCP is an ellipse for two parametric regressions. The profile is an ellipsoid/hyper-ellipsoid for multi (3 and higher)-parametric regression analysis.

Formulae 8-2:

$$b1 = slope + \frac{rab * sb}{sa * (a - int)} + sb * \sqrt{\frac{\left[(1 - rab \wedge 2) * (2 * f - (a - int) \wedge 2)\right]^2}{sa^2}}$$

Formula

$$b1 = slope + \frac{rab * sb}{sa * (a - int)} - sb * \sqrt{\frac{\left[(1 - rab \wedge 2) * (2 * f - (a - int) \wedge 2)\right]^2}{sa^2}}$$

Formula

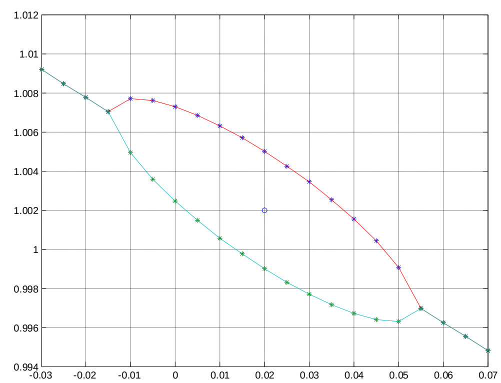
MatLabProg 8-2

```
%
% ellipConfConta0a1.m
%
x = [];
y = [];
for a = -0.03 :0.005:0.07
x = [x;a];
b1 = slope + rab * sb/sa*
(a - int) + sb*sqrt((1-
rab^2)*(2*f-(a-
int)^2/sa^2));
b2 = slope + rab * sb/sa*
(a - int) - sb*sqrt((1-
rab^2)*(2*f-(a-
int)^2/sa^2));
y = [y;b1,b2];
end
```

$b1b2 = [b1,b2]$ Formula	

<pre>%
% Formulas_ellipConfCont.m (R S Rao) 30-8-1993
%
 	
function [x,y] = Formulas_ellipConfCont(X,x,y)
if nargin <3
int = 0.02;
slope = 1.002;
sa = 1.13e-2;
sb= 1.91e-3;
rab = -0.85;
f = 4.46;
usage('[x,y]', 'Formula_ellipConfCont','(X,x,y)')
end
%
ellipConfConta0a1
%
figure,
[x,y],plot(x,y,'*',x,y),grid,hold on
plot(int,slope,'bo')</pre>	

Example 8.2	

<pre>Example 8.2
int = 0.02;
slope = 1.002;
sa = 1.13e-2;
sb= 1.91e-3;
rab = -0.85;
f = 4.46;</pre>
	

Advanced residuals: In linear least squares, ordinary residuals, variance and sdy are illustrated. Here, advanced residuals viz. studentized, Jack-knife, PRESS etc. are described. DFBETAS, likelihood/Cook distances etc. also derived from residuals.

PRESS (predicted residual error sum of squares: The least squares parameters are calculated by excluding ith point. Then, ycali and residyi are calculated for excluded point. The process is repeated for all (NP) data points. The studentized version of PRESS is not discussed here.

Formulae 8-3:	MatLabProg 8-3

$$ti = \frac{ei}{\sigma i * (1 - vii)^{0.5}}$$ ti follows Student t-distribution with df = NP-par -1
 	
☞ Valid only if residuals are
	<pre>%
	% studres.m
	%
	function [studentizedResiduals] = (X,x,y)
	%
	prin = 0;
	if nargin < 3
	clean
	usage('[studentizedResiduals]',</pre>

homosedastic

```
'studres','(X,x,y)');
data_xy
end

%
 [a,ycal,resid] = Formulas_LS(X,x,y);
 [sda,ycal,resid,vary,sdy] = Formulas_Resid(
X,x,y,a,prin);

[Catcher,hat,diagHat,cutoff_h]=Formulas_hat(X,x,y);
%
% Studentised Residual
%
 [xr,xc] = size(x);

 sde = (ones(xr,1)-diagHat) * vary;
 studentizedResiduals = resid./sde;
```

```
              Cook's Distance
..........................................
Xyresidy standRes
............................................
1.00001.00001.9762 -0.0418-1.1409
1.00002.00004.03880.02140.5829
1.00003.00006.03400.01720.4693
1.00004.00008.06090.04461.2178
1.00005.00009.9984-0.0173 -0.4709
1.00006.000011.990  -0.0241 -0.6583
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

.................................
 #CookDista0a1
.................................
1.00000.11020.07711.9869
2.00000.00740.00452.0020
3.00000.00220.01302.0000
4.00000.01470.01501.9979
5.00000.00480.01532.0015
6.00000.03670.00172.0066

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
%
% cook.m (R S Rao) 30/04/93
%
function [cookDist]=cook(X,x,y)
%
if nargin < 3,
 clean
 data_xy
end
prin = 0
%
%inf_cook
%
%one point leave out
%
n = length(x);
zd= [];
za1 = [];
[a,ycal,resid] = Formulas_LS(X,x,y );
[ycal,residy,sdy] = ordResid(X,x,y,a)

[mx,nx] = size(x);
for i = 1 : n
tx = [X(1:i-1,:);X(i+1:n,:)];
ty = [y(1:i-1,:);y(i+1:n,:)];
 b = tx\ty;

d= (a -b)' * X' * X * (a -b)/(nx*sdy);
zd =[zd;d];
za1 = [za1,b];
cookDist = zd;
end
```

```
         Mahalanobis distance (Mah_Dist)

-------------------------------------------
 X,yysimul,y-ysimul
-------------------------------------------
1.00001.00001.91712.0000 -0.0829
1.00002.00003.97414.0000 -0.0259
1.00003.00005.96746.0000 -0.0326
1.00004.00007.94348.0000 -0.0566
```

```
%
%MD.m (R S Rao) 30/04/93
%
function [Mah_Dist] = MahDist( X,x,y )
%
if nargin < 3,
 clean
 data_xy
end
```

```1.00005.000010.0090 10.00000.0090```
```1.00006.000012.0313 12.00000.0313```
```---```
 	
```Mah_Dist =```
```1.7857```
```0.6429```
```0.0714```
```0.0714```
```0.6429```
```1.7857```
	```[Catcher,hat,diagHat,cutoff_h]=Formulas_hat(X);```
	```[np,npar] = size(X);```
	```one = ones(np,1);```
	```Mah_Dist =(np -1)* (diagHat-one./np);```
	```NO = [1:np]';```

9. State-of-knowledge and Future scope of regression

The exemplary datasets from basic chemistry are analysed in out text book 'computer applications in chemistry'. In this review, simulated data sets with very small number of points (4-10) are chosen to lessen number churning jugglery, to be independent of discipline, easy to remember, to appreciate expected/normally unexpected results, visualize the smooth transition of deterministic to fuzzy through probabilistic paradigms and to develop confidence to analyse large complicated experimental data not only for regression but many other computations (dimension reduction, clustering, classification/ pattern recognition).

Anhouse dataset achieve from select monographs have been in use in peer learning/training programs for post graduates/researchers in chemistry and imparting hands on experience in interdisciplinary workshops since 1990s. The earlier FORTRAN, Dbase-III-Plus and Turbo-prolog programs from this laboratory have been rewritten in MATLAB during the first few years of acquisition of MATLAB (early version in 1991). Additional data files from recent editions/versions of these research compendia are under the processing of culmination into dataset bases. The formats chosen are excel/mat (of Matlab), capsules of objects with interfaces rendering them readable in MATLAB software for calculation with m-function. The sub-task wise exerts of results and full datasets (on DVD) for practice will be discussed separately [164]. The programs with MATLAB specific matrix/tensor, object oriented, Boolean patterns, 2D-/3D-graphics/surfaces were developed with computational and scale up perspective. Since, FORTRAN was in the core in many earlier packages GAMESS, GAUSSIAN etc. and is used in pedagogic training here in post-graduation in chemistry, FORTRAN flavour/style may be found here and there. Mere algorithmic approach and algebraic solution was translated into programing language in last century. Still, it is the core of training for joining the high way of computations through the lanes of theory, derivations, and solution methods to arrive at result at ease. The input output stylish formats, GUI, pull-down/popup context sensitive menus etc. are the realm of package developers and not the prime focus of computational/ pedagogic world.

The yester years' practice was analysing a piece of data of individual's interest with a complete trust on the jargon and call it a day. It is no doubt coveted and continues with most end users of inter disciplinary/sometimes core science groups. The computational intelligence is at high end, while knowledge based systems for input check have been in routine practice now. The choices of methods have been smartly implemented in Berny algorithm of GAUSSIAN, Jaguar of Schrodinger, tool-boxes of MATLAB to name a few intelligent software categories. Yet, the check for suitability from necessary conditions and resorting to remedial measures is scarce. In fact, more important aspect is paradigm shift to imparting this culture in learning process through teaching/research pedagogy with simple as possible simulated (noise free and real life like) datasets from bottom of methods in mathematics/statistics/nature_inspired_procedures.

The modular approach of weighted regressions, support vectorregressions and advances in tests for normality with critical case studies from chemometrics/chemical biology literature in this decade will be reported [164].

Knowledge based numerical computation

LLS2015 is designed for paired real variables (x and y vectors).

Input check: InpChk2015 validates input vectors by checking for real numerical numbers. It generates error message even if one value is a character, imaginary value or string. Also, it also verifies the equivalency of number of data points in X and y tensors with appropriate messages.This approach amply demonstrates development of automatic modules with auto-check, correction, adaptive machine learning software.

Autotest_chkInpVec: The input for univariate statistical analysis described here is confined to real numerical values. The input checking program is developed to be sure of it to calculate statistics. The built in functions in matlab 'isnumeric' and 'isreal' and logical operators ('not' (~) , 'and' (&)) are used in chkInpVec.m. The default option in matrix algebra is to use column vector and it is included to convert a row vector into column with the appropriate message. A knowledge based program (can also be called expert system) inferring whether given data is scalar, vector (row/column), matrix (rectangular, square [skew, symmetric, upper or lower triangular]) is also incorporated. It also detects it to be a numerical or non-numerical for proceeding to inversion process. Further details of knowledge about integer/floating point/real/imaginary/quaternion elements in numerical and binary/Boolean/characters/strings of non-numerical elements in tensors is available. These tiny bits are useful in development of automatic/adaptive features in program module fabrication. Here, transparency, clarity of program steps in expert system mode is the motivation and not memory/speed etc.

```
Autotest_chkInpVec
Input
~~~~~
x =
0.9990
0.8880
0.7770
0.4440
Input isvector of real numeric data
 It is column vector
 ~~~~~~~~~~~~~~~~~~~
Input
~~~~~
x =
 1 2 3 4 5 6
Input isvector of real numeric data
It is row vector & hence converted into a column
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
x =
 1
 2
 3
 4
 5
 6
Input
~~~~~
x =
a, b, c,
 !Invalid input -- Non-numeric data
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
%
% chkInpVec.m(R S Rao20-2-1991); 2-7-15
%
function [x]= chkInpVec(x)

if nargin ==0
 x = [1:6];
end
%
xbak = x;
dispst('Input')
x
[np,r] = size(x);
for k = 1: np
valid(k,1) = [isnumeric(x(k))&
isreal(x(k))];
end
if any(~valid)
 dispst( '!Invalid input -- Non-numeric
data ');
return
else
 disp('Input isvector of real numeric
data')
end
%
%
[r,c] = size(x);

if c>1 & r==1
 dispst('It is row vector & hence converted
into a column vector')
 x = x'
return
end
if c==1 & r>1
 dispst('It is column vector')
end
```

```
% Autoest_chkInpVec.m
%

for ii = 1:3
zz = ii ;

switch zz
case 1
x = [1:6]; [x]= chkInpVec(x);
case 2
x =['a, ','b, ','c, '];[x]= chkInpVec(x);
case 3
x =[0.999 0.888 0.777 0.444]';[x]= chkInpVec(x);
otherwise
disp('test')
end
end
```

Future scope

Good computational laboratory practice (GCLP):Good laboratory practice (GLP) and good manufacturing practice (GMP) protocols are continual upgradation in different countries and in national quality maintenance/control organizations. The stipulations for QSAR models are published for righteous use in medical/pharma/food industries. However, software availability/popularity is one driving force in choice of a method in computational world. It is the need of hour to promote good computational laboratory practice (GCLP) stipulationsto enhance reliability and state-of-knowledge information for the data acquired with high effort/cost targeting high-focussed-precise-end-goals. The first step in computational data (CD) analysis is to look into primary data if already procured.But, it is preferable to design data acquisition schedule based on specific goals in a discipline.This step gives primary information about quality of data, its sufficiency and holes still present in data acquisition schedules, instruments, limitations in experimental design etc. Our in-house programs check for adequacy of data on hand for reasonable proposal of a hypothesis or endorsing/ refuting the earlier reports. It also includes limitations as well as failure conditions and remedial measures with next level/ alternate methodology. The next step is applying higher order computational techniques to infer from derived parameters, adhering to optimal path considering CPU time and accuracy demand. When data is inadequate to cope up with high end technology, lower order methods are preferred with a caution of its conclusion domain and also leaving room for upgrading experiments and data acquisition schedules to suit to high-end computations.

The concept of fit-for-task has gained popularity and it is a conglomeration of statistical test, international protocols, dynamic requirements in real life tasks and experts' propositions.It will document in detail the scope of use and also limits beyond which the current protocol is invalid or fails.

Scatter diagrams, regression lines/surfaces/Kohonen-maps for spacio-temporal data/information/ numerical knowledge evolved over last half a century enhancing the scope of inferencing tools beyond a host of calculated parameters. They are instrumental inexploratory statistical data analysis, transformation of data or projection tools are start-ups in understanding data. The (non-) parametrization, orthogonalization and projection pursuit methods are simple way of reducing the size of data tensor (number of points, dimensions in each way of multi (three-six and higher) way data). The confirmatory statistical/fuzzy/possibility analysis, checking with well tested knowledge bits of the task/discipline follows. One should conceive visualization is more than just a “pretty picture”. Visualisation is a third-eye probe of experimental-simulated-computed-output of terabytes to exabytes. These sizes are beyond datum-by-datum inspection/ identifying trends which was a coveted torch and respected in the beginning of 20^{th} century for tens to hundreds of data points. Effective visual data analysis must be based on strong mathematical foundations to reliably characterize salient features and generate new scientific knowledge. The focus of basic research should be round developing fundamental mathematical methods such as topology, statistics, high-order tensors, uncertainty, and feature extraction in tensor notation, Clifford (geometric algebra) and real-time true-color-multidimensional display. These pave an alternate route, subsuming earlier theories in to a unified one, opening a new widow, refuting earlier well accepted proposal(s). But, they are the signal posts at the cross-high ways or light-houses in the ocean of knowledge to avoid the consequences of accidents resulting in wreckage of the precious material. But, they remain as a test bed for long for unstinted six sigma limits validity for the prosperity and upholding the truth value of the proposed truth.

Regression 2016

```
%      DataAnalysis2016.m  (9-11-15 25/1/97, 7/10/92 R S Rao)
%      (Beta version 9.6, 9-8-16 under rigorous testing)
%
    clean
    diary off
!del output2016.txt
    diary output2016.txt

    %%
    Task = 0;hardModel= 0; SoftModel =0; CauseEffect = 0; DataDriven = 0; ModelFree=0;
    DistributionFree = 0;DimensionReduction = 0; MappingToHigherDimensions=0;
NoVariables_GT_NoPoints =0;
    NoPoints_GT_No_Variables=0;NatureInspingAlg=0; EnvlopEst =0;
%
    %%
    linearModel=0; replicateMeasurementsAreThere=0; ANOVA_required=0;
    advancedResid = 0;
%
    %%
    InformKBaseIntBits = 0;
    ExpertSystemAdvice=0;
    supportRefute = 0;
    CaseBase =0;
    Simul = 0;
    RealLife =0;
    State_of_Knowledge=0;
%
    %%
    UserChoice = 0;
    prin =0; graph=0;
%                 %
    %%
if Task
      Tasks = {'Response_Analysis';'Cause_effect relation';
'Classification/Distrimination';'Clustering';
'Pattern Recognition'}
end%%
%
    %%
if hardModel & CauseEffect
      Model_Hard = {'LLS';'LAD'; 'LMS';'PolyLS';'MLR';'EnvlopEst'}
end
if SoftModel & CauseEffect
      Model_Soft = {'PCR';'PLSR'; 'CR';}
end%%
%
    %%
    data_xy
    InpCheck_lls2015(X,x,y);
    [np,npar] = size(X);
    [np,xvariables] = size(x)
```

```
if xvariables > 1
     mlr2015
end

if xvariables ==1
     anova2015(X,x,y)
      lls2015
      LAD2015
      lms2015
      polyLS2015
end

if np < npar
     disp('No Least squares solution -Resor to Linear algebra Simplex')
return
end

if np > npar | np == npar
     [a,ycal,res] = Formulas_LS(X,x,y );
end

if np > npar
     [sda,ycal,resid,vary,sdy] = Formulas_Resid( X,x,y,a,prin )
end

if linearModel
%[linear] = kb_lr1(X,x,res);
end%%
%
    %%
if ANOVA_required
      [sst,ssfact,ssr] = formulas_anova(X,x,y,prin);
      Ftest2015
end

if replicateMeasurementsAreThere
      LOF_PE
end
%
if advancedResid
      [Catcher,hat,diagHat,cutoff_h]=Formulas_hat(X,x,y);
%
%  Regression parameterstatistics
%
      [stats] = regcoefstat(X,x,y);
      LLS_stats.regcoef= stats
%
%  Residual statistics
%
      [stats] = residstat(X,x,y)
      LLS_stats.res = stats;
%
%  statistical tests
%
      statTests
end%%


   %%

if ExpertSystemAdvice
      KBReport
```

```
end
if prin == 1
% tab_lr1(np,npar,a,sda,sdy,ta,standa,linear,prin)
end
if graph ==1
%graph
end


    %%
if supportRefute
     support_KB
     Refute_KB
end

if CaseBase
if Simul
        simul_CaseBase
end
if Reallife
        RealLife_CaseBase
end
end
if State_of_Knowledge
      State_of_Knowledge_MethodBase
end

   diary off
   edit output2016.txt
```

10. Knowledge based output for typical datasets

The statistical parametric estimation and residual analysis of simulated and typical data sets are reported with the suit of programs developed in this laboratory for regression analysis. The additional features are knowledge based inferences through IF-Then first order logic, a key of numerical as well as literal expert systems. The stepwise pedagogical approach adapted through sections 2 to 8 is to introduce calculation of parameters of simple straight line to multi-dimensional surfaces in cause-effect models. The derivations are separated (appendix), while Matlab functions implementing formulas are described side by side. This enables one to have at the first sight the titbits and later to focus on either on mathematical jargon or programing skills in matrix (tensor) notation. The simple as possible data sets with and without noise impart a smooth transition of utopian to real world noisy data. This throws light on behaviour of the methods of increasing care taking protection and robust procedure to combat when datasets do not adhere to the necessary conditions implied in any mathematical solution.

Dataset SI-1: It is a simple simulated data set without noise with knowledge based inferences.

Dataset: 1: for y = 0 + 1*x	
ycal = 1.0000 2.0000 3.0000 4.0000 5.0000 6.0000 resid = 1.0e-14 * 0.0666 0 -0.0444 -0.1776	→ No noise ✓ Regression parameters (slope and intercept are exactly equal to those used in model for simulation of data ✓ Residuals in y are zero (i.e. order of 10^{-14}) ✓ stand_a, t-values ✓ F regression <

-0.1776 -0.1776 >>	→ No minor/major process ॐ Analysis is adquate
− No replicates NO PE, LOF Obtain data with replicates + No heteroscedastic noise ✓ WLS not necessary + No outliers ✓ LMS, TLS, not necessary	

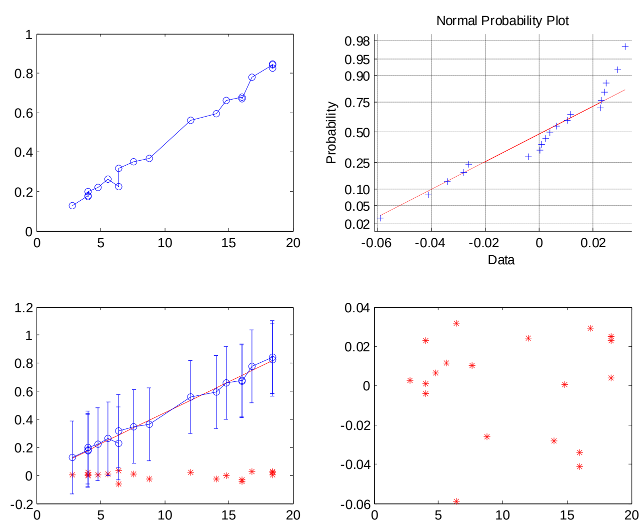
Dataset SI-2: Calibration of labetalolHcl:Kateman D-optimal design is used in choosing concentrations of analyte. It is clear that the points are equally distributed over concentration range and there more points in the beginning and end of study region.

labetalolHcl **NP :**

2.8	0.127	0.0025093
4	0.179	0.0009969
4	0.174	-0.0040031
4	0.201	0.022997
4.8	0.22	0.006322
5.6	0.261	0.011647
6.4	0.226	-0.059028
6.4	0.317	0.031972
7.6	0.349	0.01046
8.8	0.366	-0.026053
12	0.559	0.024247
14	0.596	-0.02794
14.8	0.66	0.00038518
16	0.679	-0.034127
16	0.672	-0.041127
16.8	0.778	0.029198
18.4	0.843	0.022848
18.4	0.845	0.024848
18.4	0.824	0.0038479

sdy =
0.026848

Par	sd	ta	stda
-0.00037161	0.00034829	-4.8206e-06	-1.067
0.044594	2.9237e-05	4.8561e-05	128.04



→ D-optimal experimental (Kateman) design for concentrations of analytes

→ Repition of select points

→ NP = 19 ; LMS, LLS, MLP_NN

Ref : [164]

Dataset SI-3: Data with fuzzy errors

#	x	y	Resid		
#	x	y	LMS	LLS	Fuzzy Reg
1	1	1.1	0	0.32	0.02
2	2	2	0	-0.04	-0.04
3	3	3.1	0.2	-0.2	0.14
4	4	3.8	0	-0.76	-0.08
5	5	6.5	1.8	0.68	1.70
			sdy_lms :1.0456	sdy_lls : 0.62823	----

a_lms	a_lls	sda_lls
0.2	-0.48	0.41393
0.9	1.26	0.1248

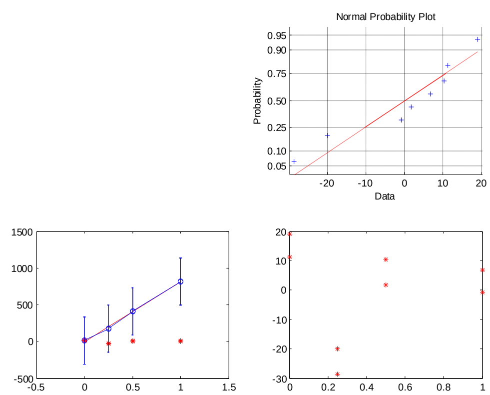
- – Data with fuzzy errors
- – Small set of data point NP =8 ; yrange very large : [8 to 800]
- + Regression Coeffient (slope) is similar to LLS
- – bur SD in LLS Slope is high
- + Residual range [1 to 30]
- + Difference in magnitudes of residuals for LMS and LLS is marginal compared to y values

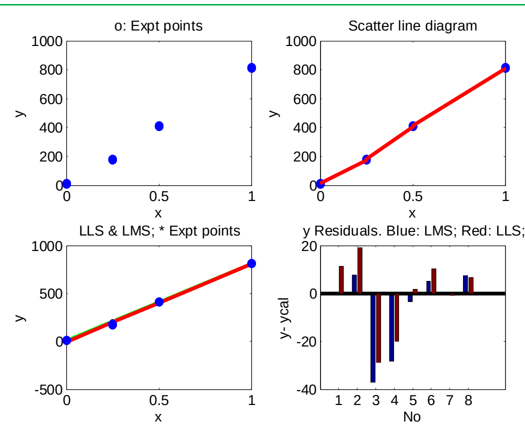
NP : 8

a_LMS	a_LLS	sda_LLS	LAD	Fuzzy Reg
8.19	-3.083	170.34	12.1	10.63
802.51	814.5	297.36	802.5	802.8

k	x	y	res_LMS	res_LLS
1	0	8.19	0	11.273
2	0	16	7.81	19.083
3	0.25	171.9	-36.917	-28.642
4	0.25	180.6	-28.218	-19.942
5	0.5	406	-3.445	1.8324
6	0.5	414.5	5.055	10.332
7	1	810.7	0	-0.71814
8	1	818.2	7.5	6.7819

sdy_LMS :19.6375 sdy_LLS : 17.6349





Dataset SI-4: The experimental Dielectric constant data versus DMSO content of aquo-DMSO mixtures is fitted into a athird order polynomial by least squares. The desired dielectric constant at desired DMSO content was calculated from parameters.

$y = a0 + a1 * x + a2 *x^2 + a3*x^3$

a0	a1	a2	a3
79.264	-0.136	0.003 728	0.000 052 45

Method: orthogonal polynomials

$y = a0 + a1 * x + a2 *x^2 + a3*x^3$			
10.00	78.20	78.22	-0.0243
18.58	77.90	77.82	-0.0873
20.00	77.50	77.61	-0.0150
32.56	76.90	76.98	-0.0776
40.00	76.40	76.43	-0.0322
52.01	74.90	74.90	0.0033
60.00	73.30	73.20	0.1031
65.01	71.7	71.77	-0.0693

Inference

- Very low residuals indicate the interpolation at intermediate of x is statistically valid.
- Measurement of dielectric constants of aquo-organic mixtures not more accurate than 0.1 under normal set of conditions and instruments

Data: Courtesy of Ref: [40]

Dataset SI-5: A small dataset, but with typical characteristics is presented in chartzz. A simple MLR model show the results to be normal at first sight.

Phase 1

```
.....................................................................
#←------x----→ y      res_LLS      hatMat(i,i)
......................................................................

1            1            1            3      -0.12857       0.47143
2            1            2            4      -0.14286       0.28571
3            1            3            5      -0.15714       0.18571
4            1            4            7       0.82857       0.17143
5            1            5            7      -0.18571       0.24286
6            0            6            8   -1.4211e-14             1
7            1            7            9      -0.21429       0.64286
sdy: 0.45513 ; vary = 0.2071
```

```
correlation matrix
        x1     x2     y
--------------------------
x1     1.00
x2    -0.41   1.00
y    -0.37   0.99   1.00
        x1     x2     y
```

```
Angles between column vectors
        x1     x2     y
--------------------------
x1     0.00
x2     40.62   0.00
y     33.41   9.07   0.00
x1    x2y
svd(x):11.98
        1.575
```

```
    ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        a,          sda,         standErra,       standa,      ta
     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      1.9143      0.72576        1.5946       3.0526       2.6376
         0.2      0.53852        1.1832      0.23664      0.37139
      1.0143     0.094221       0.20702      0.20998       10.765
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Df = 4
Data: Courtesy from
```

R D Cook, S Weisber, Residuals and influence in regression, Chapman and Hall, New York(1982)

Phase 2:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        a,          sda,          standErra,       standa,       ta
        -1.5     3.1364e+08    6.7109e+07   -1.0066e+08   -4.7825e-09
0    3.1364e+08    6.7109e+07           0             0
       1.1058       0.88747       0.18989       0.20997         1.246

   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
correlation matrix
        x1     x2     x3
--------------------------
x1    NaN
x2    NaN   1.00
x3    NaN   0.98   1.00
        x1     x2     x3
--------------------------
```

```
...................................................................
                      #<------x---->  y      res_LLS      hatMat(i,i)
....................................................................

1             1             1             3        3.3942      0.28606
                        2             1             2             4        3.2885      0.14423
                        3             1             3             5        3.1827     0.074519
                        4             1             4             7        4.0769     0.076923
                        5             1             5             7        2.9712      0.15144
                        6             1             7             9        2.7596      0.51683
```

Dataset SI-6: The x variable is age of a child in months at first word and y is Gesell adaptive score of 21 children with cyanotic heart disease. This study was carried out at University of California at Los Angeles. Mickey, Dunn and Clark analyzed this data in 1967 and have been reanalyzed extensively.

```
       ~~~~~~~~~~~~~~~~~~
          #        h(i,i)
      ---------------------------
          1      0.047922
2      0.15451
          3      0.062816
          4      0.070545
          5      0.047922
          6      0.072619
          7       0.05799
          8       0.05667
          9      0.079858
         10      0.072619
         11      0.090755
         12      0.070545
         13      0.062816
         14       0.05667
         15       0.05667
         16      0.062816
         17      0.052108
18      0.65161
         19       0.05305
         20       0.05667
         21      0.062816
```

```
%
%       hat.M 22/05/1995  (R S Rao)
%
function [h]=hat(X,x,y)
if nargin <3
           inf_hat
       load cookb_03.dat
       x = cookb_03(:,1);
       y = cookb_03(:,2);
      [r,c] = size(x);
        one = ones(r,1);
        X =  [one x];
end
       [r,c] = size(x);
        one = ones(r,1);
       hat2 = X * inv(X'*X) * X' ;
       h = diag(hat2)
        ycal = hat2 * y;
        res = ycal -y;
        pred_res = res./(one - h)
        press = pred_res'*pred_res
       [r,c]= size(hat2);
        b3 = blanks(1);

for i = 1 : r
           z = int2str(i);
for j = 1 : i
           z = [ z,b3,sprintf('%.2f',hat2(i,j))];
end
          disp([z])
end
       number = [1:r]';
       [number, h]
```

Phase 1:Hat matrix is a square symmetric matrix of size equal to NP and not calculated routine. The lower triangular matrix shows that hat (2,2) and hat(18,18) have large numerical values 0.15 and 0.65, while the range of all other elements is 0.05 to 0.09. That is why sum(diag(hat)) is 2.0

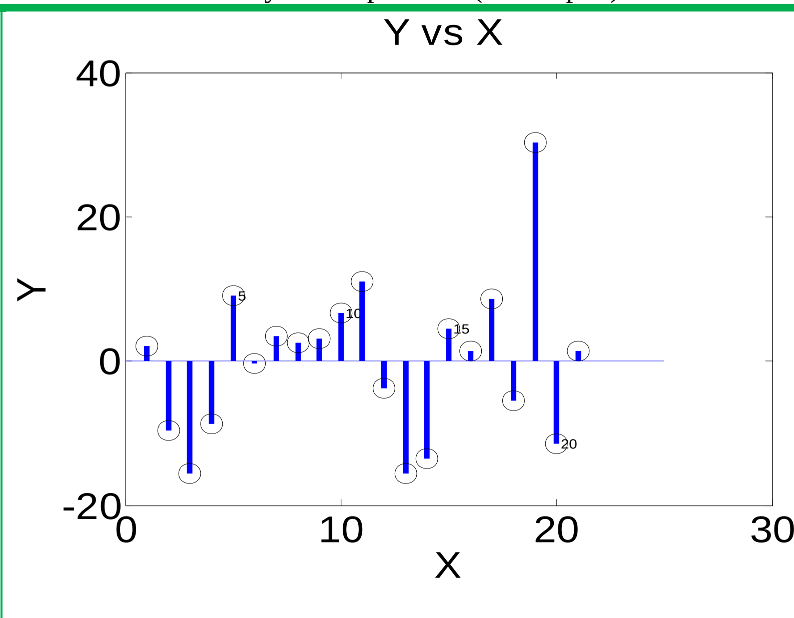
```
1 0.05
2 0.05 0.15
3 0.05 0.01 0.06
4 0.04 -0.00 0.07 0.07
5 0.05 0.05 0.05 0.04 0.05
6 0.05 0.10 0.03 0.02 0.05 0.07
7 0.05 0.08 0.04 0.03 0.05 0.06 0.06
8 0.05 0.02 0.06 0.06 0.05 0.03 0.04 0.06
9 0.04 -0.01 0.07 0.07 0.04 0.02 0.03 0.06 0.08
10 0.05 0.10 0.03 0.02 0.05 0.07 0.06 0.03 0.02 0.07
11 0.04 -0.02 0.07 0.08 0.04 0.01 0.03 0.07 0.08 0.01 0.09
12 0.04 -0.00 0.07 0.07 0.04 0.02 0.03 0.06 0.07 0.02 0.08 0.07
13 0.05 0.01 0.06 0.07 0.05 0.03 0.04 0.06 0.07 0.03 0.07 0.07 0.06
14 0.05 0.02 0.06 0.06 0.05 0.03 0.04 0.06 0.06 0.03 0.07 0.06 0.06 0.06
15 0.05 0.02 0.06 0.06 0.05 0.03 0.04 0.06 0.06 0.03 0.07 0.06 0.06 0.06 0.06
16 0.05 0.01 0.06 0.07 0.05 0.03 0.04 0.06 0.07 0.03 0.07 0.07 0.06 0.06 0.06 0.06
17 0.05 0.03 0.06 0.06 0.05 0.04 0.04 0.05 0.06 0.04 0.06 0.06 0.06 0.05 0.05 0.06 0.05
18 0.06 0.30 -0.05 -0.07 0.06 0.17 0.13 -0.03 -0.09 0.17 -0.11 -0.07 -0.05 -0.03 -0.03 -0.05 -0.00
0.65
19 0.05 0.07 0.04 0.04 0.05 0.06 0.06 0.04 0.03 0.06 0.03 0.04 0.04 0.04 0.04 0.04 0.04 0.10 0.05
20 0.05 0.02 0.06 0.06 0.05 0.03 0.04 0.06 0.06 0.03 0.07 0.06 0.06 0.06 0.06 0.06 0.05 -0.03 0.04
0.06
21 0.05 0.01 0.06 0.07 0.05 0.03 0.04 0.06 0.07 0.03 0.07 0.07 0.06 0.06 0.06 0.06 0.06 -0.05 0.04
0.06 0.06
```

Phase 2:Full set analysis: Least squares show that point 19 has high residual (30.2) while all others are spread between -15 to +15. This point is removed and the analysis is repeated. (Rous4/p49)

```
              Full Data
-----------------------------------
                ----
a_LMS      a_LLS      sda_LLS
sda_LLS/sdy
-----------------------------------
                ----
120       109.87       55.862
5.06
 -1.5       -1.127       3.419
0.31
-----------------------------------
                ----


sdy_LMS :12.4826 sdy_LLS : 11.0229
var_LMS : 155.8        var_LMS :
121.5

df = 21-2 =19
```



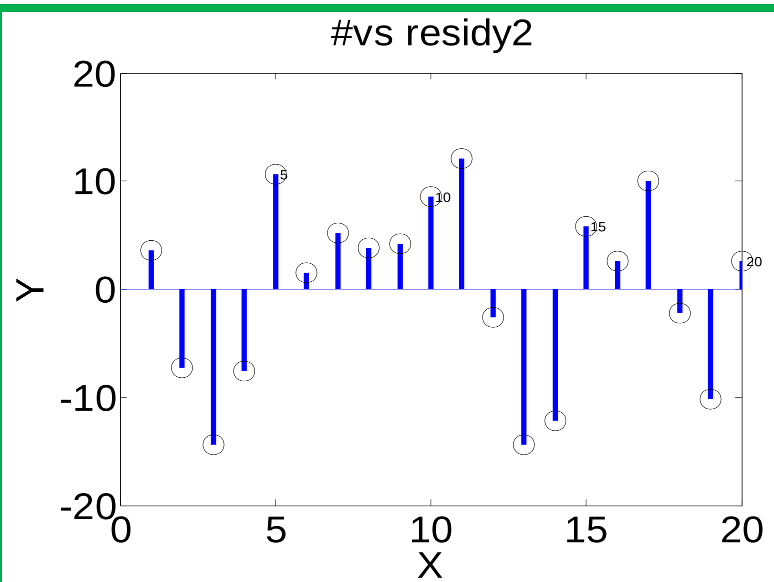
k	x	y	res_LMS	res_LLS
19	17	121	26.5	30.285

Phase 3: The all residuals now are in the range to -14 to +14 and sdy reduced to 8.6 from 11.02. LMS parameters remained same.

```
        19th Point deleted
------------------------------------
--
a_LMS   a_LLS  sda_LLS sda_LLS/sdy
------------------------------------
---
120       109.3    34.254    3.97
-1.5       -1.192.1008    0.24
------------------------------------
---
sdy_LMS :11.2008 sdy_LLS : 8.6282
var_LMS :125.46  var_LLS : 74.446
```

Data: Courtesy from

- M R Mickey, O J Dunn, V Clark, Computers and Biomedical research 1 (1967)105-9
  Note on use of stepwise regression in detecting outliers.
- R D Cook, S Weisber, Residuals and influence in regression, Chapman and Hall, New York(1982)



k	x	y	res_LMS	res_LLS
1	15	95	-2.5	2.031
2	26	71	-10	-9.5721
3	10	83	-22	-15.604
4	9	91	-15.5	-8.7309
5	15	102	4.5	9.031
6	20	87	-3	-0.33406
7	18	93	0	3.412
8	11	100	-3.5	2.523
9	8	104	-4	3.1421
10	20	94	4	6.6659
11	7	113	3.5	11.015
12	9	96	-10.5	-3.7309
13	10	83	-22	-15.604
14	11	84	-19.5	-13.477
15	11	102	-1.5	4.523
16	10	100	-5	1.396
17	12	105	3	8.65
18	42	57	0	-5.5403
19	17	121	26.5	30.285
20	11	86	-17.5	-11.477
21	10	100	-5	1.396

```
%cook03.m
 clean
        load cookb_03.dat

        x = cookb_03(:,1);
        y = cookb_03(:,2);
       [r,c] = size(x);
         one = ones(r,1);
         X =  [one x];
        hat2 = X * inv(X'*X) * X' ;

        [a,ycal,residy] = Formulas_LS(X,x,y);
```

```
        figure,plot(residy,'r*'),hold on, plot(residy),hold off
        title('#vs residy')
        [a_LMS] = lms2015(X,x,y)

plot(x,y,'x'),title('x vs y')


figure,plot(ycal,residy,'r*'), title('ycal vs residy')
figure,line_gr2(residy)


% 19 point deleted
x2 = [x(1:18,1);x(20:21,1)];
y2 = [y(1:18,1);y(20:21,1)];
[r2,c2] = size(x2);
         one = ones(r2,1);
         X2 =  [one x2];

         [a_LMS] = lms2015(X2,x2,y2)
          [a2,ycal2,residy2] = Formulas_LS(X2,x2,y2);

         figure,line_gr2(residy2)
        title('#vs residy2')
```

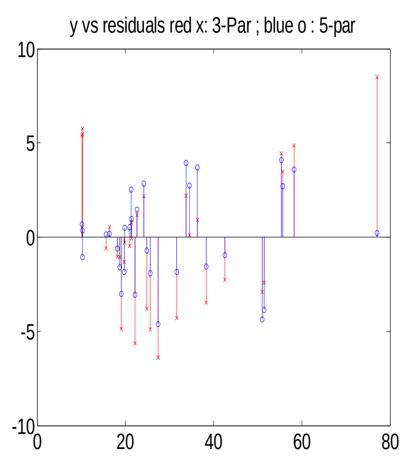
Dataset 7: The measured height, diameter and volume of 31 black cherry trees in Allegheny National Forest, Pennsylvania are analysed with MLR.

Phase 1:

```
|||||||||||||||||  statistics of regress parameters
     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
         a,          sda,        standErra,       standa,      ta
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

      -57.988       33.532       8.6382       -500.91       -1.7293
D      4.7082       1.0258      0.26426        1.2442       4.5896
H 0.33925      0.505230.13015      0.044154    0.67149

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```




```
|||||||||||||||||  statistics of regress parameters
     ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
         a,          sda,        standErra,       standa,      ta
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
       65.567       332.32        124.72        8177.3        0.1973
D      -21.464       13.495        5.0647       -108.71       -1.5904
H      -1.7574        24.95        9.3635       -16.455     -0.070436
D*log(D)7.2037         3.71        1.3943        10.044         1.939
D*log(D)0.40494        4.69        1.7621       0.71354      0.086244

            ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
8.3           70          10.3        5.4623       -1.0626
          8.6          65          10.3        5.7461       0.36894
          8.8          63          10.2         5.383       0.66745
         10.5          72          16.4       0.52588       0.18685
         10.7          81          18.8        -1.069       -1.5956
         10.8          83          19.7       -1.3183       -1.8445
           11          66          15.6      -0.59269       0.13503
           11          75          18.2       -1.0459      -0.60002
         11.1          80          22.6         1.187        1.4498
         11.2          75          19.9      -0.28758       0.48418
         11.3          79          24.2        2.1846        2.8406
         11.4          76            21      -0.46846       0.54408
         11.4          76          21.4     -0.068465       0.94408
         11.7          69          21.3       0.79385        2.5086
           12          75          19.1       -4.8541       -3.0319
         12.9          74          22.2       -5.6522       -3.0526
         12.9          85          33.8         2.216        3.9366
         13.3          86          27.4       -6.4065       -4.6212
         13.7          71          25.7        -4.901       -1.9107
         13.8          64          24.9        -3.797      -0.70148
           14          78          34.5       0.11182        2.7361
         14.2          80          31.7       -4.3083       -1.8582
         14.5          74          36.3       0.91474        3.7018
           16          72          38.3        -3.469       -1.5749
         16.3          77          42.6       -2.2777       -0.9745
         17.3          81          55.4        4.4571        4.0944
         17.5          82          55.7        3.4762        2.7014
         17.9          80          58.3        4.8715        3.5794
           18          80          51.5       -2.3993       -3.8748
           18          80            51       -2.8993       -4.3748
         20.6          87            77        8.4847       0.19849
```

```
Angles between column
vectors

        c1     c2     c3
--------------------------
c1     0.00
c2     11.39  0.00
c3     15.83  25.63  0.00
        c1     c2     c3
--------------------------

correlation matrix
        x1     x2     x3
--------------------------
x1     1.00
x2     0.52   1.00
x3     0.97   0.60   1.00
        x1     x2     x3
--------------------------
svd(x)
ans =
       431.04
       14.736
```

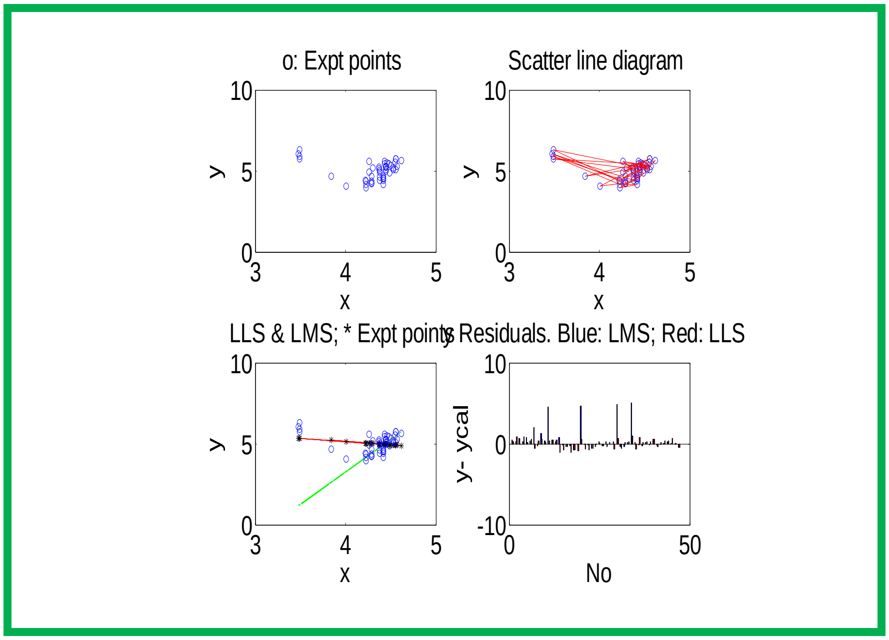
Dataset SI-8:

```
||||||||||||||||  statistics of regress parameters

       ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
a,          sda,         standErra,       standa,      ta

  6.7988      1.2353          2.19        14.889        5.504
-0.41474     0.28597       0.50698      -0.21027      -1.4503

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
sdy_LMS :1.5087 sdy_LLS : 0.56405
```



#	diagHat,StandRes,MD, studendizedResid			
1	0.022202	0.042564	0.43189	-0.43676
2	0.037341	0.73896	1.4758	-1.5041
3	0.021919	0.029558	-0.18086	0.18287
4	0.037341	0.73896	1.4758	-1.5041
5	0.021302	0.0011823	0.3095	-0.31285
6	0.02706	0.26603	0.90583	-0.91834
7	0.078054	2.6118	-0.98609	1.027
8	0.038652	0.79926	0.64986	-0.6628
9	0.021919	0.029558	0.95378	-0.96441
10	0.022202	0.042564	0.23687	-0.23954
11	0.1941	7.95	0.67127	-0.74775
12	0.024978	0.17026	0.86604	-0.87706
13	0.028705	0.3417	0.84962	-0.86208
14	0.044409	1.0641	-1.9248	1.969
15	0.021379	0.0047293	-1.3466	1.3613
16	0.024387	0.14306	-0.68372	0.69221
17	0.022922	0.07567	-1.9581	1.9809
18	0.024387	0.14306	-1.3929	1.4102
19	0.022922	0.07567	-1.5326	1.5504
20	0.1941	7.95	0.95493	-1.0637
21	0.021379	0.0047293	-1.1339	1.1462
22	0.021379	0.0047293	-1.4175	1.4329
23	0.024387	0.14306	-0.96738	0.9794
24	0.029604	0.38308	-0.15357	0.1559
25	0.022536	0.057935	0.066936	-0.067703
26	0.024387	0.14306	-0.54189	0.54862
27	0.021379	0.0047293	-0.63748	0.6444
28	0.022536	0.057935	-0.14581	0.14748
29	0.023359	0.095769	-1.1676	1.1815
30	0.19834	8.1451	1.2312	-1.3751
31	0.022536	0.057935	-0.99679	1.0082
32	0.037341	0.73896	0.34112	-0.34767
33	0.026314	0.23174	0.47298	-0.47933
34	0.1941	7.95	1.6641	-1.8537
35	0.022922	0.07567	-1.2489	1.2635
36	0.045977	1.1362	1.3071	-1.3383
37	0.033717	0.57225	0.31906	-0.32458
38	0.026314	0.23174	0.47298	-0.47933

39	0.033717	0.57225	0.46089	-0.46886
40	0.024978	0.17026	1.0788	-1.0925
41	0.022536	0.057935	-0.64222	0.64958
42	0.026314	0.23174	0.18932	-0.19186
43	0.030555	0.42682	0.72249	-0.73379
44	0.026314	0.23174	0.61481	-0.62306
45	0.036082	0.68103	1.1138	-1.1345
46	0.026314	0.23174	0.047491	-0.048128
47	0.024387	0.14306	-0.82555	0.8358

k	x	y	res_LMS	res_LLS
1	4.37	5.23	0.49	0.24361
2	4.56	5.74	0.24	0.83241
3	4.26	4.93	0.63	-0.10201
4	4.56	5.74	0.24	0.83241
5	4.3	5.19	0.73	0.17458
6	4.46	5.46	0.36	0.51093
7	3.84	4.65	2.03	-0.55621
8	4.57	5.27	-0.27	0.36656
9	4.26	5.57	1.27	0.53799
10	4.37	5.12	0.38	0.13361
11	3.49	5.73	4.51	0.37863
12	4.43	5.45	0.47	0.48849
13	4.48	5.42	0.24	0.47923
14	4.01	4.05	0.75	-1.0857
15	4.29	4.26	-0.16	-0.75957
16	4.42	4.58	-0.36	-0.38565
17	4.23	3.94	-0.24	-1.1045
18	4.42	4.18	-0.76	-0.78565
19	4.23	4.18	0	-0.86446
20	3.49	5.89	4.67	0.53863
21	4.29	4.38	-0.04	-0.63957
22	4.29	4.22	-0.2	-0.79957
23	4.42	4.42	-0.52	-0.54565
24	4.49	4.85	-0.37	-0.086623
25	4.38	5.02	0.24	0.037755
26	4.42	4.66	-0.28	-0.30565
27	4.29	4.66	0.24	-0.35957
28	4.38	4.9	0.12	-0.082245
29	4.22	4.39	0.25	-0.6586
30	3.48	6.05	4.87	0.69449
31	4.38	4.42	-0.36	-0.56224
32	4.56	5.1	-0.4	0.19241
33	4.45	5.22	0.16	0.26679
34	3.49	6.29	5.07	0.93863
35	4.23	4.34	0.16	-0.70446
36	4.62	5.62	-0.12	0.73729
37	4.53	5.1	-0.28	0.17997
38	4.45	5.22	0.16	0.26679
39	4.53	5.18	-0.2	0.25997
40	4.43	5.57	0.59	0.60849
41	4.38	4.62	-0.16	-0.36224
42	4.45	5.06	-2.6645e-15	0.10679
43	4.5	5.34	0.08	0.40752
44	4.45	5.3	0.24	0.34679
45	4.55	5.54	0.08	0.62826
46	4.45	4.98	-0.08	0.026787
47	4.42	4.5	-0.44	-0.46565

ACKNOWLEDGEMENTS

We thank for the editorial corrections of manuscript and also to suggest this review in 'CTLab' series.

During my Ph. D program, I (RSR) tried Facit calculator (adding machine) to calculate secondary formation function values of proton-ligand and metal ligand complexes instead of using four figure logarithms; but found it not a good approach. I went for Casio FX-8 calculator (costing 400 rupees around 1974-1975), which does not have provision for logarithm function against today's Casio Scientific Graphic Calculator FX CG20 (with 2900 functions for 10K INR). I calculated slope and intercept of dozens of data sets to arrive at stability constants of proton-ligand and metal ligand complexes (as simple as ML,ML2) spending long hours of clock time for several days with the help of co-researcher (K V Bapanaiah). This is preceded by drawing graphs, jotting down interpolated values etc. In a nut shell, calculation even for a simple system required more time than performing experiment with 20 data points. Around the year 1977, A. Satyanarayana, in the research school of our teacher P V Krishna Rao, asked me to help in computerizing calculations on IBM 1130 computer with punch card and line printer available on our campus. DR A Sitapathi of applied mathematics who promised to be with us left to Nuzvid on promotion. With several ups and downs, we continued FORTRAN-IV and developed several number crunching programs for in house use. Then the venture to use state-of-art software (SCOGS, POT-3, and MINIQUAD-74) in complex equilibria changed the facet our computational approach and in turn experimental plans. This continued over the last three decades using MINIQUAD-75, SUPERQUAD and HYPERQUAD, Hyss and our software Simulation of pH metric data (SoPhD), GHS, CEES, SiteCon etc. In 1978, I brought Randu program running on DEC 10, available at I I Sc. from Bangalore for random number generation to simulate noise. But, we could not implement here as it is requires some machine dependent modules. In Pune, we had the opportunity of visiting to PDP-11, mini computer and other high end hardware. But, we have no choice than to continue with Fortran_IV and IBM 1130 until 1985, when ICIM and OMC computers were procured by our university. At about the same time, a single piece of IBM compatible PC with GWbasic was available for users. An expert system for acido basic equilibria was developed in GWBASIC with heuristic rules and GAUSS_NEWTON numerical optimization technique. I tried promoting calculations in chemistry with TI-66 programmable calculator, a gift from USA, for a brief period around 1986. I had the opportunity of using PC (8086 + 8087 coprocessor) with 10MB hard disc by 1987 to develop CEES expert system in TURBOPROLOG-2, an AI language along with tiny user interfaces in BASICA and GWBasic. In 1989, with financial assistance to conduct international workshop on 'expert systems and numerical methods in chemistry', I purchased a PC with two floppy drives (one 1.2MB and other 640KB) for my laboratory. In 1990, I used Macintosh computer in Prof Braibanti's lab in Univ of Parma, Italy. I was thrilled to transfer data on floppies between two different hardware machines viz. Apple Macintosh and IBM. Also, I performed equilibrium calculations on a super computer at Bologna, 100 km from Univ of Parma through a terminal. On return, S V V Satyanarayana, my former research student gave me IBM-PC (386) with windows operating system to switch over from chiwriter to WORD. This was a requirementfor manuscript preparation to send to Prof. Braibanti. We continued publishing with changing standards of Word and hardware. In 1999, our lab got Pentium-2 machine, internet with dial up modem in a major project on 'predictive modeling in fisheries with neural networks'. From 2006, I started using laptops with dual processors, quad and eight processors. In 2012, dell system with i7 processor and a terabyte hard disc backup memory was purchased and multiple processing became routine. Now, we are going in for a 6^{th} generation i7 system. The development and application of regression analysis in our laboratory for research and pedagogy over these four decades closely followed our house programs and trend in hardware, commercial/ academic software and most importantly the advances in chemometric/ technometric research.

REFERENCES

Monographs in Regression analysis

[1] S.Puntanen, G. P. H. Styan, Chapter 52: Random Vectors and Linear Statistical Models, **2013**.
[2] S. Puntanen, G. A. F. Seber, G. P. H. Styan,Chapter 53: Multivariate Statistical Analysis,**2013**.
[3] S. Chatterjee, A. S. Hadi, Regression Analysis by Example, 5th Edition. Wiley, **2012**.
[4] G. G. Vining, A. P. Elizabeth, Douglas C. Montgomery,Introduction to Linear Regression Analysis, 5th Edition, John Wiley & Sons, **2012**.
[5] S.Puntanen, G. P. H. Styan, J. Isotalo, Matrix Tricks for Linear Statistical Models: Our Personal Top Twenty, Springer, **2011**.
[6] R. Dennis Cook, S. Weisberg, Applied Regression Including Computing and Graphics, John Wiley & Sons, **2009**.
[7] D. S. Bernstein, Matrix Mathematics: Theory, Facts, and Formulas**,** PrincetonUniversityPress, 2009.
[8] G. A. F. Seber, A Matrix Handbook for Statisticians. Wiley, **2008.**
[9] G. A. F. Seber, A. J. Lee, Linear Regression Analysis, 2nd Edition. Wiley. **2006**.
[10] S, Weisberg, Applied Linear Regression, 3rd Edition. Wiley. **2005**.
[11] K. M. Abadir, J. R. Magnus, Matrix Algebra, Cambridge University Press, **2005**.
[12] S. M. Stigler, Statistics on the Table: The History of Statistical Concepts and Methods, Harvard University Press, **1999**.
[13] N. R. Draper, H. Smith, Applied Regression Analysis, 3rd Edition. Wiley, **1998**, (Published Online:27 AUG 2014**)**.
[14] S.N. Deming, Y. Michotte, D.L. Massart, L. Kaufman, B.G.M. Vandeginste, Chemometrics: A Textbook, Elsevier, 1988**.**
[15] M. A. Sharaf, D. L. Illman, B. R. Kowalski, Chemometrics, John Wiley & Sons, **1986**.
[16] R. Sambasiva Rao and G Nageswara Rao,Computer applications in Chemistry, Himalya Publisher, New Delhi (India), **2005.**

E-man

[17] K RamaKrishna and R Sambasiva Rao, *J. Applicable Chem*., 2015, 4(6): 1597-1690.
Evolution of Mimics of Algorithms of Nature (E-man) Part 6: Research Tutorial on bat and Mosquito algorithms.
[18] K. RamaKrishna, G. Ramkumar and R. Sambasiva Rao, *J. Applicable Chem.*, **2013,** 2, 6, 1413-1458.
Evolution of Mimics of Algorithms of Nature (E-man) Part 5: Tutorial on Big_Bang–Big_Crunch algorithm.
[19] K. RamaKrishna, Ch. V. Kameswara Rao and R. Sambasiva Rao, *J. Applicable Chem.*, **2013,** 2, 5, 1007-1034.
E-man Part 4: Tutorial on prospects of charged system search (CSS) algorithm in chemical sciences.
[20] K. RamaKrishna, Ch. V. Kameswara Rao, R. Sambasiva Rao, *J. Applicable Chem.*, **2013,** 2, 4, 698-713.
Eman-Part III: Tutorial on gravitational algorithm in Structure activity relationships (SXR).
[21] K. Viswanath, R. Sambasiva Rao, Ch. V. Kameswara Rao, K. Rama Krishna, B. Rama Krishna and G. E. G. Santhosh, *J. Applicable Chem.*, **2012,** 1, *1*, 109-124.
Eman (Evolution of Mimics of Algorithms of Nature)-Part II: Application of neural networks for classification of bauxite.

Swarm Intelligence

[22] K. RamaKrishna, R. Sambasiva Rao,*J. Applicable Chem*., 2014, 3 (2), 449-492.
Swarm_Intelligence (SI)-State-of-Art (SI-SA), Part I: Tutorial on Firefly algorithm

Neural Networks

[23] K RamaKrishna, Ch. V. Kameswara Rao, V.Anantha Ramam, R. Sambasiva Rao, *J. Applicable Chem.*, **2014**, 3 (6), 2209-2311.

Mathematical Neural Network (MaNN) Models, Part VI: Single-layer perceptron [SLP] and Multi-layer perceptron [MLP] Neural networks in ChEM- Lab.

[24] M Venkata Subba Rao, V Ananta Ramam, V Muralidhara Rao, R Sambasiva Rao, *Asian J Chem*., **2010**, 22, 5937-5950.
Neural network modelling used as a chemometric tool for kinetic investigations

[25] I.Suryanarayana, A Braibanti, R Sambasiva Rao, V Ananta Ramam, D Sudarsan, G Nageswara Rao, *Fisheries Research*,**2008**, 92, 115-139.
Neural Networks in Fisheries Research

Regression

[26] Antonio Braibanti, Nageswara Rao Gollapalli, S. B. Jonnalagadda, D.Sudarsan, R.Sambasiva Rao,*Ann. Chim.* (Rome) **2001**, 91, 29-39.
Envirometrics Part I: Modeling of water salinity and air quality data

[27] Laila H Abdel Rehman and K Sambasiva Rao, G. Nageswara Rao, R. Sambasiva Rao, *J. Ind. Council. Chemists*, **2001**, 17, 33-50.
The State of the art of Chemometrics

[28] G. Nageswara Rao, V Ananta Ramam, S. Satyanarayana Rao and R. Sambasiva Rao, *J. Ind. Chem. Soc*., **1998**,75, 236-47.
Kinetometrics Part I: KINTOB -- A tool box for kinetics

[29] K Sambasiva Rao, R. Sambasiva Rao, *J Korean Phys. Soc*., **1998**, 32, S1850-S1851.
Fuzzy artificial neural networks models in preparation of ceramics

[30] V Anantha Ramam, G. Nageswara Rao, S V Rama Sastri and R. Sambasiva Rao,*Ind J Chem*. **1997**, 36A,964-969.
Kinetometrics : II The role of residual analysis in the estimation of rate constants

[31] G. Nageswara Rao, V Anantha Ramam and R. Sambasiva Rao,*Bull. Soc. Kinet. Ind*., **1997**, 19(2),1-6.
KINTOB: Kinetic tool box

[32] A. Braibanti, K Sambasiva Rao, Ch S V Prasad and R. Sambasiva Rao, Samyak, *J Chem*., **1997**, 1(1),17-21.
Multi-dimensional graphics in chemistry I. Applications of three dimensional surfaces and contour diagrams

[33] S.V.V. Satyanarayana Rao, J.S.V.M. Lingeswara Rao, A. Ravindra Babu, D. Murali Krishna and R. Sambasiva Rao, *J. Ind. Chem. Soc*. **1996**,73,9-19.
Chemometric Investigation of Complex Equilibria in Solution phase: V Correlation of Formation Constants of Mn(II) or Zn(II) Complexes of AADH/FAH with Co-solvent Characteristics

[34] A.Ravindra Babu, J.S.V.M Lingeswara Rao, D. Murali Krishna and R. Sambasiva Rao, *Anal. Chim. Acta*. **1995**, 306, 297-300.
Acid-base equilibria of 2-Furoic acid hydrazide and Adipic acid hydrazide in aqueous-organic Media

[35] J.S.V.M Lingeswara Rao, N. Satyanarayana, D. Muralikrishna, G.Nageswara Rao and R. Sambasiva Rao, *Chem. & Edu*. **1994**, 26-33.
Chemical Applications of Linear Kalman Filter

[36] A.Braibanti, E.Fisicaro, F. Dallavalle, J.D. Lamb, J.L.Oscarson, R. Sambasiva Rao, *J. Phys. Chem*. **1994**, 98,626-634.
Molecular Thermodynamic Model for the Solubility of Noble gases in water

[37] A.Ravindra Babu, D.Murali Krishna and R. Sambasiva Rao, *Ind.J.Chem*.,**1993**, 32A,1064-1071.
Chemometric investigation of complex equilibria in solution phase: part IV - Solute-solvent interactions in the complexation of adipic acid dihyrazide and 2-furoic acid hydrazide with Co(II) and Cu(II) in aquo- NN' dimethylformamide and aquo-dioxane media.

[38] A.Ravindra Babu, R. Sambasiva Rao, *J.Chem. & Eng*. Data **1993**, 37, 526-531.

Chemometric Investigation of Complex equilibria in solution phase III. Chemical models for the complexation of Ni(II) with AADH & FAH in aquo-DMF & aquo-DOX media-Correlation with solvent parameters.

[39] G.Nageswara Rao, S.V.V.Satyanarayana Rao, A.Ravindra Babu, R. Sambasiva Rao, *Asian J.Chem*.,**1992**,4,99.
Computer Oriented Mathematical Models for chemical systems

[40] G.Nageswara Rao, K.V. Ramana and R. Sambasiva Rao, *J. Ind.Chem.Soc*., **1991**,68, 34.
Computer augmented modelling of complexes of amino acids in aquo-organic mixtures I Acido-basic equilibria of L-alanine and L-Dopa in aquo-DMSO media

[41] P.V. Krishna Rao, R. Sambasiva Rao and Ch. Rambabu,*Ind. Chem. J.,***1978**, April, 21.
Spectrophotometric determination of microgram quantities of nicotinoyl hydrazine

[42] P.V. Krishna Rao, R. Sambasiva Rao and Ch. Rambabu, *Acta. Ciencia Indica*, **1978**, 4, 13.
Spectrophotometric determinations of aroyl hydrazines using 2,3,5-triphenyl tetrazolium chloride

[43] P.V. Krishna Rao and R. Sambasiva Rao, *Ind. Chem. J*., **1977**, 1.
A new spectrophotometric method for the determination of osmium using nicotinoyl and benzoyl hydrazines

[44] P.V. Krishna Rao R. Sambasiva Rao, *Curr. Sci*., **1975**, 44, 551.
A new photometric method for the determination of tervalent gold with nicotinic and benzoic acid hydrazides

LMS, LTS, LAD regressions

[45] X. Xia, Z. Liu, Hu Yang, *Computational Statistics & Data Analysis,* **2016,** 96, 104 – 119.
Regularized estimation for the least absolute relative error models with a diverging number of covariates

[46] M. Roozbeh, *Journal of Multivariate Analysis,* **2016,** 147, 127 – 144.
Robust ridge estimator in restricted semiparametric regression models

[47] M. Sugiyama, Chapter 25 - Robust Regression, *Introduction to Statistical Machine Learning, Morgan Kaufmann,* **2016,** 279 – 294.

[48] L. Huang, J. Zhao, H. Wang, S. Wang, *Computational Statistics & Data Analysis,* **2016,** 103, 384 – 400.
Robust shrinkage estimation and selection for functional multiple linear model through LAD loss

[49] J. Li, W. Zeng, J. Xie, Q. Yin, *Engineering Applications of Artificial Intelligence,* **2016,** 52, 54 – 64.
A new fuzzy regression model based on least absolute deviation

[50] B. Pan, Min Chen, Yan Wang, Wei Xia, *Journal of the Korean Statistical Society*,**2015,** 44, 1-11,
Weighted least absolute deviations estimation for ARFIMA time series with finite or infinite variance

[51] Z. Yan, Y. Yao, *Chemometrics and Intelligent Laboratory Systems,* **2015,** 146, 136 – 146.
Variable selection method for fault isolation using least absolute shrinkage and selection operator (LASSO)

[52] N. H. Chan, C. Yip Yau, Rong-Mao Zhang, *Journal of Econometrics,* **2015,** 189, 285 – 296.
"LASSO estimation of threshold autoregressive models

[53] M. Chen, Ke Zhu, *Journal of Econometrics,* **2015,** 189, 313 – 320.
Sign-based portmanteau test for ARCH-type models with heavy-tailed innovations

[54] N. Tengtrairat, W.L. Woo, *Neurocomputing,* **2015,** 147, 412 – 425.
Single-channel separation using underdetermined blind autoregressive model and least absolute deviation

[55] Thomas Haschka, Eric Hénon, Christophe Jaillet, Laurent Martiny, Catherine Etchebest, Manuel Dauchez, *Computational and Theoretical Chemistry*, **2015,** 1074, 50 **-** 57.
Direct minimization: Alternative to the traditional norm to derive partial atomic charges

[56] S. Kwon, S. Lee, Y. Kim, *Computational Statistics & Data Analysis,* **2015,** 92, 53 – 67.
Moderately clipped LASSO

[57] E. Daghir-Wojtkowiak, P. Wiczling, S. Bocian, Ł. Kubik, P. Kośliński, B. Buszewski, R. Kaliszan, M. Jan Markuszewski, *Journal of Chromatography A,* **2015,** 1403, 54 – 62.
Least absolute shrinkage and selection operator and dimensionality reduction techniques in quantitative structure retention relationship modeling of retention in hydrophilic interaction liquid chromatography

[58] Zakariya Yahya Algamal, Muhammad Hisyam Lee, *Expert Systems with Applications*, **2015,** 42, 9326 - 9332,
Penalized logistic regression with the adaptive LASSO for gene selection in high-dimensional cancer classification

[59] N. Ternès, F. Rotolo, G. Heinze, S. Michiels, *Revue d'Épidémiologie et de Santé Publique,* ***2015,*** *63, Supplement 2, S58,*
A new algorithm for nonlinear L1-norm minimization with nonlinear equality constraints

[60] J. Shi, Kun Chen, W. Song, *Statistics & Probability Letters*, **2014,** 84, 113 – 120.
Robust errors-in-variables linear regression via Laplace distribution

[61] Z. Zhou, Z. Lin, *Statistics & Probability Letters*, **2014,** 90, 25 – 32.
Asymptotic theory for LAD estimation of moderate deviations from a unit root

[62] R. Wu, *Statistics & Probability Letters*,**2014,** 94, 69 – 76.
Least absolute deviation estimation for general fractionally integrated autoregressive moving average time series models

[63] Hu Yang, J. Yang, *Journal of Statistical Planning and Inference*, **2014,** 151–152, 73 – 89.
The adaptive L1-penalized LAD regression for partially linear single-index models

[64] J. Yan, F. Wang, X. Cao, Jun Zhang, *Image and Vision Computing*, **2014,** 32, 930 – 939.
Robust object tracking using least absolute deviation

[65] Lie Wang, *Journal of Multivariate Analysis*, **2013,** 120, 135 – 151.
The penalized LAD estimator for high dimensional linear regression

[66] T. Honda, *Journal of Multivariate Analysis,* **2013,** 117, 150 – 162.
Nonparametric LAD cointegrating regression

[67] R. Grbić, K. Scitovski, K. Sabo, R. Scitovski, *Applied Mathematics and Computation,* **2013,** 219, 4387 – 4399.
Approximating surfaces by the moving least absolute deviations method

[68] Pao-sheng Shen, *Journal of the Korean Statistical Society,***2013,** 42, 469 – 479.
Median regression model with left truncated and interval-censored data

[69] P. Teppola, V.M. Taavitsainen, *Analytica Chimica Acta,***2013,** 768, 57 – 68.
Parsimonious and robust multivariate calibration with rational function Least Absolute Shrinkage and Selection Operator and rational function Elastic Net

[70] C. Colombani, A. Legarra, S. Fritz, F. Guillaume, P. Croiseau, V. Ducrocq, C. Robert-Granié, *Journal of Dairy Science*, **2013,** 96, 575 – 591.
Application of Bayesian least absolute shrinkage and selection operator (LASSO) and BayesCπ methods for genomic selection in French Holstein and Montbéliarde breeds

[71] M. Kelkinnama, S.M. Taheri, *Information Sciences*, **2012,** 214, 105 – 120.
Fuzzy least-absolutes regression using shape preserving operations

[72] Eduardo L. T. Conceição, António A. T. G. Portugal, *Ind. Eng. Chem. Res.* **2012,** *51* (3), 1118–1130,
Comparison of Two Robust Alternatives to the Box–Draper Determinant Criterion in Multiresponse Kinetic Parameter Estimation

[73] Pao-sheng Shen, *Journal of Statistical Planning and Inference,* **2012,** 142, 1757 – 1766.
Median regression model with left truncated and right censored data

[74] O. Arslan, *Computational Statistics & Data Analysis,***2012,** 56, 1952 – 1965.
Weighted LAD-LASSO method for robust parameter estimation and variable selection in regression

[75] M.D. Dyar, M.L. Carmosino, E.A. Breves, M.V. Ozanne, S.M. Clegg, R.C. Wiens, *Spectrochimica Acta Part B: Atomic Spectroscopy*, **2012,** 70, *51-67*.
Comparison of partial least squares and lasso regression techniques as applied to laser-induced breakdown spectroscopy of geological samples
[76] Y. Ma, Li Li, X. Huang, S. Wang, *IFAC Proceedings,* **2011,** 44, 11208 – 11213.
Robust Support Vector Machine Using Least Median Loss Penalty
[77] S. Hosseinian, S. Morgenthaler, *Journal of Statistical Planning and Inference,* **2011,** 141, 1497 – 1509.
Robust binary regression
[78] M. englong Yang, Y. Liu, Z. You, *Neurocomputing*,**2011,** 74, 3638 – 3645.
Estimating the fundamental matrix based on least absolute deviation
[79] P. D'Urso, R. Massari, A. Santoro, *Information Sciences,* **2011,** 181, 4154 – 4174.
Robust fuzzy regression analysis
[80] Chi Wai Yu, B. Clarke, *Journal of Multivariate Analysis,* **2010,** 101, 1950 – 1958.
Asymptotics of Bayesian median loss estimation
[81] T. Qingguo, *Journal of Statistical Planning and Inference,***2010,** 140, 393 – 405.
Estimation in a semiparametric model with longitudinal data
[82] Yu-Shen Liu, K. Ramani, *Computer-Aided Design*,**2009,** 41, 293 – 305.
Robust principal axes determination for point-based shapes using least median of squares
[83] W.W. Cooper, *European Journal of Operational Research*, **2009,** 198, 361 – 362.
Origins and uses of linear programming methods for treating and regressions: Corrections and comments on Castillo et al. (2008)
[84] M. Jhun, I. Choi, *Computational Statistics & Data Analysis,* **2009,** 53, 4221 – 4227.
Bootstrapping least distance estimator in the multivariate regression model
[85] R. Cupec, R. Grbić, K. Sabo, R. Scitovski, *Applied Mathematics and Computation,***2009,** 215, 983 – 994.
Three points method for searching the best least absolute deviations plane
[86] J. Kumar, M.S. Shunmugam, *Precision Engineering,***2007,** 31, 102 – 113.
Fitting of robust reference surface based on least absolute deviations
[87] Sundarraman Subramanian, *Statistical Methodology,* **2007,** 4, 121 - 131,
Median regression analysis from data with left and right censored observations
[88] Donna L. Mohr, *Computational Statistics & Data Analysis,* **2007,** 51, 3955 - 3967.
Bayesian identification of clustered outliers in multiple regression
[89] Yixin Fang, Lincheng Zhao, *Journal of Statistical Planning and Inference,* **2006,** 136, 1302 - 1316.
Approximation to the distribution of LAD estimators for censored regression by random weighting method
[90] H. Barreto, D. Maharry, *Computational Statistics & Data Analysis*, **2006,** 50, 1391 – 1397.
Least median of squares and regression through the origin
[91] A. Dax, *Computational Statistics & Data Analysis,***2006,** 50, 40 – 60.
The solution of linear inequalities
[92] Jong-Wuu Wu, Wen-Chuan Lee, *Applied Mathematics and Computation,* **2006,** 175, 609 – 617.
Computational algorithm of least absolute deviation method for determining number of outliers under normality
[93] Sunil L. Kukreja, Johan Löfberg, Martin J. Brenner, *IFAC Proceedings Volumes,* **2006,** 39, 814 - 819.
A least absolute shrinkage and selection operator (lasso) for nonlinear system identification
[94] R.A. Jabr, *International Journal of Electrical Power & Energy Systems,* **2006,** 28, 86 - 92.
Power system state estimation using an iteratively reweighted least squares method for sequential L1-regression
[95] Tatyana I. Igumenova, ,rew L. Lee, A. Joshua W,, *Biochemistry*,**2005,** *44* (38), 12627–12639.
Backbone and Side Chain Dynamics of Mutant Calmodulin−Peptide Complexes

[96] George Michailidis, Jan De Leeuw, *Computational Statistics & Data Analysis*, **2005,** 48, 587 - 603. Homogeneity analysis using absolute deviations
[97] Mingren Shi, Mark A. Lukas, *Computational Statistics & Data Analysis*, **2005,** 48, 779 - 802. Sensitivity analysis of constrained linear L1 regression: perturbations to response and predictor variables
[98] L. Wang, J. Wang, *Journal of Multivariate Analysis*, **2004,** 89, 243 – 260. The limiting behavior of least absolute deviation estimators for threshold autoregressive models
[99] A. lex,ros Leontitsis, Jenny Pange, *Mathematics and Computers in Simulation*,**2004,** 64, 543 – 547. Statistical significance of the LMS regression
[100] Rui-Bo Sun, Bo-Cheng Wei, *Statistics & Probability Letters*, **2004,** 67, 97 - 110. On influence assessment for LAD regression
[101] T. Víšek, *Journal of Statistical Planning and Inference*,**2003,** 113, 79 – 111. The likelihood ratio method for testing changes in the parameters of double exponential observations
[102] A. Giloni, M. Padberg, *Mathematical and Computer Modelling*,***2002****, 35, 1043-1060*. Least trimmed squares regression, least median squares regression, and mathematical programming
[103] Jorge G. Adrover, Ricardo A. Maronna, Víctor J. Yohai, *Journal of Statistical Planning and Inference*,***2002****, 105, 363-375*. Relationships between maximum depth and projection regression estimates
[104] Probal Chaudhuri, *Journal of Statistical Planning and Inference*, **2000,** 91, 229 - 238. Asymptotic consistency of median regression trees
[105] D. M. Hawkins, D. Olive, *Computational Statistics & Data Analysis*, ***1999,*** 32, *119-134*. A new algorithm for nonlinear L1-norm minimization with nonlinear equality constraints
[106] Meng-Dawn Cheng, Terra M. Nash, Scott E. Kopetz, *Journal of Aerosol Science*, ***1999****, 30, 805-817*. Retrieval of aerosol optical thickness by means of the least-median-squares robust algorithm
[107] J.Bai, *Journal of Statistical Planning and Inference*,**1998,** 74, 103 – 134. Estimation of multiple-regime regressions with least absolutes deviation
[108] Han-Lin Li, *Computers & Operations Research*,**1998,** 25, 1137 – 1143. Solve least absolute value regression problems using modified goal programming techniques
[109] Youshen Xia, Jun Wang, *Neurocomputing*, **1998,** 19, 13-21. Neural networks for solving least absolute and related problems
[110] Lawrence Lessner, *Socio-Economic Planning Sciences*, **1998,** 32, 45 - 55. Estimating HIV incidence: An ill-posed problem
[111] L.A. Sarabia, M.C. Ortiz, X. Tomás, *Analytica Chimica Acta*, **1997,** 348, 11 – 18. Performance of the orthogonal least median sqares regression
[112] Y. Dodge, *Journal of Multivariate Analysis*, **1997,** 61, 144 – 158. Regression for Detecting Outliers in Response and Explanatory Variables
[113] Mia Hubert, Peter J. Rousseeuw, *Journal of Statistical Planning and Inference*, **1997,** 57, 153 - 163. Robust regression with both continuous and binary regressors
[114] S.A. Soliman, S. Persaud, K. El-Nagar, M.E. El-Hawary", *International Journal of Electrical Power & Energy Systems*, **1997,** 19, 209 – 216. Application of least absolute value parameter estimation based on linear programming to short-term load forecasting
[115] C. F. Olson, *Information Processing Letters*, **1997,** 63, 237 – 241. An approximation algorithm for least median of squares regression
[116] Matlab, 2016a, www.mathworks.com, TheMathworks, Inc, Natick, MA 01760-2098 Econometrics Toolbox, User's guide, (2316 pages); Statistics and Machine learning Toolbox, User's Guide (7916 pages).
[117] Richard William Farebrother, *Computational Statistics & Data Analysis*, **1997,** 24, 455 - 466.

The historical development of the linear minimax absolute residual estimation procedure 1786–1960

[118] László Tóthfalusi, László Endrényi, *International Journal of Bio-Medical Computing*, **1996,** 42, 181 - 190.
Algorithms for robust nonlinear regression with heteroscedastic errors

[119] M.F. Calitz, H. Rüther, *Journal of Photogrammetry and Remote Sensing*,**1996,** 51, 223 – 229.
Least absolute deviation (LAD) image matching

[120] T. E. Dielman, E. L. Rose, *Computational Statistics & Data Analysis*,**1995,** 20, 119 – 130.
A bootstrap approach to hypothesis testing in least absolute value regression

[121] C.L. Karr, B. Weck, D.L. Massart, P. Vankeerberghen, *Engineering Applications of Artificial Intelligence*, **1995,** 8, 177 - 189.
Least median squares curve fitting using a genetic algorithm

[122] M.C. Ortiz-Fernández, A. Herrero-Gutiérrez, *Chemometrics and Intelligent Laboratory Systems*, **1995,** 27, 231 - 243.
Regression by least median squares, a methodological contribution to titration analysis

[123] B. Walczak, D.L. Massart, *Chemometrics and Intelligent Laboratory Systems*, **1995,** 27, 41 - 54.
Robust principal components regression as a detection tool for outliers

[124] Douglas M. Hawkins, *Computational Statistics & Data Analysis*, **1995,** 19, 519 - 538.
Convergence of the feasible solution algorithm for least median of squares regression

[125] P. Vankeerberghen, J. Smeyers-Verbeke, R. Leardi, C.L. Karr, D.L. Massart, *Chemometrics and Intelligent Laboratory Systems*, **1995,** 28, 73 - 87.
Robust regression and outlier detection for non-linear models using genetic algorithms

[126] Amy Fisher, Paul S. Horn, *Computational Statistics & Data Analysis*, ***1994,** 17, 129-140*.
Robust prediction intervals in a regression setting

[127] Clint W. Coakley, Lamine Mili, Michael G. Cheniae, *Statistics & Probability Letters*, **1994,** 19, 399 - 408.
Effect of leverage on the finite sample efficiencies of high breakdown estimators

[128] D.F. Vecchia, J.D. Splett, *ISA Transactions*, **1994,** 33, 411 - 420.
Outlier-resistant methods for estimation and model fitting

[129] T. E. Dielman, E. L. Rose, *International Journal of Forecasting*,**1994,** 10, 539 – 547.
Forecasting in least absolute value regression with auto correlated errors: a small-sample study

[130] Peter J. Rousseeuw, Joachim Wagner, *Computational Statistics & Data Analysis*, **1994,** 17, 65 - 76.
Robust regression with a distributed intercept using least median of squares

[131] Subhash C. Narula, Pekka J. Korhonen, *European Journal of Operational Research*, **1994,** 73, 70 - 75.
Multivariate multiple linear regression based on the minimum sum of absolute errors criterion

[132] P.-T. Chang, E.S. Lee, *Computers & Mathematics with Applications*,**1994,** 28, 89 – 101.
Fuzzy least absolute deviations regression and the conflicting trends in fuzzy parameters

[133] D. V,ev, *Statistics & Probability Letters*, **1993,** 16, 117 - 119.
A note on the breakdown point of the least median of squares and least trimmed squares estimators

[134] T. Mathew, K. Nordström, *Statistics & Probability Letters*, **1993,** 16, 153 – 158.
Least squares and least absolute deviation procedures in approximately linear models

[135] W.T.M. Dunsmuir, B.A. Murtagh, *European Journal of Operational Research*, **1993,** 67, 272 – 277.
Least absolute deviation estimation of stationary time series models

[136] Chong-wei Xu, Wei-Kei Shiue, *Computational Statistics & Data Analysis*, **1993,** 16, 349 - 362.
Parallel algorithms for least median of squares regression

[137] Douglas M. Hawkins, *Computational Statistics & Data Analysis*, **1993,** 16, 81 - 101.
The feasible set algorithm for least median of squares regression

[138] M. G. Sklar, R. D. Armstrong, *Computers & Operations Research*, **1993,** 20, 83 – 93.
Lagrangian approach for large-scale least absolute value estimation

[139] A. Al-K,ari, S.A. Soliman, K. El-Naggar, *Electric Power Systems Research*, **1993,** 28, 99 - 104.
Digital dynamic identification of power system subharmonics based on the least absolute value
[140] G. E. Dallal, Peter J. Rousseeuw, *Computers and Biomedical Research*,**1992,** 25, 384 – 391.
LMSMVE: A program for least median of squares regression and robust distances
[141] C.Y. Chork ,*Journal of Geochemical Exploration*, **1991,** 41, 325-340.
An assessment of least median of squares regression in exploration geochemistry
[142] G.S. Christensen, S.A. Soliman, M.Y. Mohamed, *Analysis and Control System Techniques for Electric Power Systems, Part 4 of 4 Academic Press*, **1991,** 44(4), 345 – 487.
Power Systems State Estimation Based on Least Absolute Value (LAV)
[143] Y. Dodge, J. Antoch, Jana Jurečková, *Computational Statistics & Data Analysis*, **1991,** 12, 87 – 99.
Computational aspects of adaptive combination of least squares and least absolute deviations estimators
[144] G.S. Christensen, S.A. Soliman, *Electric Power Systems Research*, **1991,** 21, 91 – 98.
Least absolute value estimation of the generalized operational impendances of solid-rotor synchronous machines from standstill frequency response test data
[145] S.A. Soliman, G.S. Christensen, A.H. Rouhi, *Computational Statistics & Data Analysis, **1991,** 11, 97-109.*
A new algorithm for nonlinear L1-norm minimization with nonlinear equality constraints
[146] J. Joss, A. Marazzi, *Computational Statistics & Data Analysis*,**1990,** 9, 123 – 133.
Probabilistic algorithms for least median of squares regression
[147] S.A. Soliman, G.S. Christensen, S.S. Fouda, *Electric Power Systems Research*, **1990,** 19, 23 – 35.
On the application of the least absolute value parameter estimation algorithm to distance relaying
[148] Clifford M. Hurvich, Chih-Ling Tsai, *Statistics & Probability Letters*, **1990,** 9, 259 - 265.
Model selection for least absolute deviations regression in small samples
[149] Yuzhu Hu, J. Smeyers-Verbeke, D.L. Massart, *Chemometrics and Intelligent Laboratory Systems*, **1990,** 9, 31 - 44.
Outlier Detection in Calibration
[150] K.A. Clements, P.W. Davis, K.D. Frey, *Power Systems and Power Plant Control 1989*, **1990,** 371 - 375.
An efficient algorithm for computing the weighted least absolute value estimate in power system static state estimation
[151] G.S. Christensen, S.A. Soliman, *Automatica*, **1990,** 26, 389 - 395.
Optimal filtering of linear discrete dynamic systems based on least absolute value approximations
[152] I.B.C. Matheson, *Computers & Chemistry*,**1990,** , 49 – 57.
A critical comparison of least absolute deviation fitting (robust) and least squares fitting: The importance of error distributions
[153] S.A. Soliman, G.S. Christensen, D.H. Kelly, N. Liu, *Electric Power Systems Research*, **1990,** 19, 73 - 84.
An algorithm for frequency relaying based on least absolute value approximations
[154] B.C. Matheson, *Computers & Chemistry*, ***1989,*** 13, 299-304.
Robust estimation of parameters: A simple modification to all non-linear fitting algorithms to convert from minimizing the sum of squares of deviations to minimizing the sum of the absolute deviations
[155] S. C. Rutan, P. W. Carr, *Analytica Chimica Acta*, **1988,** 215, 131 – 142.
Comparison of robust regression methods based on least-median and adaptive kalman filtering approaches applied to linear calibration data
[156] James E. Gentle, V.A. Sposito, Subhash C. Narula, *Computational Statistics & Data Analysis*, **1988,** 6, 335 - 339.
Algorithms for unconstrained L1 simple linear regression

[157] S.A. Soliman, G.S. Christensen, A. Rouhi, *Computational Statistics & Data Analysis*, **1988,** 6, 341 - 351.
A new technique for curve fitting based on minimum absolute deviations
[158] A. van den Bos, *Automatica*, **1988,** 24, 803 - 808.
Nonlinear least-absolute-values and minimax model fitting
[159] J. W McKean, G. L Sievers, *Statistics & Probability Letters*, **1987,** 5, 49 – 54.
Coefficients of determination for least absolute deviation analysis
[160] A. Charnes, W.W. Cooper, T. Sueyoshi, *European Journal of Operational Research*, **1986,** 27, 146 - 157.
Least squares/ridge regression and goal programming/constrained regression alternatives
[161] Desire L. Massart, Leonard Kaufman, Peter J. Rousseeuw, Annick Leroy, *Analytica Chimica Acta*, **1986,** 187, 171 - 179.
Least median of squares: a robust method for outlier and model error detection in regression and calibration
[162] J.M. Steele, W.L. Steiger, *Discrete Applied Mathematics*, **1986,** 14, 93 - 100.
Algorithms and complexity for least median of squares regression
[163] James L Powell, *Journal of Econometrics*, **1984,** 25, 303 - 325.
Least absolute deviations estimation for the censored regression model
[164] R. Sambasiva Rao et al. (manuscript under preparation)

AppendixA0:Research Algorithms in Regression Evolution (Rare)

In 1992, we proposed a general program strategy for complex equilibria by pH metric data based on different object functions, calculation methods for equilibrium concentrations/stability constants, optimization procedures, statistical tests for validation etc. This strategy could emulate most of the programs in vogue viz. SCOGS-X, POT-3, MINIQUAD-X, SCPHD, SOPHD, ESAB, BEST etc. In computational quantum chemistry, G09 and Schrodinger suit adapt work flow concept for multi-purpose computations. Recently flow representation and execution ($$$-flow, $$$: [data, Method, algorithm, knowledge]) gained popularity in statistical packages and in discipline specific softwares. The work flow approach for regression follows here. It is not in a rigid framework, but with a target of stability/plasticity compromise, embedding detection of conflicts/updated-remedial solutions and evolving features for eventual integration with time.

ॐ Chart A0.1: Workflow of regression (Broad level)

Scatter diagram → Model → Method → Stats

ES → Noise injection → Ensemble → Conscientious Intelligent System advice **(CiSa)**

Workflow of linear regression(Method level) --

("low-hanging fruit" approach)

Scatter diagram
Data
Residuals

Model
Mean
Lin (x,y)
MLR
poly

Method
LS
LAD
LMS
LTS

Stats	
Reg Coe	Residy
	Standardize
SD	studentized

Noise injection
Influence
Monte carlo

ES
Parametrization
Curve fitting

Categories of regression

Regression	
Binary	
Bionomial	
Poisson	
Logistic	
Probit	
	ordinary
	Mutlinomial
	Ordered
Logit	
	Mutlinomial
	Ordered
LASSO	
Least Angle	
Symbolic	

Typical KBs for .RegMethods

if	numerical data & real
Then	Numerical regression
if	numerical data & binary
Then	Binary regression
if	y follows Poisson distribution
Then	Poisson regression
if	y is an outcome of two possible disjoint states (traditionally denoted "success" or 1, and "failure" or 0)
Then	binomial distribution

Regression Statistical
Linear in parameters
Ordinary
LAD
Multiple linear in X
Polynomial in x

Regression Statistical
Unit weighted
Weighted
Maximum Likelihood
Iterative

Insensitive to outliers

Homosedastic & Normal	Noise in y	UWLS
Heteroscedastic normal	Noise in y	WLS
Non-normal	Noise in y	MLE
Normal	Noise in x and y	
Fuzzy	Noise in y	Fuzzy reg

#Outliers	Outliers in
A few	y only
Cluster	xolnly
	Both in xand y

Regression -Fuzzy sets
Fuzzy regression
Envelop estimators

Robust

Ordinary LS (OLS)
Vertical (OLS)
Horizontal OLS
Bivariate LS
Geometric Mean Reg.
Orthogonal Reg.
Deming Reg.

ॐ Chart A0.2: State-of-knowledge-research modules of regression analysis

Data	
Input	Output
X	Y

Perturbation in
Y only
X only
Both in X and y

Relationship between X and Y
Known
Not known

Model
Model driven
Data driven

Model		
Functional relationship		
Linear	→	Variable
		Parameter
Non-linear		Both

Model
Function
Algebraic
Stochastic
Fuzzy

Model Function
Algebraic -Linear
Bilinear
Tri-linear
Quadri-linear

Relationship between X and Y	
If known	
Functional	
Model	
	Parametric
	Non-parametric
If not known	
Model free	Soft
Discovery models	
	Genetic programming
	Genetic expression

Data		
Primary		
	instrument	accuracy
Transformed		
		scaling
		functional
Derived		
		parameters
		Information/ KB

Data structure	
Logical	
	Binary
Categorical	
Nominal	
Multinomial	
Numeric	
	Real
	Imaginary
	Quaternion
Image	
	Pixel
	voxel

Data structure	
Univariate	
Bivariate	
Multivariate	
	X or Y
	X and Y
3-way	
4-way	
Multi-way	

Internal model of (Response/ Explanatory variables) Data		
If apriori knowledge of model		
Linear		
	Bilinear	
	Trilinear	
	Quadrilinear	
Nonlinear		
	Polynomial	
	exponential	Gaussian
	Periodic	
		trigonometric
Residual		
		Bilinear
		Non-bilinear
IF no apriori knowledge of model		
		GP with known operators, functions

Noise	
Deterministic	
Probabilistic	
	Stastistical
Fuzzy	

Distribution	Model	function
Normal	$\mu = X * par$	
Poisson	$\log(\mu) = X * par$	log
Binomial	$\log(\mu/(1-\mu)) = X * par$	logit
Probit	$\mathrm{Norm}(\mathrm{inv}(\mu)) = X * par$	probit
	$\log(-\log(1-\mu)) = X * par$	comploglog
Gamma	$1/\mu = X * par$	reciprocal
	$\log(-\log(\mu)) = X * par$	loglog
Inverse Gaussian (With P = -2)	$\mu^p = X * par$	p (integer)

Noise strucutre
Known
unknown

Local	Single	Constraints	objFn characteristics
Global	Multiple	No	
Pareto		= ; < ; >	

Chart A0.3: Expert system driven method flow of Reg2015

				Regression		
Scatter diagram	→	ANOVA (with model_set)	→	Phase I	→	Phase II
If 2D &3D-surfaces are linear, proceed Else non-linear analysis		✓ Model with min(ResidSumSq)		○ Outlier detection (LMS, LTS)		○ Model_set
				○ Remove outliers		✓ Model with min(SD_par & sdy)
Phase III						
Residual distribution	→	If normal	→	○ Detailed statistics- ○ Joint confidence limits of parameters/response		
		If normal & heteroscedastic		✓ Weighted least squares		
		If any other distribution		– WLS not suitable		

Chart A0.4: Reg2015 Road map

NC KB Choice of method Failure/break points Remedial measures	Formulas_ Algebraic Matrix Tensor object	MethodBase $$$2015 LLS2015 LAD2015 LMS2015 Univar2015 ANOVA2015	StepByStep_
Inform_	InpChk_	oo_	autotest

Chart A0.5: Output of model

Parameters	Parameter statistics Std_par Standardized_par t-values	ycal	Residuals in y residy sdy

Advanced statistics

Resid Stand Student Jacknife DFFIT	regcoef Conf interval cc	confycal conf	Distances - Residy Cook Mahalanobis
Influence stats	Tests Durbin-Watson	if Replicate observations Then Lack of fit & Pure error	

Advances in regression methods from intelligent computational evolution perspective

Although computational intelligence, nature inspired algorithms, artificial intelligence are sparkles recent time in regression methods, there is a natural evolution just need based in solving real life tasks which mirrors ICE.

The regression spread its wings into neural networks, support vector machines, nature inspired algorithm, fuzzy/rough sets and so on. Probabilistic NNs, regression NNs etc. were discussed in our earlier reviews on mathematical NNs (MNNs) [$$$]. The intelligent computational features of support vector-, Genetic-, fuzzy-, interval-, non-linear- regressions will be reported separately [$$$].

Appendix A1: Symbolic differentiation of matrices

In linear algebra, the product of vectors, matrices and /or their products have a key role. The rules of differentiation of algebraic and transcendental functions when applied to matrices, they are of not only extended interest, but derivations are simple and elegant. Top down and bottom up complexity (scalar to matrix through vectors) becomes trivial. In recent years, all most all engineering/applied sciences/commerce employ 3way-/4way tensors and datasets up to six-way are predominant/prevalent. The differential operators for tri- and quadri-linear cause-effect models and matlab with tensor algebra tool boxes have opened new computational jargon. The symbolic mathematical tool box mostly relieves the drudgery of expansions of polynomial equations, differentiation/integration etc. These will be described in a separate context ($$$)

In linear least squares task with two regression parameters (viz. slope and intercept [a1,a0]), the design matrix (X) is a rectangular one of size [NP x 2] and response (y) is a column vector (NP x 1). The partial derivatives of product of matrices of interest here with respect to parameters (a) are collected in table A1-1. The details of steps of expansion of products, differentiation and end result are demonstrated considering three data points and two LLS parameters to be estimated statistically.

Table A1-1: Partial derivatives of matrices with respect to vectors

$\dfrac{\partial\left[y^T * y\right]}{\partial a}$	$= 0$

$\frac{\partial\left[\mathrm{y}^{T} * X * a\right]}{\partial a}$	$= X^{T} * y$
$\frac{\partial\left[a^{T} * X^{T} * y\right]}{\partial a}$	$= X^{T} * y$
$\frac{\partial\left[a^{T} * X^{T} * X * a\right]}{\partial a}$	$= 2 * \left(X^{T} * X\right) * a$

Table A1-1b: Details of partial derivatives of matrices with respect to vectors with NP = 3

$\frac{\partial\left[\mathrm{y}^{T} * \mathrm{y}\right]}{\partial a}$	y is not a function of a; Thus, its derivative wrt to a is zero	$= 0$
$\frac{\partial\left[\mathrm{y}^{T} * X * a\right]}{\partial a}$	$= {}_{1}\begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}^{3} * {}_{3}\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix}^{2} * {}_{2}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix}^{1}$ $= {}_{1}\begin{bmatrix} y_1 + y_2 + y_3 & x_1 * y_1 + x_2 * y_2 + x_3 * y_3 \end{bmatrix}^{2} * {}_{2}\begin{bmatrix} a_0 \\ a_1 \end{bmatrix}^{1}$ $= {}_{1}\left[\, a_0 * \left(y_1 + y_2 + y_3\right) + a_1 * \left(x_1 * y_1 + x_2 * y_2 + x_3 * y_3\right)\right]^{1}$ $\frac{\partial\left[\mathrm{y}^{T} * X * a\right]}{\partial a_0} = \left(y_1 + y_2 + y_3\right) + 0$ $\frac{\partial\left[\mathrm{y}^{T} * X * a\right]}{\partial a_1} = 0 + \left(x_1 * y_1 + x_2 * y_2 + x_3 * y_3\right)$ $\frac{\partial\left[\mathrm{y}^{T} * X * a\right]}{\partial a} = \begin{bmatrix} \left(y_1 + y_2 + y_3\right) \\ \left(x_1 * y_1 + x_2 * y_2 + x_3 * y_3\right) \end{bmatrix} \quad = X^{T} * y$ $= \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \end{bmatrix} * \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$	$= X^{T} * y$

$\frac{\partial\left[a^T * X^T * y\right]}{\partial a}$		$= X^T * y$
$\frac{\partial\left[a^T * X^T * X * a\right]}{\partial a}$		$= 2*\left(X^T * X\right)*a$

Appendix A2: Derivation of Linear least squares (LLS) in matrix notation

Probability theory for estimation of regression parameters

The probability of observing a datum (yi) from normal distribution is described in chart A2-1.

Chart A2-1: probability application in regression

$$prob(y_i) = \frac{1}{\sigma * \sqrt{2*\Pi}} * e^{\frac{-(yi-\alpha-\beta*xi)^2}{2*\sigma^2}}$$

$$Let\, k = \frac{1}{\sigma * \sqrt{2*\Pi}}$$

$$prob(y_i) = k * e^{\frac{-(yi-\alpha-\beta*xi)^2}{2*\sigma^2}}$$

For instance in the case of points (1,2, and NP)

$$prob(y_1) = k * e^{\frac{-(y_1-\alpha-\beta*x_1)^2}{2*\sigma^2}}$$

$$prob(y_2) = k * e^{\frac{-(y_2-\alpha-\beta*x_2)^2}{2*\sigma^2}}$$

....

$$prob(y_{NP}) = k * e^{\frac{-(y_{NP}-\alpha-\beta*x_{NP})^2}{2*\sigma^2}}$$

The simultaneous occurrence of these probabilities happens when their product is considered

$$prob(y_i \,|\, i = 1:NP) =$$

$$\left[k * e^{\frac{-(y_1-\alpha-\beta*x_1)^2}{2*\sigma^2}}\right] * \left[k * e^{\frac{-(y_2-\alpha-\beta*x_2)^2}{2*\sigma^2}}\right] * \left[k * e^{\frac{-(y_{NP}-\alpha-\beta*x_{NP})^2}{2*\sigma^2}}\right]$$

$$= k^{NP} * e^{\frac{-\sum_{i=1}^{NP}(y_i-\alpha-\beta*x_i)^2}{2*\sigma^2}}$$

If prob(.) is maximum, the regression line is best representation of data points. It happens when $\alpha\, and\, \beta$ are such that sum of squares term $SSR = \sum_{i=1}^{NP}\left(y_i - \alpha - \beta * x_i\right)^2$ is minimum.

Parameters	
Population	Sample
α	a0
β	a1

If	$\min(SSR) = \min\left[\sum_{i=1}^{NP}\left(y_i - \alpha - \beta * x_i\right)^2\right]$ is achieved

μ	mean
σ	std

Then	$prob(y_i \mid i=1:NP)$ is maximum

Chart A2-1b: object functions and goals in regression

Method	norm	objFn			Goal
least-absolute-deviation (LAD)	l1	Sum(abs(devy))	$one^T * abs(resid) =$	$(one)^T * abs(y - X*a)$	Min(objFn1)
Least squares	l2	Sum(abs(devy.^2))	$resid^T * resid =$	$(y - X*a)^T * (y - X*a)$	Min(objFn2)
minimax	l∞				Minmax(objFn3)
least-deviation (LD)	…..	Sum(devy)	$one^T * resid =$	$(one)^T * (y - X*a)$	Min(objFn0)

α and β are called population parameters in statistical theory and applicable for a large number of measurements (→ infinity ideally; in realistic sense since a century NP>30, and recently million in rare experiments of CERN). Small samples correspond to (NP <30, but many times NP<20; many studies involved 4 to 10 data points with special modified statistics). In experimental studies sample parameters a0,a1 correspond to intercept and slope of straight line, mean and standard deviation to μ and σ .

Derivation A2 – 1: Least squares solution of $y = fn(X;a)$

$$a = par = \left(X^T * X\right)^{-1} * X^T * y$$

$$\text{SSResid} = resid^T * resid$$

$$= (y - X*a)^T * (y - X*a)$$

expanding RHS

$$= y^T * y - y^T * (X*a) - (X*a)^T * y - (X*a)^T * (X*a)$$

since, $(X*a)^T \xrightarrow{=} a^T * X^T$

$$= y^T * y - y^T * X*a - a^T * X^T * y - a^T * X^T * X*a$$

….. contd

$$\frac{\partial\, resid^T * resid}{\partial a} = 0 - X^T * y - X^T * y + 2*\left(X^T *\right.$$

$$= 0$$

$$\Rightarrow -2*X^T*y + 2*\left(X^T*X\right)*a = 0$$

$$\Rightarrow 2*\left(X^T*X\right)*a = 2*X^T*y$$

$$\Rightarrow \left(X^T*X\right)*a = X^T*y$$

Premultiplying by $\left(X^T*X\right)^{-1}$

$$\Rightarrow \left(X^T*X\right)^{-1} * \left(X^T*X\right)*a = \left(X^T*X\right)^{-1} * X$$

$$\Rightarrow I*a = \left(X^T*X\right)^{-1} * X^T*y$$

$$\Rightarrow a = \left(X^T*X\right)^{-1} * X^T*y$$

$$\Rightarrow a = \left(X^T*X\right)^{-1} * X^T*y$$

$$\frac{\partial\, resid^{T} * resid}{\partial a} = 0$$

$$= \frac{\partial\left[\mathrm{y}^{T} * \mathrm{y}\ \text{-}\ \mathrm{y}^{T} * X * a - a^{T} * X^{T} * y - a^{T} * X^{T} * X * a \right]}{\partial a} = 0$$

$$= \frac{\partial\left[\mathrm{y}^{T} * \mathrm{y} \right]}{\partial a} - \frac{\partial\left[\mathrm{y}^{T} * X * a \right]}{\partial a} - \frac{\partial\left[a^{T} * X^{T} * y \right]}{\partial a} - \frac{\partial\left[a^{T} * X^{T} * X * a \right]}{\partial a}$$

contd …..

– – – – –

SSResid: *Sum of squares of residuals*

Since $\left(X^{T} * X \right)$ is a square symmetric, many of methods available for inverse are applicable iff X is full rank. In linear least squares solution of a straight line, there is only one x variable and thus X is of rank 2.

Appendix A3: Design matrix

DesignMatrix for explanatory variables: The matlab program desmat2015.m outputs numerical vectors for given x-matrix of npar variables (colums). PolyModel.m has built set of models up to quartic and cross product terms. The outputs (chart A3-1) of autotest_desmat2015.m and X2015.m demonstrate typical models popular over half a century.

Chart A3-1: Different models generated for given x vectors

x = []	x = 1 2 3 4 5 6		x2 = 1 1 2 2 3 3 4 4 5 5 6 6
ModelPoly = 'one'	ModelPoly = '[one] ' '[lin] ' '[quad] ' '[cube] ' '[quartic] ' '[lin quad] ' '[lin cube] ' '[lin quartic] ' '[quad cube] ' '[quad quartic]' '[cube quartic]' '[lin quad cube] ' '[lin quad quartic]' '[quad cube quartic]' '[lin quad cube quartic]'		ModelPoly = '[one] ' '[lin] ' '[quad] ' '[cpb] ' '[lin quad] ' '[lin cpb] ' '[quad cpb] ' '[lin quad cpb]' Full quadratic

			x2 = 1 1 2 2 3 3 4 4 5 5 6 6	
			ModelPoly = '[one] ' '[lin] ' '[quad] ' '[cube] ' '[quartic] ' '[lin quad] ' '[lin cube] ' '[lin quartic] ' '[quad cube] ' '[quad quartic]' '[cube quartic]' '[lin quad cube] ' '[lin quad quartic]' '[quad cube quartic]' '[lin quad cube quartic]'	'[lin cpb] ' '[quad cpb] ' '[cube cpb] ' '[quartic cpb] ' '[lin quad cpb] ' '[lin cube cpb] ' '[lin quartic cpb] ' '[quad cubecpb] ' '[quad quartic cpb]' '[cube quartic cpb]' '[lin quad cube cpb] ' '[lin quad quartic cpb]' '[quad cube quartic cpb]' '[lin quad cube quartic cpb]'
	x3 = 1 1 1 2 2 2 3 3 3 4 4 4 5 5 5 6 6 6 '		x4 = 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6	
	ModelPoly = '[cpb cpt]		ModelPoly = '[cpb cpt cpq]	

MethodBase_Reg A3-1: Components of Design matrix 12-8-16	**desmat2015.m**

```
%
%                   desmat2015.m    (R S Rao) 26/9/93, 9/5/15
%                 Design matrix for Regression, Experimental Design
%
function [one,lin,quad,cube,quartic,cpb,cpt,cpq]  = desmat2015(f)
%
```

```
if nargin == 0
             clean
            disp(['[one,lin,quad,cube,quartic,cpb,cpt,cpq]  = desmat(f)'])
            f = [1 2 3; 2 3 4; 3 4 5;]
end

         %%
         lin = [];quad = [];cube= [];quartic =[];
         cpb = [];cpt = [];cpq = [];
%
         [rf,cf ] = size(f);
if rf ==0, one = [];end
         one = ones(rf,1);
if rf ==0, one = [];end
         %%
for i = 1:cf
%
%           linear, quadratic, cubic and quartic vectors
%
             lin = [lin, f(:,i)];
             quad = [quad,f(:,i).^2];
             cube = [cube,f(:,i).^3];
             quartic= [quartic, f(:,i).^4];
end
         %%
%
%              Cross product (cp) terms
%
for i = 1:cf
%_
% Binary (cpb)  if cf = 2
%
if cf > 1
for j = i+1:cf
                   cpb = [cpb, f(:,i).* f(:,j)];
end
end
%
%  ternary (cpt) if cf = 3
%
if cf > 2
for j = i+2:cf
                    cpt =  [cpt,f(:,i).* f(:,i+1).* f(:,j)];
end
end
%
%  quaternary (qpt) if cf = 4
%
if cf > 3
for j = i+3:cf
                    cpq =  [cpq,f(:,i).* f(:,i+1).* f(:,i+2).* f(:,j)];
end
end
end%  i loop

         %%
```

MethodBase_Reg A3-2: Components of Design matrix	**polyModels.m**

```
%
% polyModels.m   (R S Rao) 4/13/93, 10/27/1997,10/21/2011
```

```
%

function [ModelPoly] = polyModels(x)

if nargin == 0
    npar = 1;
else
    [np,npar]= size(x);
end
%

if npar == 0
   ModelPoly = {'one'};
end
if npar == 1
   ModelPoly = {'[one]           '%1
'[lin]           '%2
'[quad]          '%3
'[cube]          '%4
'[quartic]       '%5
%
'[lin quad]      '%6
'[lin cube ]     '%7
'[lin quartic ] '%8'
'[quad cube]     '%9
'[quad quartic]'%10
'[cube quartic]'%11
%
'[lin quad cube]   '%12
'[lin quad quartic]'%13
'[quad cube quartic]'%14
%
'[lin quad cube quartic]'%15
                };
end

if npar == 2
   ModelPoly = {'[one]           '%1
'[lin]           '%2
'[quad]          '%3
'[cube]          '%4
'[quartic]       '%5
%
'[lin quad]      '%6
'[lin cube ]     '%7
'[lin quartic ] '%8'
'[quad cube]     '%9
'[quad quartic]'%10
'[cube quartic]'%11
%
'[lin quad cube]   '%12
'[lin quad quartic]'%13
'[quad cube quartic]'%14
%
'[lin quad cube quartic]'%15
%
'[lin cpb]           '%16
'[quad cpb]          '%17
'[cube cpb]          '%18
'[quartic cpb]       '%19
%
'[lin quad cpb]      '%20
'[lin cube cpb]    '%21
```

```
'[lin quartic cpb] '%22
'[quad cubecpb]    '%23
'[quad quartic cpb]'%24
'[cube quartic cpb]'%25
%
'[lin quad cube cpb]    '%27
'[lin quad quartic cpb]'%28
'[quad cube quartic cpb]'%29
%
'[lin quad cube quartic cpb]'%30


                    };
end

if npar == 3
%
    ModelPoly =             { '[cpb cpt]      '} ; %36
end

if npar == 4
%
    ModelPoly =             { '[cpb cpt cpq]      '} ; %36
end
```

```
Data(x,y) → ModelDef  → Design matrix  → Condition of X →
```

First order	lin	X1 X2 X3
Second order	Quad	X1.^2 X2^2 X3.^2
	Cpb	X1* X2*
Third order	Cube	X1.^3 X2.^3 X3.^3
	Cpt	X1* X2* X3*
	$(x_i)^2 * (x_j)$ i=1,2, …,npar-1; j = 1,2, …, npar	

Models using designMatrix program: The models considered in polyModel.m are pure, linear/quadratic/cubic/quartic with one (univariate) or more number of (multi-variate in) x variables. Further a combination of them with and without cross products in second and third order are also generated. The output of autotest_desmat2015 amply demonstrates the vectors for different number of X columns ranging from 0 to 3.

```
MatLabProg A3-1
%
%  autotest_desmat2015.m   (R S Rao) 10-11-15,(11/8/97, 09/06/94 Univ of Parma, Italy)
%

f = [1 2]';
            [one,lin,quad,cube,quartic,cpb,cpt,qpt]  = desmat2015(f)
f = [1 2 ; 2 3];
            [one,lin,quad,cube,quartic,cpb,cpt,qpt]  = desmat2015(f)
f = [1 2 3; 2 3 4; ];
[one,lin,quad,cube,quartic,cpb,cpt,qpt]  = desmat2015(f)
f = [1 2 3 4; 2 3 4 5; ];
[one,lin,quad,cube,quartic,cpb,cpt,qpt]  = desmat2015(f)

 f = [];
            [one,lin,quad,cube,quartic,cpb,cpt,qpt]  = desmat2015(f)
```

```
>> autotest_desmat2015
```

#colums				
1	**2**	**3**	**4**	0
f = 1 2 one = 1 1 lin = 1 2 quad = 1 4 cube = 1 8 quartic = 1 16 cpb = [] cpt = [] qpt = []	f = 1 2 2 3 one = 1 1 lin = 1 2 2 3 quad = 1 4 4 9 cube = 1 8 8 27 quartic = 1 16 16 81 cpb =	f = 1 2 3 2 3 4 one = 1 1 lin = 1 2 3 2 3 4 quad = 1 4 9 4 9 16 cube = 1 8 27 8 27 64 quartic = 1 16 81 16 81 256 cpb =	f = 1 2 3 4 2 3 4 5 one = 1 1 lin = 1 2 3 4 2 3 4 5 quad = 1 4 9 16 4 9 16 25 cube = 1 8 27 64 8 27 64 125 quartic = 1 16 81 256 16 81 256 625 cpb = 2 3 4 6 8 12 6 8 10 12 15 20 cpt = 6 8 24 24 30 60 qpt = 24 120	f =[] one =[] lin =[] quad =[] cube =[] quartic= [] cpb = [] cpt = [] qpt = []

	2 6 cpt = [] qpt = []	2 3 6 6 8 12 cpt = 6 24 qpt = []		

The vectors (lin, quad etc.) are components in generating X matrix. The numerical values for varying number of columns (1 to 3) are given in table (table A3-1)

Numerical X matrix for typical Models using designMatrix matlab function

```
MatLabProg A3-2:
%
% X2015.m   (R S Rao) 4/13/93, 10/27/1997,10/21/2011
%

function X2015(x)

[one,lin,quad,cube,quartic,cpb,cpt,qpt]  = desmat2015(x);
[ModelPoly] = polyModels;
%
% y = mean(y)
X_y   = [one]
format shortg
%
% y = f(x)
z1 =  ModelPoly{1,:}
X_xy =  [one eval(z1)]
%
%  y = a0 + [lin quad cpb]* par
z9 =  ModelPoly{9,:}
X_fullQuad = [one  eval(z9)]
```

- Any desired can be picked up from ModelPoly(i,) vector. X matrix can also be generated using eval function.
- Ex.:
  - → X_9 = [one eval(z9)]
  - → X_1 = [one eval(z1)]

Table A3-1: X-matrices for different number of columns

x	X		
	Model :Mean	[lin]	Model : [lin quad cpb]
	1	1 1	1 1 1
1	1	1 2	1 2 4
	1	1 3	1 3 9
2	1	1 4	1 4 16
	1	1 5	1 5 25
3	1	1 6	1 6 36

- x is a vector of six points
- X is a column vector of ones (size 6x1) for mean model.
- X matrix for linear model contains intercept term (col 1) and x-data points (col 2). For quadratic model, the third vector (col 3) is squares of elements of x.

4 5 6				Since there is one x, binary product terms are not there (or null [])
Col 1	1	1 2	1 2 3	

x	X		
	Model :Mean	[lin]	• x matrix contains two variables • Second variable (col. 2) incidentally is square root of col 1. • X matrix for lin model has three vectors corresponding to regression parameters (a0,a1,a2)
1 1 2 1.414 3 1.7321 4 2 5 2.2361 6 2.4495	1 1 1 1 1 1	1 1 1 1 2 1.4142 1 3 1.7321 1 4 2 1 5 2.2361 1 6 2.4495	
Col 1	1	1 2 3	

Model : [lin quad cpb]						
	1	1	1	1	1	1
	1	2	1.4142	4	2	2.8284
	1	3	1.7321	9	3	5.1962
	1	4	2	16	4	8
	1	5	2.2361	25	5	11.18
	1	6	2.4495	36	6	14.697
Col	1	2	3	4	5	6
	one	← lin	-><-	quad ->	cpb	

• This is a full quadratic model containing linear (col. 2,3) Quadratic (col. 4,5) and binary cross product term (col 6) in X matrix

Information, dispersion, hat matrices

From design (X) matrix, information/dispersion/catcher and hat matrices are calculated which shed information on special distribution of x space, condition of matrix regarding orthogonality, inter column (variable) correlation etc. before response measurements (y) are made. The use of experimental design in enhancing the information content can also be assessed. Thus, examples chosen are simple numerical vectors to matrices to enhance the comprehension of numerical computation, matrix implementation through software without any advanced tools.

```
MethodBase.Reg A3-3
%
%    infmat.m (R S Rao) 20/02/1993; 22-5-15
%

function yy = infmat(z)
if nargin <1
             x = [1:9]';
             z= [ones(9,1),x]
```

	end

Elimination of unit vector from X:For regression models with intercept term, X marix contains 'one' vector. It is eliminated in probing into characteristics of design matrix. Of course in mean centered models, this term does not exist in X, but this routine does not have any ill effect.
 	
Ex. Input: [1 1; 2 1.4142; 3 1.7321; 4 2] output: [1; 2; 3; 4]
	`%  Elimination of one vector from X`
	`%  ------------------------------------------`
	`[rz,cz] = size(z);`
	`ione = 0;`
	`for i = 1:cz`
	`colum = z(:,i:i);`
	`if all(colum == 1)`
	`ione = i;`
	`end`
	`end`
	`if ione ==1`
	`z = z(:,2:cz);`
	`end`
	`if ione == cz`
	`z = z(:,1:cz-1);`
	`end`

	Input		output
Ex.	1	1	1
	2	1.4142	2
	3	1.7321	3
	4	2	4

	Formula	Matlab code	size
$$$matrix	**x**		**Np x npar**

Information matrix (infMat): The design matrix (X) premultiplied by its transpose (X^t) results in a square matrix called information matrix of size (npar x napr).	$X^T * X$	`%  infMat`
		`%  ----------`
		`XtX = X' * X %infMat`
			npar x npar

Inverse of information matrix: The inverse of information matrix [$(XtX)^{-1}$] is known as dispersion matrix of same size as that of infMat.	$(X^T * X)^{-1}$	`%  Dispersion Matrix (inverse of infMat)`
		`%  ----------------------------------------`
		`iXtX = inv(XtX) %invInfMat or DispersionMat`
			npar x npar

Catcher matrix: The post multiplication of transpose of design matrix with inverse of information matrix is called catcher matrix	DispMat* X'	`%  Catcher Matrix (inverse of infMat * Xt)`
		`%  -----------------------------------------`
		`CatcherMat = inv(X'*X) * X'`
			Npar x np

Hat matrix:The pre-muliplication of X with catcher matrix is the popular hat matrix. H is invariant under non singular transformation i.e. Collinearity bet columns of H is irrevelent to understand behaviour	X*catcherMat	`%  Hat Matrix (X * CatcherMat)`
		`%  ----------------------------`
		`Hat = X * CatcherMat`
			Np x np

of H . It indicates the extent of leverage.			
Diagonal of hat matrix:The diagonal elements of hat matrix	diag(hatMat)	`% Diagonal of Hat Matrix (diagHat)` `% -------------------------------` `diagHat = diag(Hat), h = diagHat;`	Npar x 1
Cut-off of diagonal values of hat matrix:It is a function of number of observations and number of model parameters (npar)	2*npar/np	`%` `% Cut off value for h` `% -------------------` `cutoff_h = 2*npar/np,`	1 x 1
Determinant: The determinant of a well-conditioned matrix is non-zero and has a positive value.	Det(.)	`% determinant` `` % ``````````` `` `det_XtX = det(XtX);` `det_iXtX = det(iXtX);`	1 x 1

SVD : singular value decomposition of a matrix (square or rectangular)		`%` `% SVD` `% ---` `[U s V] = svd(X),`	

Diagonal elements of hat matrix:Their magnitudes throw light on spacing of x values and regarding outliers.

KB. A3-1: KBs for diagonal elements of hatMat

If wide variation in h(i,i)
Then non homogenous spacing of rows of X

If max [h(i,i)] is not considerably smaller than 1
Then outlier goes undetected when residuals are observed
If max [h(i,i)] is close to 1
Then robust regreession does not work

Properties of hat matrix:The numerical characteristics of transpose, square and rank of hat matrix, their Matlab code with examples follow (MatLabProg A3-3).

Exam 7.1: For the x vector is [1;2;3], the design and other matrices are calculated.

Table A3-2: Hat matrix

X =		infMat =	catcherMat =

```
X =
1
2
3
X =
     1     1
     1     2
     1     3
```

```
infMat =
     3     6
     6    14
invInfMat =
    2.3333   -1.0000
   -1.0000    0.5000
```

```
catcherMat =
    1.3333    0.3333   -0.6667
   -0.5000    0.0000    0.5000
Hat =
    0.8333    0.3333   -0.1667
    0.3333    0.3333    0.3333
   -0.1667    0.3333    0.8333
```

		Table A3-2b: Hat matrix properties
HatMat = X *inv(X'*X) *X'		HatMat = 0.8333 0.3333 -0.1667 0.3333 0.3333 0.3333 -0.1667 0.3333 0.8333
Properties		Example
HatMat and its trace are equal	transposeHat = Hat' transposeHat-Hat =0	transposeHat = 0.8333 0.3333 -0.1667 0.3333 0.3333 0.3333 -0.1667 0.3333 0.8333 transposeHat-Hat = 1.0e-15 * 0 0.1110 0.3331 -0.1110 0 0.1110 -0.3331 -0.1110 0
HatMat and its square are equal	hatSquare = Hat*Hat hatSquare-Hat =0	hatSquare = 0.8333 0.3333 -0.1667 0.3333 0.3333 0.3333 -0.1667 0.3333 0.8333 hatSquare-Hat= 1.0e-15 * -0.3331 -0.1110 0.2220 0 0.1665 0.2220 0.4996 0.3331 0.2220 Is a zero matrix
trace and rank of hat matrix are equal	traceHat = trace(Hat) rankHat = rank(Hat) rankHat-traceHat =0	traceHat = 2 rankHat = 2 rankHat-traceHat = 0

Applications of Hat matrix: some of typical applications of hat matrix are

- ✓ Calculation of ycal of regression model
- ✓ Studentized residuals
- ✓ Cut-off values of h
- ✓ Predictive residuals and press
- ✓ Detecting outlying observations with regard to x-values i.e. Those excessively influencing regression parameters and other statistics
- ✓ Hat matrix is also called projection matrix as it projects vector of observed y onto vector of ycal.

  In yesteryears, ycal was also called yhat $\left(\hat{y}\right)$

Press	Function [Press] = press2015(X,x,y) [resy] = ordResid(X,x,y) [diaghat] = hatMat(X) pred_res = resy./(one - diagHat) press = pred_res'*pred_res

Studendized residual	

Appendix A4: Inverse of a matrix

Solution of linear algebraic equations: In regression analysis, least squares estimates of parameters of model are estimated by solving $X * par = y$. The solution is $par = \left(X^T * X\right)^{-1} * y$.

Inverse of matrix: Assuming that X (or $X^T * X$) is well conditioned , ordinary inverse is used for LSS

$par.inv = inv\left(X^T * X\right) * y$

Matlab built in function: At the command line or in a function $par.Xbyy = X \backslash y$ is par vector

Pseudo inverse: If $\left(X^T * X\right)$ is ill conditioned (i.e. singular/nearly singular) simple inverse fails or results in wrong values of parameters and/or inflated standard deviations of parameters. In such cases, pseudo inverse (pinv in Matlab software) gives optimal values.

$par.pinv = pinv\left(X^T * X\right) * y$

	par		
$X^T * X$	$par.inv = inv\left(X^T * X\right) * y$	$par.Xbyy = X \backslash y$	$par.pinv = pinv\left(X^T * X\right) * y$

Appendix A5 :Condition of X matrix

The design matrix X is mostly rectangular. To assess characteristics (determinant, inverse etc.), it is converted into a square matrix by pre- or post- multiplication with X^T (MethodBase.X A5-1).The numerical examples using identity/singular/partially correlated matrices and KB are described in Output A5-1.

MethodBase.X A5-1

```
%
%          KB_Xcond.m  18/3/1997 ; 9/11/15
%

function kb_xcond(X)
    dispst('X matrix')

    [r,c]=size(X);
if r ~=c
dispst(['????????? Rectangular matrix,
 X','''*X calculated'])
        X = X'*X
end
```

Output A5-1: om999.m

```
X =
     1     0
     0     1

X matrix
~~~~~~~~
svd eig invX

~~~~~~~~~~~~
eigX =
     1
     1
U =
     1     0
     0     1
s =
     1     0
     0     1
V =
     1     0
     0     1
invX =
     1     0
     0     1
```

	pinvX = 1 0 0 1 Matrix condition characteristics detX = 1

`% Condition number of x' *x and inv(x' * x)`
`% ''''''''''''''''''''''''''''''''''''''''`
`condX= cond(XtX);`
`rankX = rank(XtX);`
`CondX_eig= condeig(X),`
`%`
`CondX_est=condest(X);CondX_r=rcond(X);`
`%`
`CondX_1=cond(X,1);CondX_2=cond(X,2);`
`CondX_Fro=cond(X,'fro');CondX_inf=cond(X,'inf');`
	condX =
	1
	rankX =
	2
	CondX_eig =
	1
	1
	CondX_est =
	1
	CondX_r =
	1
	CondX_1 =
	1
	CondX_2 =
	1
	CondX_Fro =
	2.0000
	CondX_inf =
	1

```
function om999
clean
  v = [ 1 2 3];
   X = eye(2,2),kb_xcond(X)
    X = ones(2,2),kb_xcond(X)
    X = ones(1,1),kb_xcond(X)
    X = zeros(1,1),kb_xcond(X)
    X = zeros(2,2),kb_xcond(X)
  X = [v; sqrt(v)],kb_xcond(X)
  X = [ 1 2; 3 5],kb_xcond(X)
  X = [v' v'.^2],kb_xcond(X)
```

A5-1b: om999.m	A5-1c: om999.m	A5-1d: om999.m	A5-1e: om999.m
X = 1 1 1 1 X matrix	X = 1 X matrix	X = 0 X matrix	X = 0 0 0 0 X matrix

```
~~~~~~~~
          svd eig invX
~~~~~~~~~~~~~
eigX =
   0
   2
U =
  -0.7071  -0.7071
  -0.7071   0.7071
s =
   2   0
   0   0
V =
  -0.7071  -0.7071
  -0.7071   0.7071
invX =
  Inf  Inf
  Inf  Inf
pinvX =
   0.2500   0.2500
   0.2500   0.2500
Matrix condition
characteristics
detX =
   0
condX =
  Inf
rankX =
   1
CondX_eig =
   1.0000
   1.0000
CondX_est =
  Inf
CondX_r =
   0
CondX_1 =
  Inf
CondX_2 =
  Inf
CondX_Fro =
  Inf
CondX_inf =
  Inf
```

```
~~~~~~~~
          svd eig invX
          ~~~~~~~~~~~~~
eigX =
   1
U =
   1
s =
   1
V =
   1
invX =
   1
pinvX =
   1
Matrix condition
detX =
   1
condX =
   1
rankX =
   1
CondX_eig =
   1
CondX_est =
   1
CondX_r =
   1
CondX_1 =
   1
CondX_2 =
   1
CondX_Fro =
   1
CondX_inf =
   1
```

```
          ~~~~~~~~
svd eig invX
          ~~~~~~~~~~~~~
eigX =
   0
U =
   1
s =
   0
V =
   1
invX =
  Inf
pinvX =
   0
      Matrix condition
detX =
   0
condX =
  Inf
rankX =
   0
CondX_eig =
   1
CondX_est =
  Inf
CondX_r =
   0
CondX_1 =
  NaN
CondX_2 =
  Inf
CondX_Fro =
  NaN
CondX_inf =
  NaN
```

```
~~~~~~~~
          svd eig invX
          ~~~~~~~~~~~~~
eigX =
   0
   0
U =
   1   0
   0   1
s =
   0   0
   0   0
V =
   1   0
   0   1
invX =
  Inf  Inf
  Inf  Inf
pinvX =
   0   0
   0   0
     Matrix condition
detX =
   0
condX =
  Inf
rankX =
   0
CondX_eig =
   1
   1
CondX_est =
  Inf
CondX_r =
   0
CondX_1 =
  NaN
CondX_2 =
  Inf
CondX_Fro =
  NaN
CondX_inf =
  NaN
```

Output A5-1f

```
X =
    1.0000    2.0000    3.0000
    1.0000    1.4142    1.7321
                           X matrix
                           ~~~~~~~~
        ????????? Rectangular matrix, X'*X calculated
        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
X =
    2.0000    3.4142    4.7321
    3.4142    6.0000    8.4495
    4.7321    8.4495   12.0000
                         svd eig invX
                         ~~~~~~~~~~~~~
eigX =
   -0.0000
    0.1287
   19.8713
U =
```

```
X =
     1     2
     3     5

X matrix

~~~~~~~~

svd eig invX

~~~~~~~~~~~~~
eigX =
   -0.1623
    6.1623
U =
   -0.3574   -0.9340
   -0.9340    0.3574
s =
    6.2429         0
         0    0.1602
```

```
   -0.3104    0.8165    0.4869
   -0.5491    0.2641   -0.7929
   -0.7760   -0.5134    0.3663
s =
   19.8713         0         0
         0    0.1287         0
         0         0    0.0000
V =
   -0.3104    0.8165    0.4869
   -0.5491    0.2641   -0.7929
   -0.7760   -0.5134    0.3663
invX =
   1.0e+15 *
   -1.6979    2.7653   -1.2775
    2.7653   -4.5036    2.0806
   -1.2775    2.0806   -0.9612
pinvX =
    5.1855    1.6844   -3.2457
    1.6844    0.5573   -1.0324
   -3.2457   -1.0324    2.0790
            Matrix condition
detX =
  -7.6277e-16
condX =
   1.6672e+18
rankX =
     2
CondX_eig =
    1.0000
    1.0000
    1.0000
CondX_est =
   1.1018e+17
CondX_r =
   9.0757e-18
CondX_1 =
   2.3544e+17
CondX_2 =
   1.6672e+18
CondX_Fro =
   1.4234e+17
CondX_inf =
   2.3544e+17
```

```
V =
   -0.5061    0.8625
   -0.8625   -0.5061
invX =
   -5.0000    2.0000
    3.0000   -1.0000
pinvX =
   -5.0000    2.0000
    3.0000   -1.0000
               Matrix
condition characteristics
detX =
   -1.0000
condX =
   38.9743
rankX =
     2
CondX_eig =
    1.0124
    1.0124
CondX_est =
   56.0000
CondX_r =
    0.0179
CondX_1 =
   56.0000
CondX_2 =
   38.9743
CondX_Fro =
   39.0000
CondX_inf =
   56.0000
```

```
X =
     1     1
     2     4
     3     9
                         X matrix
                         ~~~~~~~~
        ????????? Rectangular matrix, X'*X calculated
        ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
X =
    14    36
    36    98
                       svd eig invX
                       ~~~~~~~~~~~~
eigX =
    0.6827
  111.3173
U =
```

```
   -0.3469   -0.9379
   -0.9379    0.3469
s =
  111.3173         0
         0    0.6827
V =
   -0.3469   -0.9379
   -0.9379    0.3469
invX =
    1.2895   -0.4737
   -0.4737    0.1842
pinvX =
    1.2895   -0.4737
   -0.4737    0.1842
            Matrix condition
detX =
   76.0000
condX =
  163.0465
rankX =
     2
CondX_eig =
     1
     1
CondX_est =
  236.2632
CondX_r =
    0.0042
CondX_1 =
  236.2632
CondX_2 =
  163.0465
CondX_Fro =
  163.0526
CondX_inf =
  236.2632
```

Supplementary information

SI-1: Typical statistical packages

Name	Promotor	Language
Maple	Maplesoft	
Mathematica	Wolfram Research	
MATLAB	MathWorks	C++ Java
Minitab	Minitab Inc.	
Origin	OriginLab	C++
R	R Foundation	C Fortran R
SAS	SAS Institute	
SPlus	Insightful Inc.	
SPSS	IBM	Java
Stata	StataCorp LP	C
Statgraphics	Statpoint Tech.	C++

Name	Promotor	Language
BMDP	Statistical Solutions	
Epi Info	Centers for Disease Control and Prevention	Microsoft C#
MedCalc	MedCalc Software bvba	
NLOGIT	Econometric Software, Inc. William Greene	Fortran C++
Orange	Bioinformatics Laboratory, Faculty of Computer and Information Science, University of Ljubljana	Python Cython

	Inc.	
Statistica	Dell Software	C++
StatPlus	AnalystSoft	
Statsmodels	Statsmodels Developers	Python C
Statwing	Statwing Inc.	
SYSTAT	Systat Software Inc.	
TSP	TSP International	Fortran
UNISTAT	Unistat Ltd	
WINKS	TexaSoft	Fortran Visual Basic

SI 2: Toolboxes of MATLAB

- Optimization
- Neural Network
- Partial Differential Equation
- Statistics and Machine Learning
- Wavelet
- Global Optimization
- Fuzzy Logic
- Curve Fitting

Toolboxes of Matlab

- Aerospace
- Antenna
- Audio System
- Bioinformatics
- Communications System
- Computer Vision System
- Control System
- Data Acquisition
- Database
- Datafeed
- DSP System
- Econometrics
- Financial Instruments

- Image Acquisition
- Image Processing
- Image Processing and Computer Vision
- Instrument Control
- LTE System
- Mapping
- Model Predictive Control
- Model-Based Calibration
- OPC
- Parallel Computing
- Phased Array System
- Polyspace Code Prover

- RF
- Robotics System
- Robust Control
- Signal Processing
- Symbolic Math
- System Identification
- Trading
- Vehicle Network
- Vision HDL
- Wireless CoSymbolic Math
- WLAN System

SI 3: Statistics Toolbox of MATLAB

Statistics Toolbox
Getting started

Fx: Functions
File I/O

Parametric Regression Analysis
Linear

User's Guide		Generalized Linear Regression
Fx: Functions	Parametric Regression Analysis	Nonlinear Regression
Examples	Multivariate Methods	
Demos		
Release Notes		

SKI4:Typicalm (function) files for regression in Statistics Toolboxe of MATLAB

Linear Regression	
anova	Analysis of variance for linear model
lasso	Regularized least-squares regression using lasso or elastic net algorithm
mnrfit	Multinomial logistic regression
mvregress	Multivariate linear regression
plsregress	Partial least-squares regression
regress	Multiple linear regression
robustfit	Robust regression
stepwisefit	Stepwise regression

Generalized Linear Regression	
glmfit	Generalized linear model regression
Lassoglm	Lasso or elastic net regularization for generalized linear model regression

Nonlinear Regression	
Nlinfit	Nonlinear regression

AUTHORS' ADDRESSES

1. **R. Sambasiva Rao**
School of Chemistry,
Andhra University,
Visakhapatnam 530 003, A.P
2. **K. Ramakrishna**
Department of Chemistry,
Institute of Science, GITAM University,
Visakhapatnam, 530 017, India
Email: karipeddirk@gmail.com