**Revised Chapter 17 in *Specifying and Diagnostically Testing Econometric Models* (Edition 3)**


## Chapter 17



## Model Building Using Nonlinear Nonparametric Methods

## 17.0 Introduction

In contrast to Chapter 14 that was concerned with nonlinear methods that implicitly assumed normally distributed errors[1], the procedures discussed in this chapter represent nonparametric

1 While GAM, ACE and MARS models use nonparametric methods in the first stage, once the smoothing transformation is selected, OLS is used to estimate the

nonlinear model alternatives.[2] When confronted with a problem where nonlinearity cannot be reasonable assumed away, there are a number of possible ways to proceed.[3]

Option 1: Model exact functional form. Direct Estimation of a nonlinear specification is clearly the best choice if the model is known for certain.

Option 2: GAM and ACE Models. The GAM model is an especially valuable nonlinear exploratory tool that investigates possible nonlinearity by fitting polynomials to the right hand side variables. Graphic analysis of the smoothed series gives feedback on whether there is low dimensional nonlinearity. ACE models smooth both the right and left hand side variables and make estimation of such models as $y = \exp(x + z^2)e$ possible. While neither method detects variable interactions, both allow manual incorporation of interaction variables in the model. Comparison of GAM leverage plots with OLS plots indicate the type of nonlinearity that is being estimated.

Option 3: MARS Modeling provides an automatic method to identify locally linear partitions of the data based on threshold values and potential interactions among the variables. As a special case of MARS modeling, lags of the dependent variable can be included in the modeling dataset to handle time series applications in the spirit of Threshold Autoregressive (TAR) models. Model nonlinearity can be displayed using leverage plots that map the knots and interactions found at specific regions in the n-dimensional nonlinear space. The MARS estimator is of the shrinkage class that provides a way to reduce the number of explanatory variables while allowing for the possibility of nonlinearity. An advantage of MARS over GAM and ACE models is that 2-3 way interactions can be detected and modeled. Graphical analysis of the knot vectors that are identified and used in the OLS estimation step involving transformed data can be inspected to identify specific thresholds present in the data.

Option 4: LOESS Modeling. This approach is especially useful if there is substantial local structure. It is not suitable for large datasets or where there are many right hand side variables.

Option 5: Recursive Covering Models. This approach, discussed in Friedman (1996a) and Hastie-Tibshirani-Friedman (2001, 415), unifies the advantages of the K nearest neighbor modeling approach that involves a large number of overlapping regions based on the training dataset and a CART approach that identifies a smaller number of highly customized disjoint regions.[4]

---

final model. These approaches are discussed in chapter 14. The various LOESS models discussed in chapter 14 work in a relatively similar manner.

2 This draft has been helped by the many suggestions of Bill Lattyak. Any remaining problms are my responsibility.

3 Prior chapters have been concerned with various means by which to test for nonlinearity such as recursive residual analysis that was discussed in chapter 9 and various nonlinearity tests such as the Hinich test that were discussed in chapter 7.

4 Discussion of recursive covering was removed from Hastie-Tibshirani-Friedman (2009) that added discussion of the more capable Random Forest method of analysis.

Option 6: The Projection Pursuit model is a nonparametric multiple regression approach that only assumes continuous derivatves for the regression surface. In contrast to recursive partitioning methods that can be characterized as local averaging procedures, projection pursuit models a regression surface as a sum of general smooth functions of linear combinations of the predictor variables. Graphical analysis using leverage plots can be employed to interpret the estimated model.

Option 7: Exploratory Projection Pursuit Analysis can be used to explore possible nonlinearity in multivariate data by assigning a numerical index to every projection that is a function of the projected data density The number of large values in the projection index indicates the complexity of the data.

Option 8: Random Forest Modeling. A random forest model uses bagging to improve the performance of a CART type model. The performance of a classification problem is improved by estimating many models and by voting to select the appropriate class. For a problem involving a continuous left-hand side variable, averaging is used to improve the out-of-sample performance. The basic idea of the random forest method is to randomly select a bagged dataset, estimate a model using a fixed number of randomly selected input variables and, using this model make predictions for the out-of-bag data. This is repeated multiple times. The random forest technique is especially suitable for classification problems involving many possible outcomes. While probit and logit models can be used when there are a small number of classes such as in the models discussed in chapter 3, for research problems containing large numbers of classes, these methods are not suitable. Random Forest models can be used successfully in such cases, as well as cases typically addressed through classical probit and logit models. For continuous left hand side variable problems random forest methods are suitable for high dimension nonlinear problems involving many right hand side variables. For near linear models or for models with few right hand side variables, random forest models will not perform as well.

Option 9. If there is no left-hand side variable, the options listed above are not applicable. Cluster analysis that includes both k-means and hierarchical models attempts to place variables in a predetermined number of classes and can be used for exploratory data analysis.

The goal of the rest of this chapter is to outline the use of options 5-9. The developer of B34S was fortunate to be able to obtain Fortran code for a number of these procedures as GPL code or directly from the originators of the methodology.[5]

---

5 Discussion of the recursive covering approach (rcover) and code for its implementation was obtained from Friedman (1996a, 1996b). His work was supported by the Department of Energy under contract number DE-AC03-76SF00515 and NSF foundation grant # DMS-9403804. Information on the projection pursuit (ppreg) method and code was obtained from Friedman-Stuetzle (1981) based on the above mentioned Department of Energy grant. Friedman (1987) is the basic reference for exploratory projection pursuit. Friedman (1989) is the basic reference for regularized discriminate analysis (rda). Breiman (2001) is the basic source for the random forest approach (ranforest). Version 3.1 of the Breiman code is what has been implemented. All of the Fortran code has been extensively extended and enhanced before implementation in the B34S Data Analysis Program. The developer of B34S is grateful for being able to obtain

## 17.1 Recursive Covering – A Compromise between K- NN methods and CART

Recursive covering (rcover) is employed in classification problems involving discrete choice datasets. Assume a model $y = f(x) + e$ where $f(x)$ is a single valued deterministic function of $k$ predictor variables. The space $x \in R^k$ of input variables is covered by a set of local regions $\{R_m\}_1^M$ for which the model "learns" a simple approximator $\hat{f}_m(x)$ based on either 0, 1 or 2 order polynomials such that in that region $\hat{y} = \hat{f}_{m^*(x)}(x)$. Define $\| x - x' \|$ as a distance measure between $x$ and $x'$. The local region has $size(R_m) = ave_{x,x'} \in R_m \| x = x' \|$ chosen in which the prediction point is most centered. The two goals are to minimize bias-squared and variance. The smaller $size(R^m)$ the less the bias-squared since a low order polynomial can more likely approximate the region. However this will be associated with a larger variance. The K nearest neighbor (K-NN) local learning method defines each local learning region $R_m$ in terms of its center $u_m$ and the $K$ closest points. The region is defined as

$$R_m(u_m) = \{x_i \mid \| x_i - u_m \| \le d_m^{(j)}\} \qquad (17.1\text{-}1)$$

where $d_m^{(j)}$ is the $j^{th}$ order statistic of $\{\| x_i - u_m \|_{i=1}^j$. The K-NN approach has been shown to work especially well in settings where there are few independent variables. If there are many independent variables the shape of the region becomes larger and more complex and less able to be modeled as a low order polynomial. The CART model, also called a recursive partitioning model, uses a top down strategy to recursively partition the data. The top region $R_0$ contains all the data. Next two sub-regions are formed. From each of these sub-regions two more sub-regions are formed, thus modeling the data into a tree structure. For each split point one split value is used. The splitting occurs until a region meets a local terminal criterion. The linear splitting function $g(x, \alpha) = a' x$ is defined in terms of the input data $x$ and a set of parameters $\alpha$. Each split chooses the direction $\alpha$ that gives the most improvement by the local approximator $\hat{f}_m(x)$ in each sub-region insuring directions of high bias are split first. From the parent region $R$ two sub-regions $R_l$ (region left) and $R_r$ (region right) are defined. The logic is for $x \in R$ if $g(x, \alpha^*) \le s^*$ => $x \in R_l$. If $g(x, \alpha^*) > s^*$ => $x \in R_r$ where $s^*$ is the split point. Although the CART model produces a graphic that visually displays the in-out relationship, unlike the K-NN model that produced overlapping regions, this is accomplished by producing disjoint regions. Each prediction point is contained in only one region resulting in bias near region boundaries. Data fragmentation is also a concern and the results are often very sensitive to minor changes in the training data.

---

this GPL code. The discussion of these approaches is heavily based on these sources together with the monumental reference Hastie-Tibshirani-Friedman (2001, 2009).

The recursive covering approach combines features of K-NN and CART. First a large number of overlapping regions are produced and modeled using a low order polynomial. Like CART a splitting function $g(x,\alpha)$ is defined but unlike CART two split points $(s_1^* < s_2^*)$ are used. The algorithm works as follows: For $x \in R$ if $g(x,\alpha^*) \le s_2^* \Rightarrow x \in R_l$. If $g(x,\alpha^*) > s_1^* \Rightarrow x \in R_r$. $R_l$ includes $s_1^*$ and terminates on the right at $s_2^*$ and on the left at a value $< s_1^*$. $R_r$ has a lower point at $s_1^*$ includes $s_2^*$ and terminates on the right at a value $> s_2^*$. Points in the interval $s_1^* < g(x,\alpha^*) \le s_2^*$ are assigned to both regions $R_l$ and $R_r$. The process stops if the number of data points falls below a user set threshold as is the case with the K-NN approach.

### 17.2 Regularized Discriminate Analysis – Linking linear and quadratic discriminate analysis

The regularized discriminate analysis (RDA) approach to classification, proposed by Friedman (1989), can be thought of as a compromise between linear discriminate analysis (LDA) that assumes all classes in a k classification model have a common covariance matrix $\hat{\Sigma}$ and quadratic discriminate analysis (QDA) that estimates a covariance matrix $\hat{\Sigma}_k$ for each of the $k$ classes. RDA models have been found to be useful in small sample high-dimensional problems. The regularized covariance matrix $\hat{\Sigma}_k(\alpha)$ is defined as

$$\hat{\Sigma}_k(\alpha) = \alpha\hat{\Sigma}_k + (1-\alpha)\hat{\Sigma}. \qquad (17.2\text{-}1)$$

The RDA approach searches over the range 0-1 to set the appropriate $\alpha$ where $\alpha = 0$ ($\alpha = 1$) implies the LDA (QDA) model. Options allow the LDA covariance matrix itself to be shrunk toward the scalar covariance.

$$\hat{\Sigma}(\gamma) = \gamma\hat{\Sigma} + (1-\lambda)\hat{\sigma}^2 I \qquad (17.2\text{-}2)$$

### 17.3 Projection Pursuit Regression

The Projection Pursuit Modeling approach, PPR, (implemented in B34S with the command **ppreg**) makes few general assumptions about the regression surface except the implicit assumption that the underlying but unknown function can be differentiated. While Friedman-Stuetzle (1981) provides a comprehensive discussion of the procedure, Hastie-Tibshirani-Friedman (2001, 347-350) contains a simplified and clear discussion from which this summary is based. Assume $X$ contains $k$ columns of data and the desired model is $y = f(X) + e$. Define $\omega_m$, $m = 1,2,...,M$ as unit k-vectors of unknown parameters. The projection pursuit regression estimates $f(X) = \sum_{m=1}^{M} g_m(\omega_m^T X)$, where $\omega_m^T \equiv \omega_m^{'}$, which is an additive model in terms of derived features $V_m = \omega_m^T X$. The unknown parameters $g_m$ as well as the directions $\omega_m$

are estimated to form the ridge function $g_m(\omega_m^T X)$. The scalar variable $V_m$ is the projection of $X$ onto the unit vector $\omega_m$ selected to fit the model well. Nonlinear functions are modeled as linear combinations. The example given was a model that contained two input series where the product $x_1 x_2$ was part of the model. It was noted that $x_1 x_2 = [(x_1 + x_2)^2 - (x_1 - x_2)^2]/4$. In general if $M$ is taken arbitrarily large, the PPR model can approximate any continuous function in $IR^k$ space and is thus a universal approximator. Hastie-Tibshirani-Friedman (2009, 390) note "this generality comes at a price. Interpretation of the fitted model is usually difficult, because each input enters into the model in a complex and multifaceted way. As a result, the PPR model is most useful for prediction, and not very useful for producing an understandable model of the data." It can be argued that this is essentially correct, although with judicious use of 3-D graphs and leverage plots it is possible to simulate what the surface looks like once assumptions are made about various values of the input variables.[6]

The PPR algorithm's goal is to minimize

$$\sum_{i=1}^{N}[y_i - \sum_{m=1}^{M} g_m(\omega_m^T x_i)]^2 \tag{17.3-1}$$

by alternately changing $g_m$ and $V_m$ over $m = 1,2,\ldots$ where $M$ is the number of trees in the model. A scatter plot smoother, such as a smoothing spline, can be used to obtain an estimate of $g$ given $\omega_1^T x_i$. Assuming just one term (M=1) given the estimate $g$, a Gauss-Newton search can be used to update $\omega_1^T x_i$. The update formula is

$$g(\omega_{1,new}^T x_i) \approx g(\omega_{1,old}^T x_i) + g'(\omega_{1,old}^T x_i)(\omega - \omega_{1,old})^T x_i \tag{17.3-2}$$

which is used to solve (17.3-1) from

$$\sum_{i=1}^{N}[y_i - \sum_{m=1}^{M} g_m(\omega_m^T x_i)]^2 \approx \sum_{i=1}^{N} g_m'(\omega_{m,old}^T x_i)^2 [(\omega_{m,old}^T x_i + \frac{y_i - g_m(\omega_{m,old}^T x_i)}{g_m'(\omega_{m,old}^T x_i)}) - \omega_m^T x_i]^2 \tag{17.3-3}$$

The right hand side of (17.3-3) is minimized by an OLS regression of $\omega_{m,old}^T x_i + \frac{y_i - g_m(\omega_{m,old}^T x_i)}{g_m'(\omega_{m,old}^T x_i)}$ on $x_i$, with weights $g_m'(\omega_{m,old}^T x_i)^2$ and a constraint on the intercept to equal 0.0 to produce $\omega_{m,new}$. These steps are repeated until convergence in achieved for that M value. After convergence is achieved, M is set to M+1 and the process starts again up to the upper limit of M. The PPR implementation allows the search to proceed over a range of M values from a lower

---

6 If M=1 we have a single index model which is a bit more general than a linear regression and is easier to be understood.

bound (:mu) to an upper bound (:m) to investigate how the sum of squared errors, the sum of absolute errors and the maximum error change. Examples showing the sensitivity of these values in models with varying numbers of observations and varying amounts of noise and a number of nonlinear models with known answers are given later.

Equation 17.3-1 can be reverse engineered by the use of the variables `%omega` and `%gamma` which is shown with the gas data in the example below. OLS is used to scale the projection `test=%gamma*(transpose(%omega)*transpose(%x)).` Test output verifies these calculations. In practice forecasts are usually done from within the `ppreg` command.

Table 17.1 Reverse Engineering the calculation of YHAT from a Projection Pursuit Model

```
b34sexec options ginclude('gas.b34'); b34srun;
b34sexec matrix$
call loaddata;
call echooff;
call  olsq(gasout gasin{1 to 6} gasout{1 to 6} :print)$
call ppreg(gasout gasin{1 to 6} gasout{1 to 6} :savex
                                               :print)$
/;  call print(%omega,%gamma);
/; uses Hastie-Tibshirani-Friedman (2009) page 389 eq 11.2

test=%gamma*(transpose(%omega)*transpose(%x));
test=vector(:test);
call olsq(%yhat test :noint :print);
adjtest=%coef(1)*test;
dd=adjtest-%yhat;
call tabulate(%y,%yhat,%res,test,adjtest,dd);
b34srun $
```

Edited output from running the example in Table 17.1 is given next. Note that the vector dd is everywhere 0.0.

```
Variable    Label                                      # Cases      Mean        Std. Dev.    Variance      Maximum       Minimum

 TIME      1                                             296   148.500       85.5921       7326.00       296.000       1.00000
 GASIN     2 Input gas rate in cu. ft / min              296 -0.568345E-01   1.07277       1.15083       2.83400      -2.71600
 GASOUT    3 Percent CO2 in outlet gas                   296   53.5091       3.20212       10.2536       60.5000       45.6000
 CONSTANT  4                                             296   1.00000       0.00000       0.00000       1.00000       1.00000

 Number of observations in data file   296
 Current missing variable code         1.000000000000000E+31

 B34S(r) Matrix Command. d/m/y 24/ 8/09. h:m:s  8:43:29.


 =>  CALL LOADDATA$


 =>  CALL ECHOOFF$


 Ordinary Least Squares Estimation
 Dependent variable                     GASOUT
 Centered R**2                          0.9946641074363697
 Adjusted R**2                          0.9944329496357792
 Residual Sum of Squares                16.13858295915815
 Residual Variance                      5.826203234353124E-02
 Standard Error                         0.2413752935648784
 Total Sum of Squares                   3024.532965517241
 Log Likelihood                         7.364694190042868
 Mean of the Dependent Variable         53.50965517241379
 Std. Error of Dependent Variable       3.235044356946151
 Sum Absolute Residuals                 48.13385294539017
 F(12,      277)                        4302.965787421152
```

```
F Significance                        1.000000000000000
1/Condition XPX                       1.929993696611666E-08
Maximum Absolute Residual             1.430814663262517
Number of Observations                290

Variable   Lag    Coefficient        SE               t
GASIN       1     0.63160860E-01     0.75989856E-01   0.83117489
GASIN       2    -0.13345763         0.16490508      -0.80929968
GASIN       3    -0.44123536         0.18869442       -2.3383593
GASIN       4     0.15200749         0.19021604       0.79913078
GASIN       5    -0.12036440         0.17941884      -0.67085705
GASIN       6     0.24930584         0.10973982        2.2717902
GASOUT      1      1.5452265         0.59808504E-01    25.836234
GASOUT      2    -0.59293307         0.11024897       -5.3781279
GASOUT      3    -0.17105674         0.11518138       -1.4851076
GASOUT      4     0.13238479         0.11465530        1.1546329
GASOUT      5     0.56869923E-01     0.10083191       0.56400722
GASOUT      6    -0.42085617E-01     0.42891891E-01  -0.98120217
CONSTANT    0      3.8241094         0.85547296        4.4701698

Projection Pursuit Regression

Number of Observations                         290
Number of right hand side variables            13
Maximum number of trees                        20
Minimum number of trees                        20
Number of left hand side variables             1
Level of fit                                   2
Max number of Primary   Iterations (maxit)     200
Max number of Secondary Iterations (mitone)    200
Number of cj Iterations (mitcj)                10
Smoother tone control (alpha)                  0.000000000000000E+00
Span                                           0.000000000000000E+00
Convergence (CONV) set as                      5.000000000000000E-03
Left Hand Side Variable                        GASOUT

Series           Mean          Max           Min
GASOUT          53.51         60.50         45.60

Right Hand Side Variables

#   Series    Lag     Mean          Max          Min
  1 GASIN      1  -0.5980E-01     2.834        -2.716
  2 GASIN      2  -0.5789E-01     2.834        -2.716
  3 GASIN      3  -0.5678E-01     2.834        -2.716
  4 GASIN      4  -0.5661E-01     2.834        -2.716
  5 GASIN      5  -0.5729E-01     2.834        -2.716
  6 GASIN      6  -0.5853E-01     2.834        -2.716
  7 GASOUT     1    53.50         60.50         45.60
  8 GASOUT     2    53.48         60.50         45.60
  9 GASOUT     3    53.47         60.50         45.60
 10 GASOUT     4    53.45         60.50         45.60
 11 GASOUT     5    53.43         60.50         45.60
 12 GASOUT     6    53.42         60.50         45.60
 13 CONSTANT   0    1.000         1.000         1.000

Given # of trees                               20
# primary   iterations used                    7
# secondary iterations used                    4
# cj iterations used                           10
Residual sum of squares                        4.070586101904265
Total    sum of squares                        3024.532965517241
Mean of the Dependent Variable                 53.50965517241379
Std. Error of Dependent Variable               3.235044356946151
Sum Absolute Residuals                         23.79513466577958
Maximum Absolute Residual                      0.5945024373631753
Residual Variance                              1.469525668557496E-02

Variable Importance for Model with # Trees  20
Series Number       Importance
        7             1.00000
        8            0.800017
       10            0.787139
        4            0.718387
        9            0.683195
        2            0.585851
        5            0.566841
        3            0.545039
       11            0.525467
        6            0.392487
       12            0.222954
        1            0.217705
       13             0.00000

Ordinary Least Squares Estimation
Dependent variable                    %YHAT
Centered R**2                         0.9888463656992637
Adjusted R**2                         0.9888463656992637
Residual Sum of Squares               33.51679994344680
Residual Variance                     0.1159750863095045
Standard Error                        0.3405511507975043
Total Sum of Squares                  3005.011554057680
Log Likelihood                        -98.60622737365155
Mean of the Dependent Variable        53.50965517241374
Std. Error of Dependent Variable      3.224587392965318
Sum Absolute Residuals                77.14296774447519
1/Condition XPX                       1.000000000000000
```

```
Maximum Absolute Residual            1.193559732520846
Number of Observations               290

Variable   Lag    Coefficient         SE                t
TEST         0      1.9245949         0.71798410E-03    2680.5536

Obs       %Y          %YHAT       %RES        TEST         ADJTEST     DD
     1      52.79       52.68      0.1069       27.37        52.68       0.000
     2      52.34       52.23      0.1137       27.14        52.23       0.000
     3      52.13       52.03      0.1031       27.03        52.03       0.000
     4      51.99       51.94      0.5408E-01   26.99        51.94       0.000
     5      51.98       51.77      0.2189       26.90        51.77       0.000
     6      52.29       52.00      0.2901       27.02        52.00       0.000
     7      52.86       52.82      0.4067E-01   27.45        52.82       0.000
     8      53.91       53.70      0.2022       27.90        53.70       0.000
     9      54.84       55.10     -0.2624       28.63        55.10       0.000

   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

   280      53.01       52.87      0.1373       27.47        52.87       0.000
   281      53.96       53.71      0.2485       27.91        53.71       0.000
   282      55.75       55.56      0.1945       28.87        55.56       0.000
   283      57.06       57.27     -0.2044       29.76        57.27       0.000
   284      57.72       57.55      0.1710       29.90        57.55       0.000
   285      58.64       58.35      0.2913       30.32        58.35       0.000
   286      58.37       58.66     -0.2867       30.48        58.66       0.000
   287      58.22       58.08      0.1416       30.18        58.08       0.000
   288      57.82       57.92     -0.9682E-01   30.09        57.92       0.000
   289      57.19       57.39     -0.2003       29.82        57.39       0.000
   290      57.04       56.99      0.4161E-01   29.61        56.99       0.000
```

## 17.4 Exploratory Projection Pursuit

The goal of exploratory projection pursuit, PPEXP, is to be able to detect nonlinearity in high-dimensional multivariate data using both inspection of graphs and an index. The analysis proceeds by assigning a numerical index to every projection that is a function of the projected data density. The next step is to apply a transformation to the data to remove the structure present in the solution projection while preserving the remaining multivariate structure that is not captured by this projection. This can be thought of as transforming the data to look Gaussian in the chosen projection. Since the number of possible nonlinear shapes is vast, it may be useful to view a number of these lower dimensional representations of the data density to explore for relationships that were not anticipated. Since Friedman (1987) provides a complete discussion of the PPEXP method, only a summary will be given here. Hastie-Tibshirani-Friedman (2001, 500) (2009, 565) briefly discuss the PPEXP method and show how it is related to independent component analysis (ICA).

To motivate exploratory projection pursuit recall that the central limit theorem states that if a series $x$ has <u>any</u> distribution with mean $\mu$ and variance $\sigma^2$ then the distribution of $\bar{x}$ approaches the normal distribution with mean $\mu$ and variance $\sigma^2 / N$ as the sample size $N$ increases. This suggests that in an OLS model, any departure from normality implicit in the X matrix may not be observed since two-dimensional projections look Gaussian. Long tails or clusters in the data would be revealed by non-Gaussian projections. The goal of PPEXP analysis is to develop a projection index that will allow discovery of nonlinear structure in the main body of the data (not the tails) that is manifested by clusters. How fast the value of this index tails off indicates the degree of nonlinearity in the data since more linear models can be characterized by fewer projections. Three-dimensional graphical displays indicate an overall view of the process while two-dimensional graphs of slices can highlight the differences found by each projection by observation. The interpretation of the three-dimensional graph must be tempered by the fact that while a surface is represented, in fact there is data only at the points of each projection. For example projection k and k+1 represent discreet values that are interpolated in the three dimensional plot.

There are a number of switches that can be set to control the PPEXP analysis. The number of projections is set as `p`. While the default = 5, it is best to set `p` sufficiently large such that the projection pursuit index has fallen off. This can usually be detected by a graph of this index. The weight of each observation can be set by a vector $w$ although the default of 1.0 is usually used. The first step in PPEXP analysis is to sphere the data by performing a linear transformation that removes all, location, scale and correlation structure. Using Friedman's (1987) notation, assume $Y$ is a random variable in $R^p$ and define

$$\Sigma = E[(Y-EY)(Y-EY)^{'}] = UDU' \tag{17.4-1}$$

where $\Sigma$ is of rank $q$. The components of $Z$ are

$$Z_j = (1/\sqrt{D_j})\sum_{i=1}^{p} U_{ij}(Y_i - EY_i) \qquad 1 \le j \le q \tag{17.4-2}$$

The combinations $X_1 = \alpha' Z$ and $X_2 = \beta' Z$ have variance $\alpha'\alpha$ and $\beta'\beta$ respectively. If the constraint $\alpha'\alpha = \beta'\beta = 1$ is enforced, all combinations have unit variance. If in addition $\alpha'\beta = 0$ is imposed, the correlation of $X_1$ and $X_2$ will be zero.[7]

Define $\Phi(X_i)$ as the standard normal cdf defined over the range $[-1,1]$. The transformations $R_1 = 2\Phi(X_1) - 1$ and $R_2 = 2\Phi(X_2) - 1$ transform $X_1$ and $X_2$. The insight underlying the exploratory projection pursuit index is that if $X_1$ and $X_2$ have a joint standard normal distribution, then $R_1$ and $R_2$ will be uniformly distributed on the square $(-1, 1)\times(-1, 1)$. An important switch is the order of the Legendre polynomial expansion $J$ where

$$\begin{aligned} P_0(R_i) &= 1 \\ P_1(R_i) &= R_i \\ P_j(R_i) &= [(2j-1)R_i P_{j-1}(R_i) - (j-1)P_{j-2}(R_i)]/j \quad \text{for } j \ge 2 \end{aligned} \tag{17.4-3}$$

Coefficients $a_j$ and $b_j$ are given by

7 Friedman (1987,255-256) outlines the optimization strategy to insure these constraints.

$$a_j = \frac{(2j+1)}{2}\int_{-1}^{1} P_j(R_1)p_{R_1}(R_1)dR_1 = \frac{(2j+1)}{2}E_{R_1}[P_j(R_1)]$$
$$b_j = \frac{(2j+1)}{2}\int_{-1}^{1} P_j(R_2)p_{R_2}(R_2)dR_2 = \frac{(2j+1)}{2}E_{R_2}[P_j(R_2)] \qquad (17.4\text{-}4)$$

Typical values for the maximum Legendre polynomial $J$ is $2 \le J \le 8$ with 2 as the default. A larger value results in more smoothing. Friedman suggests increasing $J$ as the sample size increases. The bivariate projection index is defined as

$$I(\alpha,\beta) = \sum_{j=1}^{J}(2j+1)E^2[P_j(R_1)]/4 + \sum_{j=1}^{J}(2j+1)E^2[P_j(R_2)]/4$$
$$+\sum_{j=1}^{J}\sum_{k=1}^{J-j}(2j+1)(2k+1)\times E^2[P_j(R_1)P_k(R_2)]/4 \qquad (17.4\text{-}5)$$

where sample averages are used for the expectations. Sphering robustification is controlled by the trimming threshold $\tau$. Observations are ignored in the Sphering calculation (only) if their Mahalanobis[8] distance from the mean is greater than $\tau$. A value of $\tau = 0$ indicates no robustification. The maximum dimensionality of search space can be controlled by the parameter $q$ which defaults to the number of right hand side variables. Solutions are constrained to lie in the subspace spanned by those eigenvectors of the covariance matrix whose eigenvalues are larger than $\theta$ times the largest eigenvalue. A typical value for $\theta = .001$ although the default is 1.0 which limits the search. Since numerical optimization is needed to obtain estimates of $\alpha_m$ and $\beta_n$ the number of iterations must be set. The default # of iterations is 200. The convergence threshold $\xi = .01$. Maximization 'converges' when improvement from the previous iteration is less than $\xi$ times its value at the current iteration. Output can be adjusted by varimax rotation. If this is not done (:novarimax) the linear combinations and associated adjusted data plots are rotated so that the variance of the original (unsphered) variable loadings on the vertical solution coordinate (ordinate) is maximized. Friedman suggests that this sometimes helps in interpreting the solutions. In the numerical optimization phase the derivatives of the bivariate index are:

---

8 The Mahalanobis distance from a group of values with mean $\mu = (\mu_1, \mu_2, \ldots$ for a covariance matrix $S$ and a multivariate vector $x = (x_1, x_2, \ldots$

is $D_M(x) = \sqrt{(x-\mu)' S^{-1}(x-\mu)}$ .

$$
\begin{aligned}
\frac{\partial I}{\partial \alpha_m} = {} & (1/\sqrt{2\pi})\sum_{j=1}^{J}(2j+1)E[P_j(R_1)] \\
& \times E[P_j^{'}(R_1)e^{-(1/2)X_1^2}(Z_m - \alpha_m X_1 - \beta_m X_2)] \\
& + (1/\sqrt{2\pi})\sum_{j=1}^{J}\sum_{k=1}^{J-j}(2j+1)(2k+1) \\
& \times E[P_j(R_1)P_k(R_2)] \\
& \times E[P_j^{'}(R_1)P_k(R_2)e^{-(1/2)X_1^2}(Z_m - \alpha_m X_1 - \beta_m X_2)]
\end{aligned} \tag{17.4-6}
$$

$$
\begin{aligned}
\frac{\partial I}{\partial \beta_n} = {} & (1/\sqrt{2\pi})\sum_{k=1}^{J}(2k+1)E[P_k(R_2)] \\
& \times E[P_k^{'}(R_2)e^{-(1/2)X_2^2}(Z_n - \alpha_n X_1 - \beta_n X_2)] \\
& + (1/\sqrt{2\pi})\sum_{j=1}^{J}\sum_{k=1}^{J-j}(2j+1)(2k+1) \\
& \times E[P_j(R_1)P_k(R_2)] \\
& \times E[P_j(R_1)P_k^{'}(R_2)e^{-(1/2)X_2^2}(Z_n - \alpha_n X_1 - \beta_n X_2)]
\end{aligned} \tag{17.4-7}
$$

As noted, the user must set $J$ as the order of the polynomial expansion of the density in (17.4-5).

The bivariate projection index $I(\alpha, \beta)$ defined in (17.4-5) measures departure from normality. After each iteration, the non-normal structure is removed from the data and estimation of another projection pursuit term proceeds. The process is terminated when the projection pursuit algorithm cannot find a projection that deviates substantially from normality. Friedman (1987) estimates the multivariate density approximation as

$$
\tilde{\imath} \qquad Z)\prod_{k=1}^{K}[\sum_{j=0}^{J}(2j+1)\times E_{k-1}[P_{jk}]P_j(2\Phi(\alpha_k^T Z)-1)]/2 \tag{17.4-8}
$$

where $E_{k-1}[P_{jk}]$ is the expected value of the associated (adjacent) Legendre polynomial under $p_{k-1}(Z)$.[9]

A useful feature of exploratory projection pursuit is that by graphical means it is possible to determine where the nonlinearity is present in the data. This is done by analysis of the two 3-D plots of the projections. An example of the capability is shown next using a simulated dataset

9 This discussion of the theory follows very closely the discussion in Friedman (1987) with sections of sentences lifted as needed.

with 1000 observations. Model one was $y1 = 10 + 5x + 5z + 20e$ while model two added a term $5|x|^3$ for the last 50% of the sample. In the reported case both *x* and *z* were rectangularly distributed in the range 0 to 1. The error term was iid(0,1) and was multiplied by 20 to provide noise. The absolute value was given in the equation to allow using normally distributed right hand side variables without a code change.

Table 17.2 Detection of Nonlinearity in last 50% of sample

---

```
/;
/; The data y1 is 100% linear
/; The data y2 is set so that first 50% is nonlinear
/;
%b34slet noob   = 1000;
%b34slet noise  = 20.;
%b34slet nonlin = 5.;

b34sexec data noob=%b34seval(&noob)$
build y1 y2 x z e1 noise$
gen noise=%b34seval(&noise);
gen e1=rn()$

/; turn on one or the other pair to generate x and z

gen x =10*rn()$
gen z =10*rn()$

gen x =10*rec()$
gen z =10*rec()$

gen y1= 10 + 5*x + 5*z + noise*e1 $
gen y2=y1;

gen if(kount().gt.(%b34seval(&noob)/2.))
      y2=y1+%b34seval(&nonlin)*(abs(x)**3);
b34srun$

b34sexec matrix;
call echooff;
call loaddata;
call load(ppexp_p);

/; sets number of projections
mm=5;

/; sets order of legenre.  Larger => smoother

jj=2;

fei=.1e-4;
   fei=1.   ;

nei = 1  ;
/; nei = 2  ;

trm=.1;
/; trm=.8;

mod1=1;
mod2=1;
```

```
if(mod1.ne.0)then;
call ppexp(y1 x z :mm mm :jj jj :fei fei
                                :nei nei
                                :trm trm :print);
ppi_m1=%ppindex;
call ppexp_p(%xpa,%mm,%nob,0,'a',%ppindex);
call dodos('copy ppexp_1.wmf model_1a.wmf',:);
call dodos('copy ppexp_2.wmf model_1b.wmf',:);
call dodos('copy ppindex.wmf ppindex1.wmf',:);
endif;

if(mod2.ne.0)then;
call ppexp(y2 x z :mm mm :jj jj :fei fei
                                :nei nei
                                :trm trm print);
ppi_m2=%ppindex;
call ppexp_p(%xpa,%mm,%nob,0,'b',%ppindex);
call dodos('copy ppexp_1.wmf model_2a.wmf':);
call dodos('copy ppexp_2.wmf model_2b.wmf':);
call dodos('copy ppindex.wmf ppindex2.wmf':);
call ppexp_p(%xpa,%mm,%nob,1    ,'b',%ppindex);
endif;

b34srun$
```

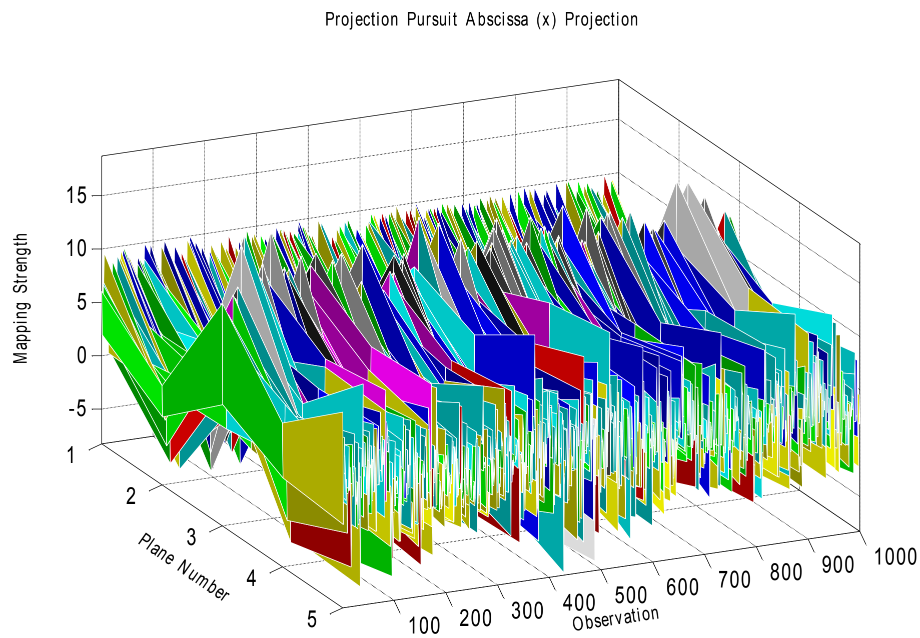
The exploratory projection pursuit index for model 1 was

```
      1              2              3              4              5
0.163213E-01   0.224295E-03   0.768322E-04   0.929875E-04   0.408952E-04
```

which contained only one large term suggesting a simple model. For the nonlinear model 2 the index was

```
      1              2              3              4              5
0.380199E-01   0.351555E-01   0.382482E-02   0.344095E-03   0.533835E-04
```

with two large terms and one moderately large term.




‘

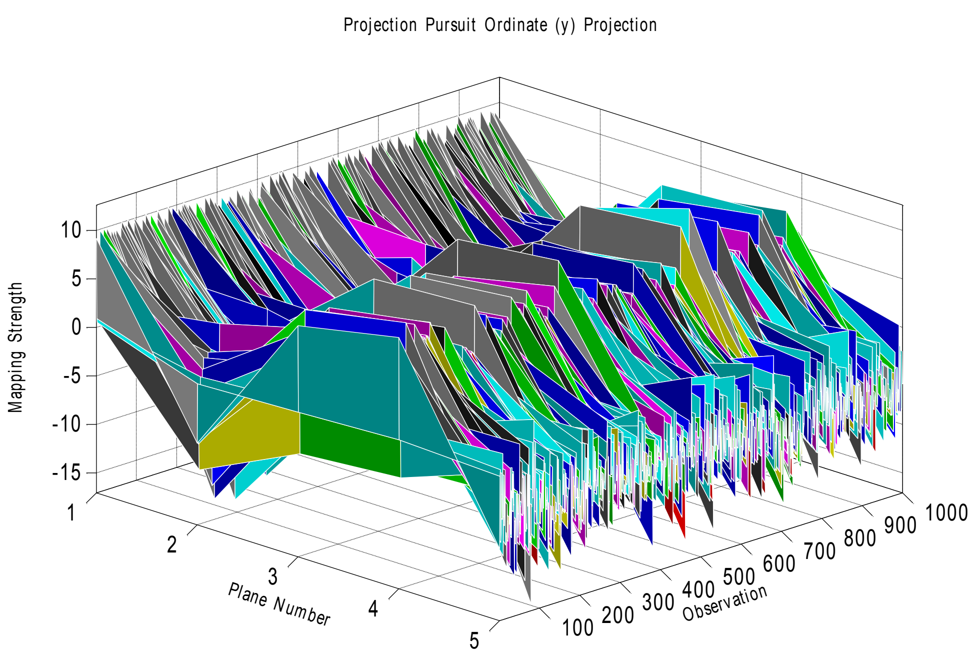



Figure 17.1 x and y projections for linear Model 1

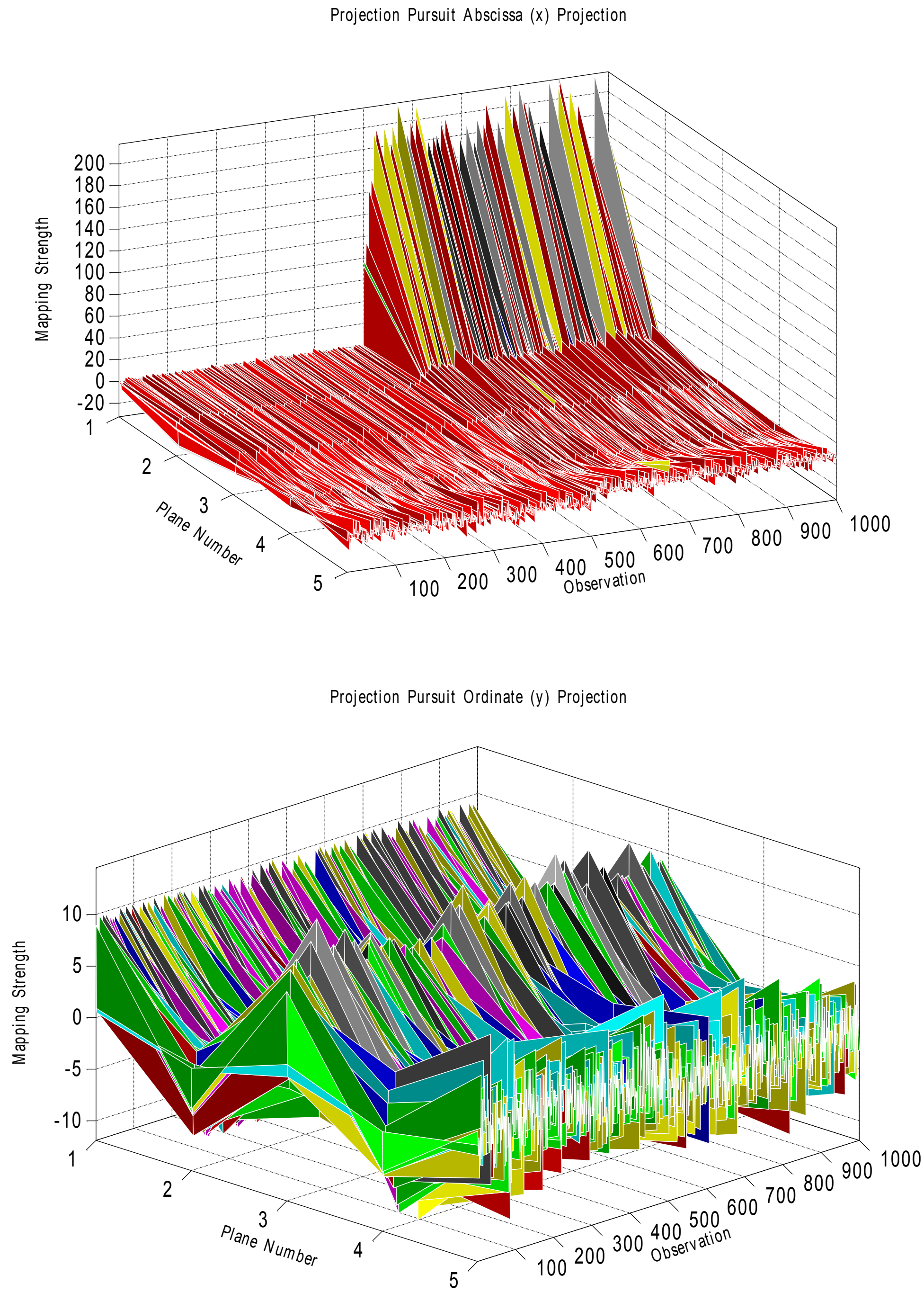



Figure 17.2 x and y for non-linear model 2

Figure 17.1 shows the *x* and *y* projections for the linear model 1. As expected, they are relatively flat. Figure 17.2, on the other hand, shows the effect of nonlinearity in the last 50% of the sample. This is most dramatic in the x projection slice graphs shown in Figures 17.3, 17.4 and 17.5. Note that the nonlinear effect that is a function of the observation shows less in projection 2 than in 1 and still less in projection 3 than 2. Projections 4 and 5 show much smaller nonlinear effects and are not shown except in Figure 17.2.

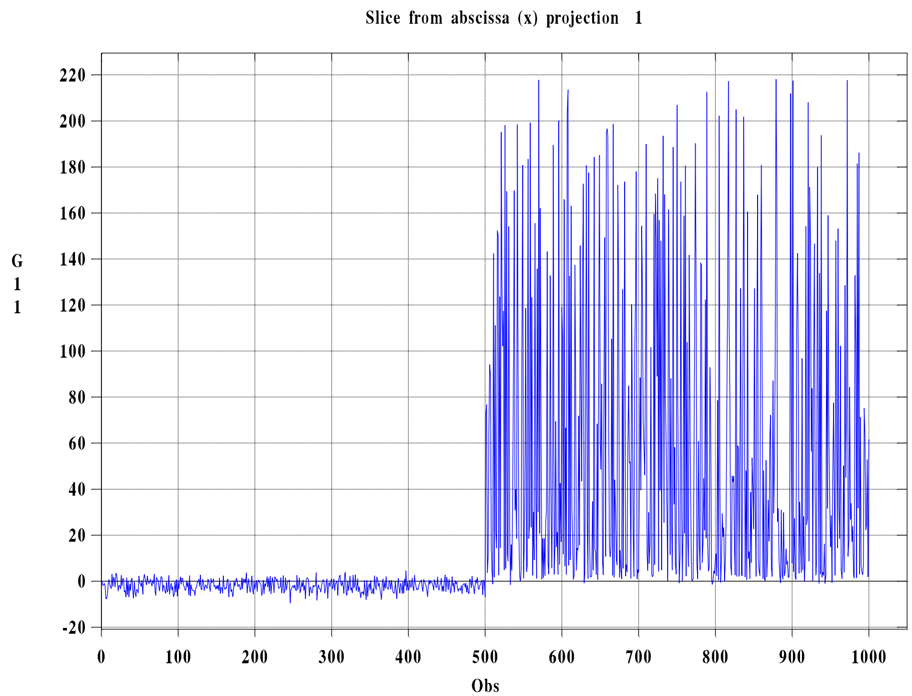



Figure 17.3 Slice from abscissa (x) for projection 1

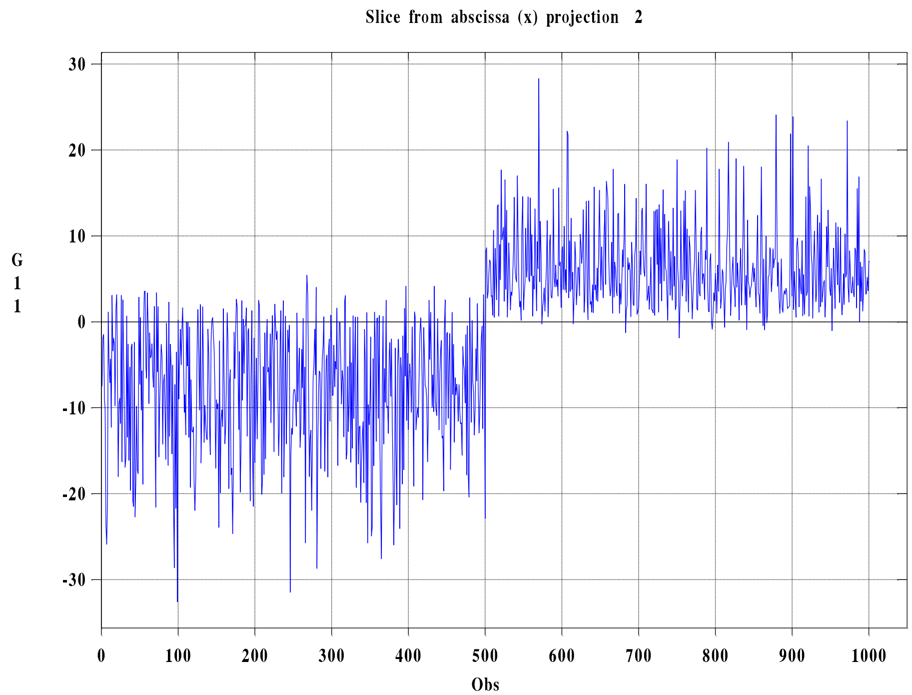



Figure 17.4 Slice from abscissa (x) for projection 2

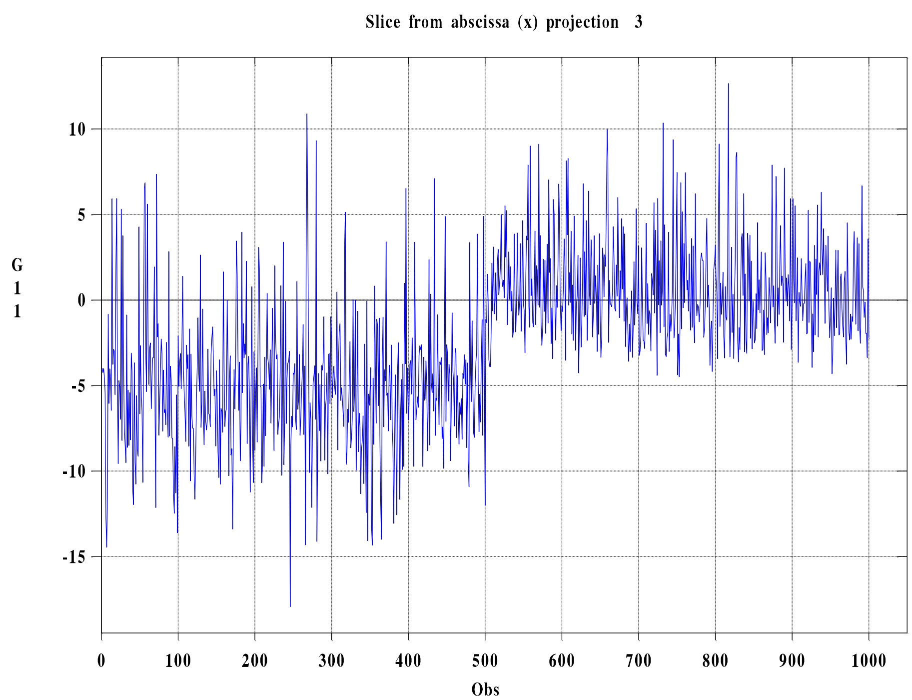



Figure 17.5 Slice from abscissa (x) for projection 3

## 17.5 Random Forest

For classification problems, the Random Forest approach[10] uses both bagging and voting to improve the performance of a CART type model. For continuous left hand side variable problems, bagging and averaging of the predictions of many models are used to determine $\hat{y}$. Unlike a least squares model that requires explicit modeling of interactions and or nonlinearities, a random forest model is non parametric and allows interactions and nonlinearities to be learned from the data. For non-continuous left hand side variable problems, bagging and voting is used to form the expected classification $\hat{y}$.

At the heart of the random forest method is the concept of bagging which facilitates model validation using the training dataset which is a subset of the dataset to estimate the model which is then validated using the out-of-bag observations that were not used to estimate that model. The basic idea behind bagging is to build a large collection of de-correlated trees and then to use voting or averages to develop the consensus value for $\hat{y}$. Assume there are N observations and $p$ potential right hand side variables in the original or training dataset. Assume a maximum of B trees will be estimated. The procedure is for $b = 1$, B

1. Draw a bootstrap sample $Z^*$ of size N.

2. Grow a random-forest tree $T_b$ by repeating the following steps for each node until the a minimum node size of $n_{\min}$ is reached.

   2.1. Select $m$ variables at random from the $p$ right hand side variables where $m \le p$.
   2.2 Pick best variable/split point among the $m$.
   2.3 Split the node into two sub- nodes.

3. Save the ensemble of trees $\{T_b\}_1^B$.

For a regression random forest model at new point $x$, $\hat{y} = \hat{f}_{rf}^B$, which is obtained by averaging

$$\hat{f}_{rf}^B = \frac{1}{B}\sum_{b=1}^{B} T_b(x) \qquad (17.5\text{-}1)$$

10 The name "RandomForests" is licensed by Salford systems as a trade mark for their version of the Leo Breiman and Adele Cutler original code. The B34S implementation uses code obtained from Adele Cutler's web page that is freely available and is explicitly released under the GNU General Public License. This code has been modified in a number of places to implement BLAS routines and fix a number of "bugs" that showed up. It can be run stand alone. The discussion of the Random Forest procedure follows closely the excellent treatment in Hastie-Tibshirani-Friedman (2009, Chapter 15).

For a classification problem define the class prediction of the $b^{th}$ random forest tree as $\hat{C}_b(x)$. For all such B models the prediction is based on the majority vote or

$$\hat{C}_{rf}^{B}(x) = majority\ vote\{\hat{C}_b(x)\}_1^B \tag{17.5-2}$$

In the B34S implementation B is set by `:maxtree`, $m$ is set by `:mtry` and $n_{\min}$ is set by `:ndsize`. Tables 17.3 and 17.4 illustrate the sensitivity of the results to these settings for a continuous data example using real data, and two simulation examples. The number of variables selected randomly, *m,* varied between 3-10 while the minimum node size, $n_{\min}$, varied between 1 to *m*. Since each model estimated had 200 trees, a total of 10,400 trees (52*200) were estimated, An OLS model was estimated to provide a reference sum of squares of 11,297.7549.

Table 17.3 Random Forest Tests on the Boston Housing Data

---

```
/;
/; Test of RF on REG data.
/; See Hastie-Tibshirani-Friedman (2009, 587-604)
/;
b34sexec options ginclude('b34sdata.mac') member(bostonh);
         b34srun;
/; b34sexec list; b34srun;

b34sexec matrix;
call loaddata;
call echooff;

call olsq(medv crim  zn  indus nox rm age dis
rad tax ptratio b lstat :print);
olsres=%res;
olsyhat=%yhat;

maxtree=200;
ihold=0;

ii1=3;
ii2=10;

need=sum(integers(ii1,ii2));
_mtry  =array(need:);
_ndsize=array(need:);
_rss1  =array(need:);
_rss2  =array(need:);
_rss3  =array(need:);

icount=0;

do mtry=ii1,ii2;

do ndsize=1,mtry;
icount=icount+1;
call ranforest(medv crim  zn  indus nox rm age dis
rad tax ptratio b lstat
:imp :savex :savemodel :yhatav
:reg  :maxtree maxtree
/; print
:mtry mtry  :ndsize ndsize
```

```
:holdout ihold);
  _mtry(icount)=mtry;
_ndsize(icount)=ndsize;
  _rss1(icount)=%rss;
  _rss2(icount)=%rss2;
  _rss3(icount)=%rss3;
enddo;

enddo;
call print(' ':);
call tabulate(_mtry,_ndsize,_rss1,_rss2,_rss3
:title 'Random Forest results for various settings');

b34srun;
```

Results obtained were:

```
           Boston Housing Data              PAGE    1

Variable    Label                                      # Cases      Mean        Std. Dev.    Variance      Maximum       Minimum

CRIM      1 per capita crime rate by town                506   3.61352       8.60155       73.9866       88.9762   0.632000E-02
ZN        2 prop. resid land zoned> 25,000 sq f          506   11.3636       23.3225       543.937       100.000       0.00000
INDUS     3 prop. non-retail business acres per town     506   11.1368       6.86035       47.0644       27.7400      0.460000
CHAS      4 Charles River dummy 1 if tract on river      506  0.691700E-01  0.253994     0.645130E-01   1.00000       0.00000
NOX       5 nitric oxides concentration(pp 10 mill.      506  0.554695      0.115878     0.134276E-01  0.871000      0.385000
RM        6 average number of rooms per dwelling         506   6.28463      0.702617      0.493671      8.78000       3.56100
AGE       7 prop. owner-occupied units built < 1940      506   68.5749       28.1489       792.358       100.000       2.90000
DIS       8 weighted dist. 5 Boston employment cent.     506   3.79504       2.10571       4.43402       12.1265       1.12960
RAD       9 index of accessibility radial highways       506   9.54941       8.70726       75.8164       24.0000       1.00000
TAX      10 full-value property-tax rate per $10,000     506   408.237       168.537       28404.8       711.000       187.000
PTRATIO  11 pupil-teacher ratio by town                  506   18.4555       2.16495       4.68699       22.0000       12.6000
B        12 1000(Bk - 0.63)^2. Bk=proportion blacks      506   356.674       91.2949       8334.75       396.900      0.320000
LSTAT    13 % lower status of the population             506   12.6531       7.14106       50.9948       37.9700       1.73000
MEDV     14 Median value occupied homes in $1000         506   22.5328       9.19710       84.5867       50.0000       5.00000
CONSTANT 15                                              506   1.00000       0.00000       0.00000       1.00000       1.00000

Number of observations in data file   506
Current missing variable code         1.000000000000000E+31

B34S(r) Matrix Command. d/m/y  9/ 4/09. h:m:s  8:30:47.

=>  CALL LOADDATA$

=>  CALL ECHOOFF$

Ordinary Least Squares Estimation
Dependent variable                       MEDV
Centered R**2                            0.7355165089722999
Adjusted R**2                            0.7290787769391713
Residual Sum of Squares                  11297.75493513496
Residual Variance                        22.91633861082143
Standard Error                           4.787101274343528
Total Sum of Squares                     42716.29541501976
Log Likelihood                           -1503.756025055460
Mean of the Dependent Variable           22.53280632411067
Std. Error of Dependent Variable         9.197104087379817
Sum Absolute Residuals                   1665.528771275704
F(12,      493)                          114.2508736286829
F Significance                           1.000000000000000
1/Condition XPX                          3.461932057380833E-09
Maximum Absolute Residual                26.37545629480741
Number of Observations                   506

Variable   Lag    Coefficient         SE              t
CRIM        0    -0.11313908         0.33112989E-01   -3.4167582
ZN          0     0.47052458E-01     0.13846883E-01    3.3980542
INDUS       0     0.40311454E-01     0.61707445E-01   0.65326727
NOX         0    -17.366999           3.8512241       -4.5094751
RM          0     3.8504917          0.42140201        9.1373358
AGE         0     0.27837565E-02     0.13308964E-01   0.20916403
DIS         0    -1.4853739          0.20118679       -7.3830587
RAD         0     0.32831101         0.66542335E-01    4.9338667
TAX         0    -0.13755829E-01     0.37657000E-02   -3.6529275
PTRATIO     0    -0.99095803         0.13139909       -7.5415897
B           0     0.97414509E-02     0.27060574E-02    3.5998686
LSTAT       0    -0.53415762         0.51071610E-01   -10.458993
CONSTANT    0      36.891960          5.1465158        7.1683371

Random Forest Results for various settings

Obs      _MTRY       _NDSIZE     _RSS1       _RSS2       _RSS3
```

```
     1            3          1   0.3234E+05   7593.       3210.
     2            3          2   0.3511E+05   5838.       1526.
     3            3          3   0.1568E+05   5497.       1356.
     4            4          1   0.4412E+05   7732.       3164.
     5            4          2   0.2528E+05   5796.       1558.
     6            4          3    9765.       5319.       1321.
     7            4          4   0.1530E+05   5337.       1326.
     8            5          1   0.2604E+05   8037.       3295.
     9            5          2   0.2038E+05   5834.       1582.
    10            5          3   0.1442E+05   5565.       1335.
    11            5          4   0.1483E+05   5429.       1357.
    12            5          5   0.1196E+05   5206.       1354.
    13            6          1   0.3326E+05   7679.       3102.
    14            6          2   0.2358E+05   5829.       1529.
    15            6          3   0.1804E+05   5432.       1287.
    16            6          4    8879.       5550.       1401.
    17            6          5   0.3106E+05   5405.       1460.
    18            6          6   0.1315E+05   5663.       1612.
    19            7          1   0.2953E+05   7415.       3099.
    20            7          2   0.1168E+05   6004.       1631.
    21            7          3   0.1458E+05   5535.       1375.
    22            7          4   0.1428E+05   5399.       1358.
    23            7          5   0.3090E+05   5600.       1518.
    24            7          6   0.1891E+05   5287.       1578.
    25            7          7    8499.       5384.       1696.
    26            8          1   0.2727E+05   7537.       3170.
    27            8          2   0.2143E+05   5772.       1539.
    28            8          3   0.2246E+05   5369.       1311.
    29            8          4   0.1792E+05   5267.       1298.
    30            8          5   0.1648E+05   5500.       1498.
    31            8          6   0.1266E+05   5707.       1619.
    32            8          7   0.1358E+05   5527.       1821.
    33            8          8    9463.       5919.       1996.
    34            9          1   0.2118E+05   7433.       3072.
    35            9          2   0.1621E+05   5895.       1633.
    36            9          3   0.2420E+05   5391.       1385.
    37            9          4   0.1066E+05   5348.       1372.
    38            9          5   0.2303E+05   5531.       1468.
    39            9          6   0.1919E+05   5128.       1524.
    40            9          7   0.1399E+05   5642.       1753.
    41            9          8   0.3256E+05   5750.       1952.
    42            9          9   0.1967E+05   5532.       2022.
    43           10          1   0.2231E+05   7489.       3066.
    44           10          2   0.1279E+05   6093.       1583.
    45           10          3   0.2316E+05   5279.       1270.
    46           10          4   0.3385E+05   5543.       1436.
    47           10          5   0.2005E+05   5407.       1402.
    48           10          6    9267.       5345.       1571.
    49           10          7   0.3190E+05   5617.       1764.
    50           10          8   0.3386E+05   5675.       1937.
    51           10          9   0.1433E+05   5807.       2067.
    52           10         10   0.2128E+05   5173.       2133.

 B34S Matrix Command Ending. Last Command reached.

 Space available in allocator   11856601, peak  space  used      91309
 Number variables used                87, peak number  used        128
 Number temp variables used         2420, # user temp clean          0
```

The estimated sum of squares for the average of models varied between a minimum of 1270.09 for `MTRY=10` and `NDSIZE=3` and a maximum of 3295 for `NTRY=5` and `NDSIZE=3`, both of which are substantially under the OLS benchmark of `11298.` `_RSS1`, `_RSS2` and `_RSS3` are respectively the residual sum of squares for the last model, for averaged out of bag samples and and for the average of all predictions.

Table 17.4 shows an automatic method to compare the leverage plots of alternative estimates of this model. Edited output is shown below the Table.

Table 17.4 Leverage plots of alternative models of Boston Housing Data

```
/;
/; OLS, MARSPLINE, GAM, PPREG, RF On Boston Housing Data
/; See Hastie-Tibshirani-Friedman (2009, 587-604)
/;
/; Illustrates different types of forecasts
/;
b34sexec options ginclude('b34sdata.mac') member(bostonh);
         b34srun;
/; b34sexec list; b34sr

b34sexec matrix;
call loaddata;
call echooff;
call load(contrib);

/; start ------------------------------------------
call contribi;
/;
/; specific settings
/;
_mi=2;
_m=30;
iols=4;
isave=1;
_mtry=4;
_mtree=20;

call character(fsv_info,'bostonh Test Case');
call character(l_hand_s,'medv');
call character(_args,
'crim  zn  indus nox rm age dis rad tax ptratio b lstat');
_argsg=_args;
call contribl;
call contribd;

b34srun;
```

```
Multivariate Autoregressive Splines Analysis
 Model Estimated using Hastie-Tibshirani GPL routines in
 CRAN General Public License (GPL) Library.
 Version - 1 March 2006.

 Left Hand Side Variable                          MEDV
 Penalty cost per degree of freedom         3.000
 Threshold for Forward stepwise Stopping   0.1000E-03
 Rank Test Tolerance                       0.1000E-12
 Max # of Knots  (nk)                       16
 Max interaction (mi)                       2
 Number of Observations                     506
 Number of right hand Variables             12
 tolbx    set as                            1.000000000000000E-09
 stopfac  gcv/gcvnull > stopfac => stop     10.0000000000000
 prevcrit set as                            10000000000.00000

 Series   Lag  Mean          Max          Min
 CRIM      0      3.614        88.98        0.6320E-02
 ZN        0      11.36        100.0         0.000
 INDUS     0      11.14        27.74        0.4600
 NOX       0     0.5547       0.8710        0.3850
 RM        0      6.285        8.780         3.561
 AGE       0      68.57        100.0         2.900
 DIS       0      3.795        12.13         1.130
 RAD       0      9.549        24.00         1.000
```

```
TAX       0      408.2         711.0         187.0
PTRATIO   0      18.46         22.00         12.60
B         0      356.7         396.9        0.3200
LSTAT     0      12.65         37.97         1.730

GCV with only the constant               84.75422205666118
Total sum of squares                     42716.29541501979
Final gcv                                10.85523887241490
Variance of Y Variable                   84.58672359409854
R**2 (1 - (var(res)/var(y)))             0.8878759149406787
Residual Sum of Squares                  4789.525540532765
Residual Variance                        9.484208991154006
Residual Standard Error                  3.079644296205977
Sum Absolute Residuals                   1150.079657209633
Max Absolute Residual                    13.20050331160597
# of coefficients after last fwd step    14

MARS Model Coefficients                                       SE                t      Non Zero    %     Importance  #

MEDV      =        25.704643                                0.45734108          56.2     506   100.000               1
   -0.93932972    * max(   LSTAT{  0}    -    6.0700000    , 0.0)  0.44285026E-01     -21.2     410    81.028      95.548   2
+    1.8445262    * max(   6.0700000     -    LSTAT{  0}   , 0.0)  0.24589809          7.50      95    18.775      33.790   3
+    13.391331    * max(      RM{  0}    -    6.4250000    , 0.0)  0.60322928          22.2     178    35.178     100.000   4
    -2.7757310    * max(   6.4250000     -       RM{  0}   , 0.0)  0.61980591         -4.47     327    64.625      20.173   5
   -0.56095199    * max(     DIS{  0}    -    1.4395000    , 0.0)  0.86750263E-01     -6.46     482    95.257      29.128   6
+    104.54672    * max(   1.4395000     -      DIS{  0}   , 0.0)   7.2772620         14.3      23     4.545      64.714   7
   -0.57802209    * max(      RM{  0}    -    6.4250000    , 0.0)  0.54336221E-01     -10.6     165    32.609      47.920   8
                  * max(     RAD{  0}    -    1.0000000    , 0.0)
+    2.1255996    * max(     NOX{  0}    -   0.71800000    , 0.0)  0.45089067          4.71      37     7.312      21.236   9
                  * max(   LSTAT{  0}    -    6.0700000    , 0.0)
+    2.4059781    * max(  0.71800000     -      NOX{  0}   , 0.0)  0.21683312          11.1     368    72.727      49.983  10
                  * max(   LSTAT{  0}    -    6.0700000    , 0.0)
    -7.1078323    * max(    CRIM{  0}    -    4.8714100    , 0.0)  0.74650006         -9.52      18     3.557      42.891  11
                  * max(   1.4395000     -      DIS{  0}   , 0.0)
    -68.861720    * max(   4.8714100     -     CRIM{  0}   , 0.0)   14.661276         -4.69       5     0.988      21.158  12
                  * max(   1.4395000     -      DIS{  0}   , 0.0)
+   0.25316834    * max(   6.4250000     -       RM{  0}   , 0.0)  0.42740539E-01      5.92     272    53.755      26.683  13
                  * max(   LSTAT{  0}    -    9.0400000    , 0.0)
+    1.3873301    * max(   6.4250000     -       RM{  0}   , 0.0)  0.34135130          4.06      54    10.672      18.308  14
                  * max(   9.0400000     -    LSTAT{  0}   , 0.0)

Analysis of GCV, RSS and KNOT by Variable before prune step

Obs      _GCV        _RSS        _KNOT       _VAR        _LAG
     1      28.36     0.1401E+05   6.070     LSTAT                0
     2      19.91       9643.      6.425     RM                   0
     3      17.97       8525.      1.440     DIS                  0
     4      15.46       7262.      1.000     RAD                  0
     5      13.32       6126.     0.7180     NOX                  0
     6      11.54       5201.      4.871     CRIM                 0
     7      10.86       4790.      9.040     LSTAT                0


Generalized Additive Models (GAM) Analysis
Reference: Generalized Additive Models by Hastie and Tibshirani. Chapman (1990)
Model estimated using CRAN General Public License (GPL) routines.

Gaussian additive model assumed
Identity link - yhat = x*b + sum(splines)

Response variable ....      MEDV
Number of observations:     506
Residual Sum of Squares     7491.667403904000
# iterations                1
# smooths/variable          30
Mean Squared Residual       14.80566680613439
df of deviance              480.9967219912970
Scale Estimate              15.57529825336222
Primary   tolerence         1.000000000000000E-09
Secondary tolerance         1.000000000000000E-09
R square                    0.8246180449143116
Total sum of Squares        42716.29541501979

Model  df       coef        st err     z score      nl pval      lin_res       Name     Lag
------------    ----        ------     -------      -------      -------       ----     ---
        1.      39.1267      4.243       9.222                                 intcpt
        2.00   -.173857     0.2730E-01  -6.369       0.9295        7574.       CRIM      0
        2.00   0.134731E-01  0.1142E-01  1.180       0.9177        7569.       ZN        0
        2.00   0.296049E-01  0.5087E-01 0.5819       0.7513        7535.       INDUS     0
        2.00   -20.1537      3.175      -6.348       0.7230        7532.       NOX       0
        2.00    3.46435     0.3474       9.972        1.000        9244.       RM        0
        2.00   0.405577E-02  0.1097E-01 0.3696       0.7020        7529.       AGE       0
        2.00   -1.27843     0.1659      -7.708       0.9985        7693.       DIS       0
        2.00   0.345222     0.5486E-01   6.293       0.6822        7527.       RAD       0
        2.00   -.127505E-01  0.3104E-02 -4.107       0.8532        7551.       TAX       0
        2.00   -.842441     0.1083      -7.777       0.9174        7569.       PTRATIO   0
        2.00   0.730865E-02  0.2231E-02  3.276       0.8772        7557.       B         0
        2.00   -.602103     0.4210E-01  -14.30        1.000        8473.       LSTAT     0
       -----
        25.0

Projection Pursuit Regression

Number of Observations                        506
Number of right hand side variables            13
Maximum number of trees                        30
```

```
Minimum number of trees                        30
Number of left hand side variables             1
Level of fit                                   2
Max number of Primary   Iterations (maxit)     200
Max number of Secondary Iterations (mitone)    200
Number of cj Iterations (mitcj)                10
Smoother tone control (alpha)                  0.000000000000000E+00
Span                                           0.000000000000000E+00
Convergence (CONV) set as                      5.000000000000000E-03
Left Hand Side Variable                        MEDV

Series           Mean          Max           Min
MEDV            22.53         50.00         5.000

Right Hand Side Variables

#   Series    Lag     Mean          Max           Min
  1 CRIM        0    3.614         88.98        0.6320E-02
  2 ZN          0    11.36         100.0         0.000
  3 INDUS       0    11.14         27.74        0.4600
  4 NOX         0   0.5547        0.8710        0.3850
  5 RM          0    6.285         8.780         3.561
  6 AGE         0    68.57         100.0         2.900
  7 DIS         0    3.795         12.13         1.130
  8 RAD         0    9.549         24.00         1.000
  9 TAX         0    408.2         711.0         187.0
 10 PTRATIO     0    18.46         22.00         12.60
 11 B           0    356.7         396.9        0.3200
 12 LSTAT       0    12.65         37.97         1.730
 13 CONSTANT    0    1.000         1.000         1.000

Given # of trees                               30
# primary   iterations used                    2
# secondary iterations used                    4
# cj iterations used                           10
Residual sum of squares                        1844.310625018926
Total    sum of squares                        42716.29541501976
Mean of the Dependent Variable                 22.53280632411067
Std. Error of Dependent Variable               9.197104087379817
Sum Absolute Residuals                         667.0448505035151
Maximum Absolute Residual                      11.18610328018268
Residual Variance                              3.740995182594170

Variable Importance for Model with # Trees  30
Series Number       Importance
        9            1.00000
        8           0.926321
       12           0.796980
        2           0.768418
        7           0.729669
        1           0.684751
        5           0.656946
        6           0.512240
        4           0.500845
       10           0.485334
        3           0.426597
       11           0.259527
       13           0.142314E-15

Random Forest Analysis Ver. 3.1 - 30 May 2009 build
Regression option selected.

Number of Observations                            506
Number of right hand side variables               12
Maximum number of trees (maxtree)                 20
Maximum number of nodes (nrnodes)                 203
Number of Variables to select at each node (mtry)  4
Minimum node size (ndsize)                        5
Left Hand Side Variable                        MEDV

Series           Mean          Max           Min
MEDV            22.53         50.00         5.000

Right Hand Side Variables

  # Series   Lag      Mean          Max           Min
  1 CRIM       0     3.614         88.98        0.6320E-02
  2 ZN         0     11.36         100.0         0.000
  3 INDUS      0     11.14         27.74        0.4600
  4 NOX        0    0.5547        0.8710        0.3850
  5 RM         0     6.285         8.780         3.561
  6 AGE        0     68.57         100.0         2.900
  7 DIS        0     3.795         12.13         1.130
  8 RAD        0     9.549         24.00         1.000
  9 TAX        0     408.2         711.0         187.0
 10 PTRATIO    0     18.46         22.00         12.60
 11 B          0     356.7         396.9        0.3200
 12 LSTAT      0     12.65         37.97         1.730

Total Sum of Squares                            42716.29541501976
Sum of Squared Residuals for last bagged model   7847.284386971902
Sum of Squared Residuals for averaged OOB model  6202.295655814791
Sum of Squared Residuals for averaged model      1664.667401950134
Centered R**2 for %YHAT                         0.8162929553995765
Centered R**2 for %YHAT2                        0.8548025853938177
Centered R**2 for %YHAT3                        0.9610296870134293
```

```
Importance Analysis
For details see Hastie-Tibshirani-Friedman (2009, 594)

Variable importance based on Randomization
   1   107.41321
   2   94.653063
   3   112.44469
   4   126.98415
   5   265.37711
   6   97.205943
   7   118.84910
   8   103.23616
   9   121.84421
  10   120.86542
  11   98.873160
  12   306.35675

Variable importance based on Gini
   1   33429.893
   2   3717.3216
   3   44219.571
   4   92468.291
   5   275491.79
   6   24697.864
   7   45404.783
   8   6426.8150
   9   21639.139
  10   43568.732
  11   11508.983
  12   238332.61
```

The $e'e$ MARS was 4789.53, while for GAM it was 7491.67, both substantially under the OLS value of 11298. GAM found that RM, DIS and LSAT were highly nonlinear with significance of 1.00, .9985 and 1.00 respectively. If these variables were restricted to be be linear, $e'e$ would increase to 9244, 7693 and 8473 respectively. For Projection Pursuit $e'e$ substantially less at 1844.3. For the last bagged model Random Forest model $e'e$ was 7847.28, for the averaged out-of-bag model $e'e$ was 6202.296, while for the averaged model $e'e$ was 1664.667. Figures 17.6 and 17.7 show leverage plots for RM and LSAT which were found to be nonlinear by the GAM model and show how the various methods attempt to capture the nonlinearity in the variable. Both Figures assume the other data in the model are at their medians.

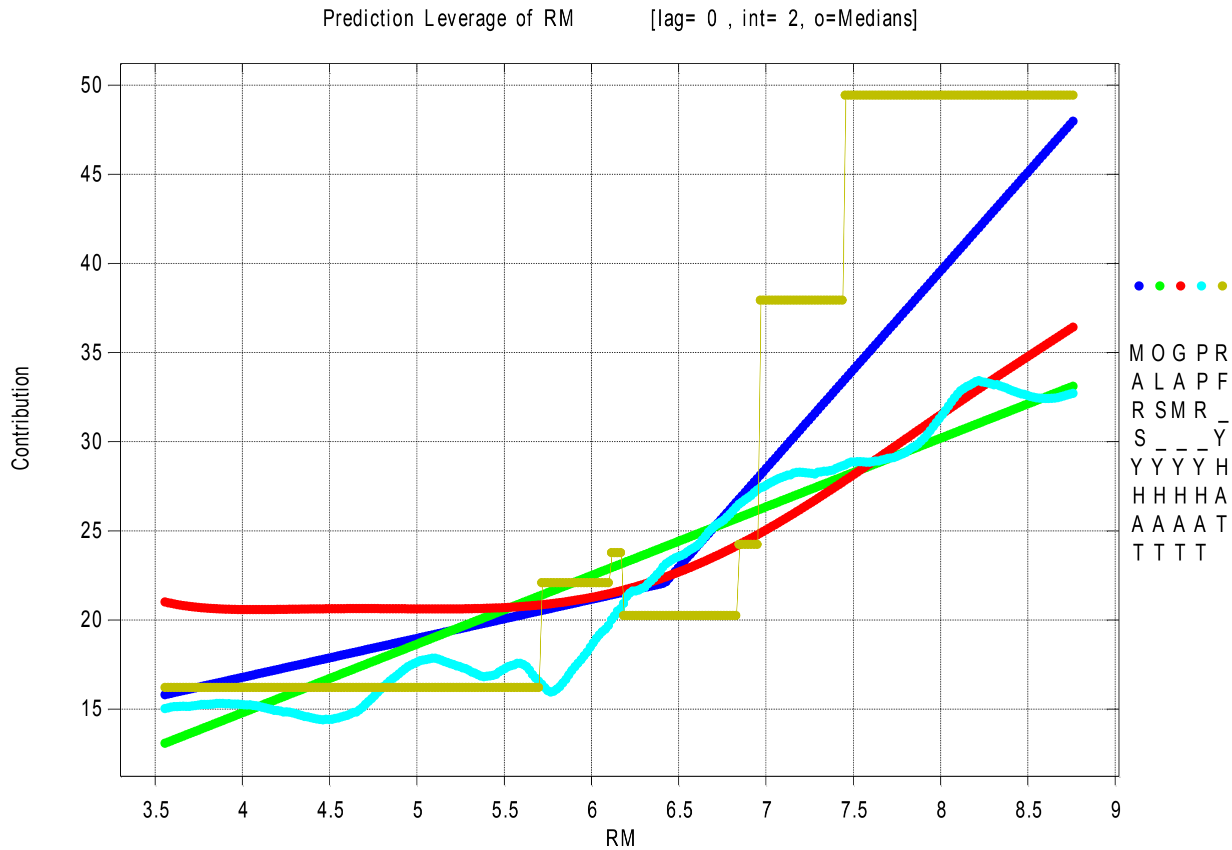



Figure 17.6 Leverage Plot for RM

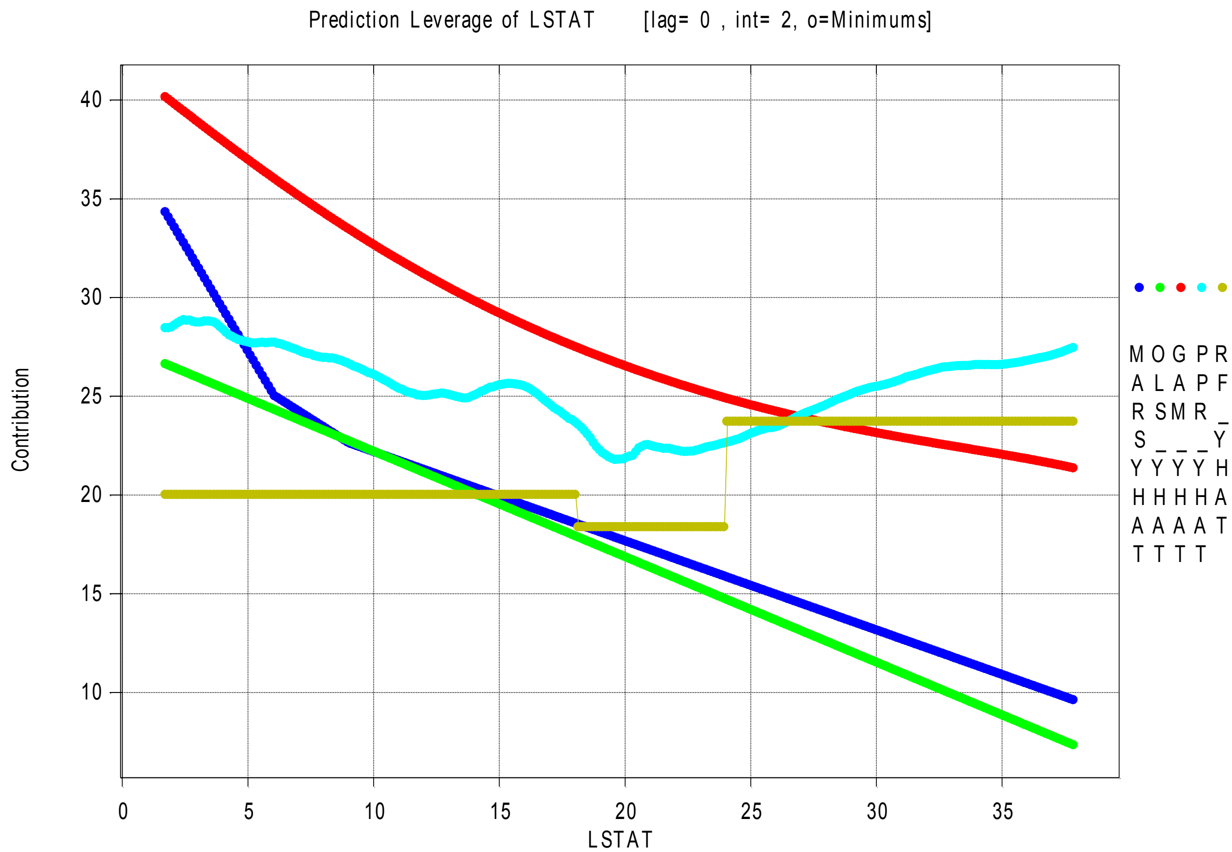



Figure 17.7 Leverage Plot for LSTAT

In the next problem a linear model and a nonlinear model are developed and tested using the setup listed in Table 17.5. First an OLS model is estimated where $e'e = 524.83$. Next increasingly more complex random forest models were estimated where as noted above $\_RSS1 = y - \hat{y}$, $\_RSS2 = y - \hat{y}_{oob}$ and $\_RSS3 = y - \hat{y}_{\text{using average}}$. In the linear case, while averaging gets close to OLS, the random forest method shows no gain as would be expected.

Table 17.5 Simulation Study of a linear and non-linear dataset

```
/;
/; Simulation study based on
/; Hastie-Tibshirani-Friedman (2009, 599)
b34sexec matrix;
call echooff;

n=500;
k=10;

/; y1 is linear
/; y2 is highly non-linear

x=rn(array(n,k:));
e=rn(array(n:));
y1=sumrows(    x )+e;
y2=sumrows(abs(x))+e;

/;  call print(y,x,e);

subroutine test(y,x,comment);
call print(comment:);
call olsq(y x :print);
olsres=%res;
olsyhat=%yhat;

maxtree=200;
ihold=0;

ii1=1;
ii2=10;

need=sum(integers(ii1,ii2));
_mtry  =array(need:);
_ndsize=array(need:);
_rss1  =array(need:);
_rss2  =array(need:);
_rss3  =array(need:);

icount=0;

do mtry=ii1,ii2;

do ndsize=1,mtry;
icount=icount+1;
call ranforest(y x
/; :imp
:savex :savemodel :yhatav
:reg  :maxtree maxtree
/; print
:mtry mtry  :ndsize ndsize
:holdout ihold);
  _mtry(icount)=mtry;
_ndsize(icount)=ndsize;
  _rss1(icount)=%rss;
  _rss2(icount)=%rss2;
  _rss3(icount)=%rss3;
enddo;

enddo;

call print(comment:);
```

```
call tabulate(_mtry,_ndsize,_rss1,_rss2,_rss3);
return;
end;
call test(y1,x,'Tests on Linear     model y=sum(x)+e');
call test(y2,x,'Tests of Non Linear model y=sum(abs(y))+e');

b34srun;
```

## Results

```
B34S(r) Matrix Command. d/m/y  9/ 4/09. h:m:s  9:45:52.

=>  CALL ECHOOFF$

Tests on Linear     model y=sum(x)+e

Ordinary Least Squares Estimation
Dependent variable                     Y
Centered R**2                          0.9064304528708895
Adjusted R**2                          0.9045169651995376
Residual Sum of Squares                524.8250232893058
Residual Variance                      1.073261806317599
Standard Error                         1.035983497126088
Total Sum of Squares                   5608.929821634533
Log Likelihood                         -721.5834715132283
Mean of the Dependent Variable         -0.2301173996548523
Std. Error of Dependent Variable       3.352661677520847
Sum Absolute Residuals                 405.6675331377948
F(10,      489)                        473.7059278936778
F Significance                         1.000000000000000
1/Condition XPX                        0.5441552376397775
Maximum Absolute Residual              3.391616654434103
Number of Observations                 500

Variable   Lag    Coefficient         SE              t
Col____1     0      1.0237249         0.44102832E-01    23.212226
Col____2     0     0.97141744         0.47942673E-01    20.262062
Col____3     0     0.92998929         0.45573189E-01    20.406500
Col____4     0      1.0181475         0.47930192E-01    21.242299
Col____5     0     0.92839937         0.46406082E-01    20.005985
Col____6     0     0.98858932         0.48893386E-01    20.219286
Col____7     0      1.0430460         0.48358771E-01    21.568911
Col____8     0      1.0129373         0.46473243E-01    21.796139
Col____9     0      1.0089965         0.45857376E-01    22.002926
Col___10     0      1.0246920         0.45022968E-01    22.759316
CONSTANT     0    -0.10983490         0.47180545E-01   -2.3279702
Tests on Linear     model y=sum(x)+e

Obs      _MTRY       _NDSIZE     _RSS1       _RSS2       _RSS3
     1           1           1    3283.       2642.       770.6
     2           2           1    5193.       2573.       757.6
     3           2           2    2824.       2345.       523.1
     4           3           1    3503.       2661.       771.1
     5           3           2    2356.       2301.       512.7
     6           3           3    2669.       2145.       485.1
     7           4           1    3464.       2552.       746.0
     8           4           2    1992.       2345.       516.7
     9           4           3    2201.       2268.       515.3
    10           4           4    2359.       2243.       549.7
    11           5           1    4252.       2606.       767.9
    12           5           2    2912.       2261.       509.1
    13           5           3    2693.       2241.       504.0
    14           5           4    2400.       2215.       540.3
    15           5           5    2303.       2230.       614.7
    16           6           1    3881.       2598.       777.1
    17           6           2    2751.       2302.       511.6
    18           6           3    2554.       2231.       497.2
    19           6           4    2652.       2250.       553.0
    20           6           5    2366.       2252.       619.0
    21           6           6    2469.       2221.       676.9
    22           7           1    3069.       2584.       766.9
    23           7           2    2317.       2265.       501.0
    24           7           3    2103.       2190.       494.6
    25           7           4    2138.       2233.       551.4
    26           7           5    2170.       2324.       637.2
    27           7           6    2693.       2254.       682.5
    28           7           7    2639.       2316.       754.1
    29           8           1    4179.       2553.       757.0
    30           8           2    2877.       2279.       515.7
    31           8           3    2594.       2227.       494.8
    32           8           4    2020.       2209.       551.8
    33           8           5    2624.       2252.       615.4
    34           8           6    2728.       2248.       676.6
    35           8           7    2554.       2266.       756.3
    36           8           8    2543.       2306.       824.9
    37           9           1    4115.       2638.       760.6
    38           9           2    1919.       2305.       506.3
    39           9           3    2309.       2257.       510.5
    40           9           4    2398.       2244.       559.2
    41           9           5    2912.       2302.       633.0
```

| | | | | | |
|---|---|---|---|---|---|
| 42 | 9 | 6 | 2736. | 2224. | 659.7 |
| 43 | 9 | 7 | 3185. | 2332. | 757.7 |
| 44 | 9 | 8 | 2554. | 2328. | 827.7 |
| 45 | 9 | 9 | 2922. | 2329. | 891.0 |
| 46 | 10 | 1 | 3668. | 2662. | 786.7 |
| 47 | 10 | 2 | 2650. | 2335. | 516.4 |
| 48 | 10 | 3 | 2639. | 2189. | 500.5 |
| 49 | 10 | 4 | 2416. | 2191. | 538.7 |
| 50 | 10 | 5 | 2324. | 2288. | 622.0 |
| 51 | 10 | 6 | 2325. | 2296. | 682.9 |
| 52 | 10 | 7 | 2824. | 2347. | 756.1 |
| 53 | 10 | 8 | 2731. | 2335. | 831.4 |
| 54 | 10 | 9 | 2757. | 2390. | 893.8 |
| 55 | 10 | 10 | 2742. | 2396. | 949.8 |

For the nonlinear model the OLS $e'e = 2485.400$ which is inferior by a factor of 6 to the random forest average results.

Tests of Non Linear model y=sum(abs(y))+e

| | |
|---|---|
| Ordinary Least Squares Estimation | |
| Dependent variable | Y |
| Centered R**2 | 1.026132618007292E-02 |
| Adjusted R**2 | -9.978728499271197E-03 |
| Residual Sum of Squares | 2485.400724914628 |
| Residual Variance | 5.082619069355067 |
| Standard Error | 2.254466471109088 |
| Total Sum of Squares | 2511.168645479060 |
| Log Likelihood | -1110.364537757051 |
| Mean of the Dependent Variable | 7.894844207532995 |
| Std. Error of Dependent Variable | 2.243301605925609 |
| Sum Absolute Residuals | 901.5882764477245 |
| F(10, 489) | 0.5069811491440812 |
| F Significance | 0.1145616129669150 |
| 1/Condition XPX | 0.5441552376397775 |
| Maximum Absolute Residual | 9.435083459129245 |
| Number of Observations | 500 |

| Variable | Lag | Coefficient | SE | t |
|---|---|---|---|---|
| Col____1 | 0 | -0.27934052E-01 | 0.95974845E-01 | -0.29105597 |
| Col____2 | 0 | 0.36464466E-01 | 0.10433096 | 0.34950764 |
| Col____3 | 0 | -0.36943336E-01 | 0.99174578E-01 | -0.37250812 |
| Col____4 | 0 | 0.17687733E-01 | 0.10430380 | 0.16957900 |
| Col____5 | 0 | -0.52334419E-01 | 0.10098709 | -0.51822883 |
| Col____6 | 0 | -0.81101111E-01 | 0.10639986 | -0.76222949 |
| Col____7 | 0 | 0.15470738 | 0.10523645 | 1.4700931 |
| Col____8 | 0 | 0.47121358E-02 | 0.10113324 | 0.46593343E-01 |
| Col____9 | 0 | -0.13479463 | 0.99793014E-01 | -1.3507422 |
| Col___10 | 0 | -0.13859682E-01 | 0.97977210E-01 | -0.14145822 |
| CONSTANT | 0 | 7.9079719 | 0.10267244 | 77.021368 |

Tests of Non Linear model y=sum(abs(y))+e

| Obs | _MTRY | _NDSIZE | _RSS1 | _RSS2 | _RSS3 |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 2531. | 1640. | 432.2 |
| 2 | 2 | 1 | 2457. | 1627. | 429.3 |
| 3 | 2 | 2 | 1563. | 1531. | 309.0 |
| 4 | 3 | 1 | 2093. | 1649. | 444.8 |
| 5 | 3 | 2 | 1675. | 1528. | 311.4 |
| 6 | 3 | 3 | 1429. | 1516. | 350.2 |
| 7 | 4 | 1 | 1991. | 1652. | 430.5 |
| 8 | 4 | 2 | 1181. | 1559. | 314.7 |
| 9 | 4 | 3 | 1216. | 1528. | 352.3 |
| 10 | 4 | 4 | 1725. | 1525. | 408.5 |
| 11 | 5 | 1 | 1929. | 1648. | 437.2 |
| 12 | 5 | 2 | 1397. | 1532. | 315.3 |
| 13 | 5 | 3 | 1458. | 1531. | 356.0 |
| 14 | 5 | 4 | 1261. | 1509. | 407.1 |
| 15 | 5 | 5 | 1438. | 1531. | 463.3 |
| 16 | 6 | 1 | 2152. | 1642. | 434.2 |
| 17 | 6 | 2 | 1460. | 1566. | 318.3 |
| 18 | 6 | 3 | 1211. | 1508. | 344.3 |
| 19 | 6 | 4 | 1465. | 1512. | 404.6 |
| 20 | 6 | 5 | 1411. | 1548. | 464.8 |
| 21 | 6 | 6 | 1581. | 1564. | 517.0 |
| 22 | 7 | 1 | 2368. | 1643. | 432.8 |
| 23 | 7 | 2 | 1233. | 1526. | 306.6 |
| 24 | 7 | 3 | 1487. | 1499. | 351.2 |
| 25 | 7 | 4 | 1391. | 1546. | 412.3 |
| 26 | 7 | 5 | 1365. | 1520. | 461.8 |
| 27 | 7 | 6 | 1506. | 1539. | 517.5 |
| 28 | 7 | 7 | 1674. | 1560. | 560.1 |
| 29 | 8 | 1 | 1654. | 1672. | 439.0 |
| 30 | 8 | 2 | 1459. | 1547. | 318.9 |
| 31 | 8 | 3 | 1265. | 1514. | 345.6 |
| 32 | 8 | 4 | 1283. | 1572. | 416.9 |
| 33 | 8 | 5 | 1588. | 1514. | 456.7 |
| 34 | 8 | 6 | 1597. | 1564. | 523.6 |
| 35 | 8 | 7 | 1547. | 1525. | 558.8 |
| 36 | 8 | 8 | 1474. | 1548. | 595.2 |
| 37 | 9 | 1 | 2225. | 1658. | 446.5 |
| 38 | 9 | 2 | 1444. | 1524. | 311.4 |
| 39 | 9 | 3 | 1310. | 1510. | 346.4 |
| 40 | 9 | 4 | 1199. | 1510. | 402.1 |

```
     41            9           5    1438.       1518.       450.9
     42            9           6    1357.       1556.       521.0
     43            9           7    1509.       1554.       559.7
     44            9           8    1651.       1549.       600.7
     45            9           9    1575.       1594.       646.4
     46           10           1    1762.       1645.       452.8
     47           10           2    1320.       1527.       311.3
     48           10           3    1406.       1543.       355.3
     49           10           4    1488.       1530.       410.3
     50           10           5    1506.       1524.       449.8
     51           10           6    1782.       1536.       517.8
     52           10           7    1378.       1544.       563.2
     53           10           8    1553.       1563.       601.3
     54           10           9    1657.       1584.       653.0
     55           10          10    1449.       1571.       679.3

B34S Matrix Command Ending. Last Command reached.

Space available in allocator   11856466, peak  space  used      83057
Number variables used                10, peak number  used        116
Number temp variables used         5131, # user temp clean          0
```

## 17.6 Cluster Analysis - Unsupervised Machine learning

Because of the availability of an outcome variable, *y* that could be used to guide / validated the analysis, the statistical procedures discussed in prior sections of this chapter have been characterized as supervised learning[11] models. When such a *y* variable is not available, machine learning is called unsupervised. Problems of the *K>>N* class, where there are more possible explanatory variables than observations, cannot be solved using supervised learning models. For such problems, reducing the number of variables to consider is of the utmost importance. Cluster analysis is an option in such situations. There are two main approaches to cluster analysis, the k-means model and hierarchical cluster model, which will be discussed in turn. Assume there are *K* columns of *N* observations. The investigator first specifies the number of classes *k* where $k \le K$ . The selected cluster technique then assigns each observation to one of the *k* classes.[12]

A k-means model minimizes the total within-cluster sums of squares. The total sums of squares computed over all non-missing values of each variable for *k* classes is

$$\phi(k) = \sum_{i=1}^{k} \sum_{j=1}^{K} \sum_{m=1}^{n_i} f_{v_{im}} w_{v_{im}} \delta_{v_{im} j} (x_{v_{im}} - \overline{x}_{ij})^2 \qquad (17.6\text{-}1)$$

where $v_{im}$ = the row index of the m$^{th}$ observation of the i$^{th}$ cluster in the data matrix *X*, $n_i$ equals the number of rows assigned to group *i, f* denotes the frequency of the observation, and *w* denotes its weight. Usually $f = 1$ and $w = 1$. $\delta = 1$ unless observation $v_{im}$ of the j$^{th}$ variable is missing. $\overline{x}_{ij}$ is the average of the non missing values for the j$^{th}$ variable for observation *i*. As indicated in the IMSL documentation, the k-means method "sequentially processes each observation and reassigns it to another cluster if by doing so results in a decrease in the total within-cluster sums of squares." Since k-means cluster analysis requires that the investigator set the number of

---

11 Hastie-Tibshirani-Friedman (2009) contains a discussion of supervised vs non-supervised learning and how this distinction affects the analysis.

12 The IMSL routine dk2ean is used to estimate the k-means model while the IMSL routines dc2ist, dc2ink and c2umb are used to estimate the hierarchical cluster moidel.

classes, unless there is a compelling reason to only investigate one class size, it might be prudent to search over a number of possible class sizes. Hastie-Tibshirani-Friedman (2009, 518-519) thus suggest calculating and plotting $\phi(k)$ and looking for a break in the rate at which $\phi(k)$ declines as $k$ increases. Assuming $k^*$ is the optimum number of classes. If $k<k^*$ then as $k$ is increased, $\phi(k)$ will fall relatively fast as each class approaches its optimum composition. If $k>k^*$ then as $k$ is increased, there will be only marginal decreases in $\phi(k)$ as $k\uparrow$.

Hierarchical clustering proceeds by first computing a distance between each observation using one of a number of possible methods. Supported methods by code number for setting :method are:

- 0. Euclidean distance (L-2 norm).

- 1. Sum of absolute differences (L-1 norm).

- 2. Maximum difference (L-infinity norm).

- 3. Mahalanobis distance (Hastie-Tibshirani-Friedman 2009 441).

- 4. Absolute value of the cosine of the angle between the vectors.

- 5. Angle in radians (0, $\pi$) between the lines through the origin defined by the vectors.

- 6. Correlation coefficient.

- 7. Absolute value of the correlation coefficient.

- 8. Number of exact matches.

Once the distance is calculated, the two clusters that are closest to each other are merged and the distance of the new cluster from all other clusters is calculated. Supported methods for this setting :dist are:

- 0. single linkage (minimum distance method). Default.

- 1. complete linkage (maximum distance)

- 2. average distance between objects within the merged cluster.

- 3. average distance between objects in the two clusters.

- 4. Ward's method that minimizes the within-cluster sums of squares. For

Ward's method the elements of distances are assumed to be Euclidean distances.

For the :`h_cluster` method, output includes the distance or similarity matrix, `%dist`, a *N-1* vector `%clevel` that indicates the level at which the clusters are joined, `%clson` an integer*4 vector of length N-1 containing the left cluster number and `%crson` an integer*4 vector of length N-1 containing the right cluster numbers. Cluster `n+i` is formed by merging clusters `%clson(i)` and `%crson(i)`.

For the `:k_mean` method, output includes the *k* by *N* matrix `%sumw` containing the sum of weights used to compute each cluster, `%wss` a vector of length *k* containing the within sum of squares and `%tss` which is `sum(%wss)`. Hastie-Tibshirani-Friedman (2009, 519) suggest using `%tss` for various *k* settings to determine the appropriate number of classes. The variable `%ave_wss` is a vector of length *k* containing within sum of squares divided `%nclus`. Finally `%clust_m` is a *k* by `nvar` matrix containing the cluster means.

For both the k-means and h-cluster methods, the vector `%iclus` of length *N* indicates to which cluster each observation is assigned and the *k* element vector `%nclus` indicates the number of observations assigned to each class. As an example, consider the IMSL test case for cluster analysis that utilizes the Fisher iris data divided by 100. The input file is listed in Table 17.6 and the edited output is shown below.

Table 17.6 Simple Cluster Test Case Using Iris Data

```
b34sexec options ginclude('b34sdata.mac') member(iris2); b34srun;

b34sexec matrix;
call loaddata;

call cluster(x1 x2 x3 x4 :k_mean 3 :print :savex);
call print(%iclus %sumw %clust_m %wss %ave_wss);

call print('+++++++++++++++++++++++++++++++++++':);
i=0;
j=0;
call cluster(x1 x2 x3 x4 :h_cluster 5 :print :dist i :method j);
call print('Sum of distance', sum(%dist));
call print(%clevel %clson %crson %iclus);
call graph(%dist :plottype meshc
           :heading 'Plot of Symetric Distance Matrix'
                  :rotation 90. :grid :d3axis :d3border
         :file 'h_dist.wmf'
        );
b34srun;
```

It was postulated that there were 3 classes. The k-means cluster procedure assigned 50 of the 150 observations to class 1, 62 to class 2 and 38 to class 3.

```
Variable      # Cases       Mean           Std Deviation       Variance          Maximum          Minimum

OBS       1       150   75.50000000       43.44536799       1887.500000       150.0000000       1.000000000
```

```
Y         2       150   2.000000000      0.8192319205      0.6711409396       3.000000000       1.000000000
X1        3       150   5.843333333      0.8280661280      0.6856935123       7.900000000       4.300000000
X2        4       150   3.057333333      0.4358662849      0.1899794183       4.400000000       2.000000000
X3        5       150   3.758000000       1.765298233       3.116277852       6.900000000       1.000000000
X4        6       150   1.199333333      0.7622376690      0.5810062640       2.500000000      0.1000000000
CONSTANT  7       150   1.000000000       0.000000000       0.000000000       1.000000000       1.000000000

Number of observations in data file   150
Current missing variable code         1.000000000000000E+31

B34S(r) Matrix Command. d/m/y 30/ 7/09. h:m:s 16:25:47.


=>  CALL LOADDATA$


=>  CALL CLUSTER(X1 X2 X3 X4 :K_MEAN 3 :PRINT :SAVEX)$


K_MEAN Cluster Analysis
Number of classes  (iclass)                     3
Number of Observations                          150
Number of variables                             4

Series   Lag      Mean          Max           Min
X1         0     5.843         7.900         4.300
X2         0     3.057         4.400         2.000
X3         0     3.758         6.900         1.000
X4         0     1.199         2.500        0.1000

Total sum of squares for all clusters           78.85144142614600

Number of observations in each cluster

          1     2     3

         50    62    38

Sum of squares within each cluster

            1              2              3
         15.1510        39.8210        23.8795

Average sum of squares within each cluster

            1              2              3
        0.303020       0.642274       0.628407


=>  CALL PRINT(%ICLUS %SUMW %CLUST_M %WSS %AVE_WSS)$
%ICLUS  = Array  of      150    elements

       1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
       1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1     1
       1     1     1     1     1     1     1     1     1     1     2     2     3     2     2     2     2     2     2     2
       2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     3     2     2
       2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2
       3     2     3     3     3     3     2     3     3     3     3     3     3     2     2     3     3     3     3     2
       3     2     3     2     3     3     2     2     3     3     3     3     3     2     3     3     3     3     2     3
       3     3     2     3     3     3     2     3     3     2


%SUMW   = Array  of        3  by       4  elements

            1              2              3              4
    1    50.0000        50.0000        50.0000        50.0000
    2    62.0000        62.0000        62.0000        62.0000
    3    38.0000        38.0000        38.0000        38.0000


%CLUST_M= Array  of        3  by       4  elements

            1              2              3              4
    1    5.00600        3.42800        1.46200       0.246000
    2    5.90161        2.74839        4.39355        1.43387
    3    6.85000        3.07368        5.74211        2.07105


%WSS    = Array  of        3   elements

      15.1510        39.8210        23.8795


%AVE_WSS= Array  of        3   elements

     0.303020       0.642274       0.628407
```

The next output is the same problem run with the hierarchical cluster model where 5 classes are postulated. The symmetric distance matrix is displayed graphically in Figure 17.6.

```
=>  CALL PRINT('++++++++++++++++++++++++++++++++++':)$
```

```
+++++++++++++++++++++++++++++++++

=>  I=0$


=>  J=0$


=>  CALL CLUSTER(X1 X2 X3 X4 :H_CLUSTER 5 :PRINT :DIST I :METHOD J)$


H_Cluster Analysis
Number of classes  (iclass)                     5
Number of Observations                          150
Number of variables                             4
Single linkage (minimum distance method) :method 0
Euclian Distance (L-2 norm). :dist 0

Series   Lag      Mean          Max          Min
X1         0     5.843         7.900        4.300
X2         0     3.057         4.400        2.000
X3         0     3.758         6.900        1.000
X4         0     1.199         2.500       0.1000

Number of observations in each cluster

          1     2     3     4     5

          4    93     1     2    50


=>  CALL PRINT('Sum of distance', SUM(%DIST))$



          Sum of distance

              56883.896

=>  CALL PRINT(%CLEVEL %CLSON %CRSON %ICLUS)$



%CLEVEL = Vector of      149   elements

     0.394447E-03   0.100154       0.100181       0.100395       0.100448       0.100797       0.141454       0.141476
     0.141549       0.141613       0.141618       0.141658       0.141688       0.141736       0.141753       0.141760
     0.141765       0.141800       0.141803       0.141809       0.141890       0.141978       0.141981       0.142125
     0.142140       0.142223       0.142266       0.142272       0.142365       0.142369       0.142370       0.142411
     0.173362       0.173538       0.173598       0.173758       0.173891       0.173933       0.174112       0.200202
     0.200225       0.200485       0.200743       0.200768       0.200877       0.223655       0.223761       0.223842
     0.223861       0.223897       0.223984       0.224148       0.224270       0.224335       0.224450       0.224479
     0.224557       0.244982       0.245007       0.245050       0.245098       0.245137       0.245256       0.245273
     0.245369       0.245407       0.245465       0.245628       0.245881       0.264721       0.264727       0.264763
     0.264776       0.264786       0.264820       0.264861       0.264917       0.264933       0.264934       0.265471
     0.283133       0.283508       0.283542       0.300148       0.300170       0.300246       0.300435       0.300658
     0.300732       0.300760       0.300988       0.316357       0.316438       0.316462       0.316771       0.316816
     0.317159       0.331985       0.332042       0.332245       0.332337       0.332478       0.332529       0.346583
     0.346594       0.346634       0.346681       0.346739       0.346857       0.347305       0.347326       0.347409
     0.360594       0.360661       0.360727       0.360846       0.360913       0.361117       0.361185       0.361197
     0.374841       0.374879       0.375076       0.387644       0.387719       0.387959       0.400674       0.412374
     0.412530       0.412740       0.413304       0.424949       0.425221       0.436276       0.436309       0.458899
     0.490056       0.490811       0.510172       0.529832       0.538607       0.539150       0.557159       0.624504
     0.632987       0.648804       0.735084       0.819236        1.64055

%CLSON  = Vector of      149    elements

     143    49   133    18    35    40   154   100    93    39    31    76   155   161   158   156   166   164    48    94
     167    82   168    38   169    92   171   138   177    22   139   180   172   140   165   173   185   127   186    85
      27   160   176    98    90   189   152   197   198   192   193   191   200   181   203   148   144   159   205   182
     145   146   162   188   178   149   183   214   213   187   131   208    57    53   217   207   151   123   194    88
      32   218   224   210   231   225   153   235   190   216   226   236   227   229   223   245   246   247    19   201
     232   215   134   248   238   206   184   130   255   249   254   260   262   263   252   251   212   266   170   257
     240   268   261   258   273   269   272   264   275   228   132   279   282   221   284   278   285   287   230   280
     290   291   292   286   293   276   296   281   294

%CRSON  = Vector of      149    elements

     102    11   129     1    10     8    29    97    83     9    30    66     2   163    96   157    50    46     4    58
      28    81    13     5     3    64   174   117    41    20   128    47    70   113    95   175    89   124    26    67
      24    43    79    75    54     7   179    36    12   196    74   199   202    71    44   111   121    68    14   209
     141   142    59   204   104   137   195   147    55    91   108   220    52    51   222   211   114   106   219    69
      21   150    87    37   234    62   105    25    56   116   125   239   122   233   244    78    77    72     6   242
     243   237    84   250    17   112   241   126    33   259    80    34    16    45   256   253   265    73    61   267
     270   271    86   103    60    99   274    15   277   119   118   101    65   283   120    23   115    63   288   289
     136   135   109    42   110   295   107   297   298

%ICLUS  = Array  of      150    elements

       5     5     5     5     5     5     5     5     5     5     5     5     5     5     5     5     5     5     5     5
       5     5     5     5     5     5     5     5     5     5     5     5     5     5     5     5     5     5     5     5
       5     5     5     5     5     5     5     5     5     5     2     2     2     2     2     2     2     1     2     2
       1     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2
       2     2     2     2     2     2     2     2     2     2     2     2     2     1     2     2     2     2     1     2
```

```
      2     2     2     2     2     2     3     2     2     2     2     2     2     2     2     2     2     4     2     2
      2     2     2     2     2     2     2     2     2     2     2     4     2     2     2     2     2     2     2     2
      2     2     2     2     2     2     2     2     2     2


=>  CALL GRAPH(%DIST :PLOTTYPE MESHC
=>             :HEADING 'Plot of Symetric Distance Matrix'
=>                   :ROTATION 90. :GRID :D3AXIS :D3BORDER
=>          )$


B34S Matrix Command Ending. Last Command reached.

Space available in allocator    5874853, peak  space  used      51833
Number variables used                36, peak number  used         38
Number temp variables used           33, # user temp clean          0
```

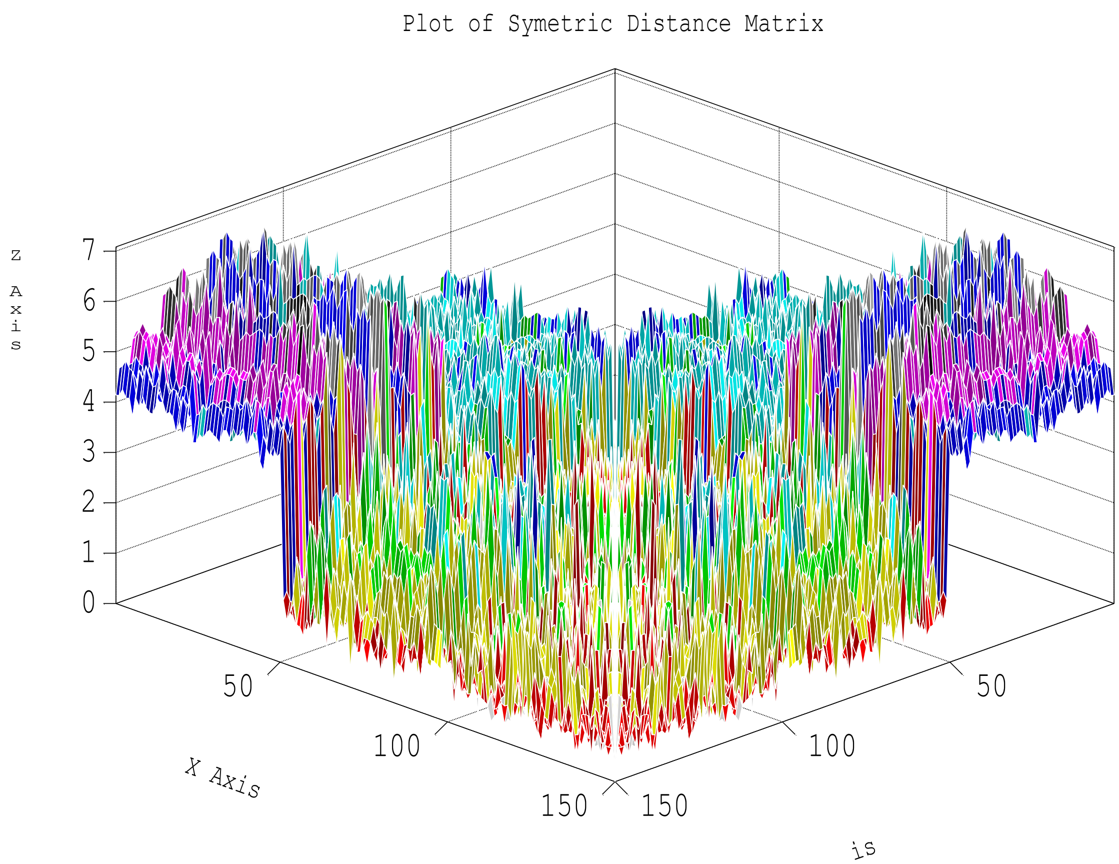



Figure 17.8 Symmetric Distance Matrix of hierarchical cluster model.

The code in Table 17.7 and its associated Figure 17.8 illustrates how to determine the appropriate number of classes.

Table 17.7 Determining the correct number of classes

```
b34sexec options ginclude('b34sdata.mac') member(iris2); b34srun;

b34sexec matrix;
call echooff;
call loaddata;

save_tss=array(9:);
ncluster=dfloat(integers(2,10));
icount=1;
do i=2,10;
call cluster(x1 x2 x3 x4 :k_mean i );
save_tss(icount)=%tss;
icount=icount+1;
enddo;
call tabulate(ncluster,save_tss);
call graph(ncluster,save_tss :plottype xyplot
           :nocontact :pgborder
           :file 'tss_test.wmf'
           :heading 'Total sum of squares vs # of classes');
b34srun;
```

Results in the following Table:

| Obs | NCLUSTER | SAVE_TSS |
|---|---|---|
| 1 | 2.000 | 152.3 |
| 2 | 3.000 | 78.85 |
| 3 | 4.000 | 71.45 |
| 4 | 5.000 | 49.82 |
| 5 | 6.000 | 39.05 |
| 6 | 7.000 | 36.84 |
| 7 | 8.000 | 32.20 |
| 8 | 9.000 | 35.27 |
| 9 | 10.00 | 30.01 |

which when graphed in Figure 17.9 shows the classic kink at the point of the appropriate number of classes which in the case of the Fisher data we know to be three.

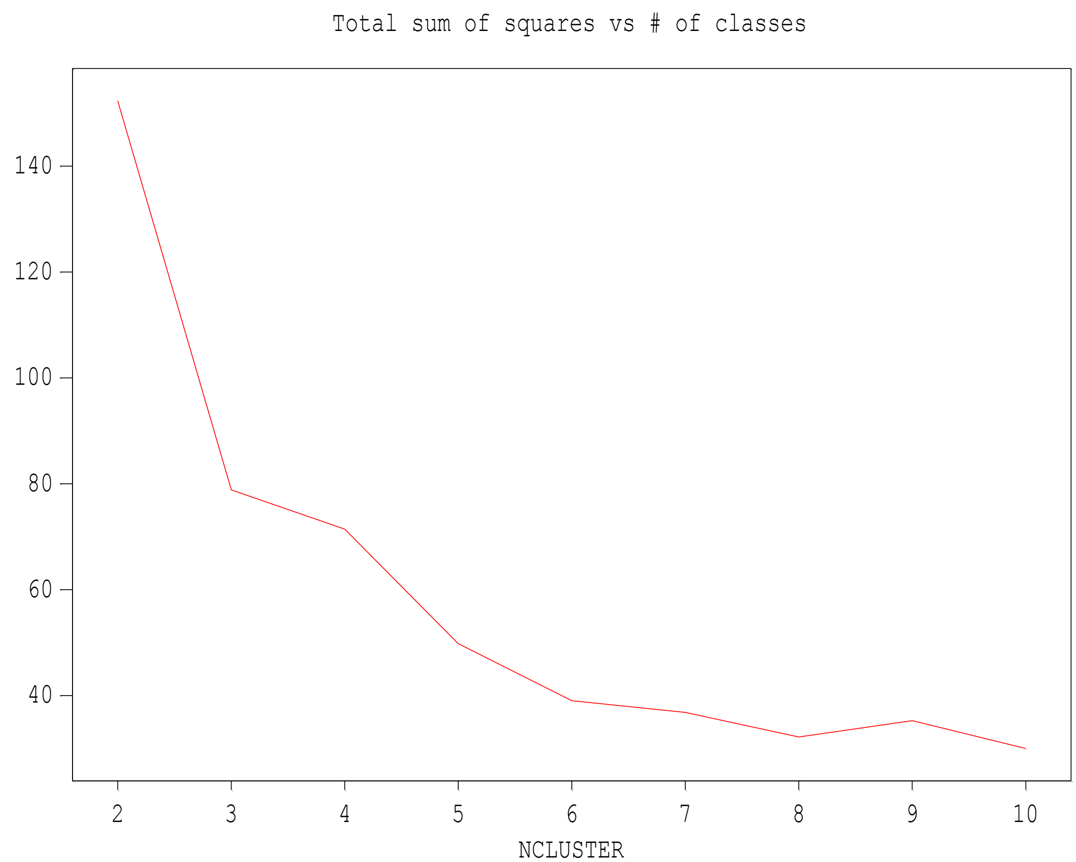



Figure 17.9 Analysis of the total sum of squares as a function of the number of classes

Table 17.8 contains the setup to study the Human Tumor DNA dataset that was discussed in Hastie-Tibshirani-Friedman (2009, 3-7, 512-513). There are 6830 genes and 64 samples of tumors. The example also calculates the total sum or squares assuming a grid of from 2 to 12 classes. As will be discussed below for $k = 3$ the class numbers calculated are 9, 34 and 21 which exactly match Hastie-Tibshirani-Friedman (2009, 513).[13]

13 The catagories do not match 100% however possibly due to column names being updated in the supplied data.

Table 17.8 Cluster Analysis applied to micro array data

```
/;
/; Replicate H-T-F (2009) Page 513 - see docase3 & docase4
/;
b34sexec options ginclude('h_t_f_data.mac')
         member(cancer);
         b34srun;
b34sexec matrix;
call loaddata;
call echooff;
/;
/; This must be first before other variables are created
/;
call names(:);

docase1=1;
docase2=1;
docase3=1;
docase4=1;

i=norows(%names%);
nn=%names%(integers(2,i-2));
/; call print(label(argument(nn(1))));
tt=label(argument(nn));

/; call print(tt);
/; call print(transpose(tt));
tt=transpose(tt);

bigx = catcol(x_1   x_2  x_3  x_4  x_5  x_6  x_7  x_8  x_9 x_10
              x_11 x_12 x_13 x_14 x_15 x_16 x_17 x_18 x_19 x_20
              x_21 x_22 x_23 x_24 x_25 x_26 x_27 x_28 x_29 x_30
              x_31 x_32 x_33 x_34 x_35 x_36 x_37 x_38 x_39 x_40
              x_41 x_42 x_43 x_44 x_45 x_46 x_47 x_48 x_49 x_50
              x_51 x_52 x_53 x_54 x_55 x_56 x_57 x_58 x_59 x_60
              x_61 x_62 x_63 x_64);

* call graph(bigx :plottype meshstepc :heading 'Cancer Data'
        :rotation 90. :grid :d3axis :d3border
/;      :file 'rawdata.wmf'
        );
* call graph(bigx :plottype meshc     :heading 'Cancer Data'
/;      :file 'rawdata.wmf'
        :rotation 90. :grid :d3axis :d3border
        );

if(docase1.ne.0)then;

call print('Testing which Genes are most similar across samples':);
i=10;
call cluster(bigx :print :k_mean i );
ii=ranker(dfloat(%iclus));
newbigx=bigx(ii,);
call compress;
call graph(newbigx
     :plottype meshstepc :heading 'Data Clustered into 10 classes'
     :rotation 90. :grid :d3axis :d3border :angle 25.
/;              :file 'sorted_data1.wmf'
        );

endif;
```

```
/;
/; Investigate the size of k
/;
if(docase2.ne.0)then;
n1=2;
n2=16;
ntotal=n2-n1+1;
nclass  =array(ntotal:);
sumclass=array(ntotal:);
s_av_wss =array(ntotal:);

icount=0;

save_tss=array(11:);
do i=n1,n2;
icount=icount+1;
nclass(icount)=dfloat(i);
call cluster(bigx :k_mean i );
save_tss(icount)=%tss;
call compress;
enddo;

call graph(nclass save_tss :plottype xyplot
                  :nocontact :pgborder :grid
                  :file 'n1.class.wmf'
     :heading 'Total Sum of squares vs # of classes for gene test');

call tabulate(nclass save_tss);
endif;

bigx=transpose(bigx);

if(docase3.ne.0)then;
i=3;
/;
/; Replicate results on page 513
/;
call print('Testing which tumors are most similar across genes':);
call cluster(bigx :print :k_mean i );
call print(%iclus);
ii=ranker(dfloat(%iclus));
newbigx=bigx(ii,);
newtt2=tt(ii,);

call compress;
call graph(newbigx
     :plottype meshstepc :heading 'Data Clustered into 3 classes'
     :rotation 90. :grid :d3axis :d3border :angle 25.
/;                :file 'sorted_data2.wmf'
        );
newlist=c8array(64:);

do j=1,64;
call pcopy(8,pointer(newtt2,j),64,pointer(newlist,j),1,-1);
enddo;

class=%iclus(ii);
call print(' ':);
call tabulate(ii,class,newlist :rjname);
endif;

/;
/; Investigate the size of k
```

```
/;
if(docase4.ne.0)then;
n1=2;
n2=12;
ntotal=n2-n1+1;
nclass  =array(ntotal:);
sumclass=array(ntotal:);
s_av_wss =array(ntotal:);

icount=0;

save_tss=array(11:);
do i=n1,n2;
icount=icount+1;
nclass(icount)=dfloat(i);
call cluster(bigx :k_mean i );
save_tss(icount)=%tss;
call compress;
enddo;

call tabulate(nclass save_tss);
call graph(nclass save_tss :plottype xyplot
                   :nocontact :pgborder :grid
                   :file 'n2_class.wmf'
:heading 'Total Sum of squares vs # of classes for Tumor test');
endif;

b34srun;
```

Edited results from running this example are

| Variable | Label | # Cases | Mean | Std. Dev. | Variance | Maximum | Minimum |
|---|---|---|---|---|---|---|---|
| X_1 | 1 CNS_1 | 6830 | 0.653012E-01 | 0.724547 | 0.524969 | 6.17500 | -5.53500 |
| X_2 | 2 CNS_2 | 6830 | 0.507640E-01 | 0.771361 | 0.594997 | 7.21996 | -5.60504 |
| X_3 | 3 CNS_3 | 6830 | 0.721869E-01 | 0.679115 | 0.461198 | 6.37499 | -3.04002 |
| X_4 | 4 RENAL_1 | 6830 | 0.938286E-01 | 0.900368 | 0.810663 | 7.56500 | -4.88000 |
| X_5 | 5 BREAST_2 | 6830 | 0.148587 | 0.992909 | 0.985868 | 7.42500 | -4.68000 |
| X_6 | 6 CNS_4 | 6830 | 0.519845E-01 | 0.800463 | 0.640741 | 7.71500 | -5.05000 |
| X_7 | 7 CNS_5 | 6830 | 0.459465E-01 | 0.831638 | 0.691621 | 8.66000 | -4.28000 |
| X_8 | 8 BREAST_3 | 6830 | 0.348982E-01 | 0.771968 | 0.595935 | 5.65000 | -5.52000 |
| X_9 | 9 NSCLC_1 | 6830 | 0.307656E-01 | 0.675583 | 0.456412 | 5.14500 | -4.24000 |
| X_10 | 10 NSCLC_2 | 6830 | 0.719695E-01 | 0.913155 | 0.833851 | 7.63501 | -4.79500 |
| X_11 | 11 RENAL_2 | 6830 | 0.829875E-01 | 0.746204 | 0.556821 | 7.26000 | -3.70500 |
| X_12 | 12 RENAL_3 | 6830 | 0.975109E-01 | 0.763894 | 0.583535 | 5.80000 | -4.43500 |
| X_13 | 13 RENAL_4 | 6830 | 0.438711E-01 | 0.617856 | 0.381746 | 4.62000 | -4.40000 |
| X_14 | 14 RENAL_5 | 6830 | 0.803056E-01 | 0.675523 | 0.456332 | 5.78000 | -2.97500 |
| X_15 | 15 RENAL_6 | 6830 | 0.865903E-01 | 0.754648 | 0.569493 | 5.75000 | -4.75000 |
| X_16 | 16 RENAL_7 | 6830 | 0.682287E-01 | 0.724551 | 0.524975 | 6.06000 | -3.15000 |
| X_17 | 17 RENAL_8 | 6830 | 0.369183E-01 | 0.763342 | 0.582691 | 5.61500 | -4.58000 |
| X_18 | 18 BREAST_4 | 6830 | 0.541427E-02 | 0.819278 | 0.671216 | 5.88000 | -5.38000 |
| X_19 | 19 NSCLC_2 | 6830 | 0.441087E-01 | 0.811423 | 0.658407 | 7.35500 | -4.79249 |
| X_20 | 20 RENAL_9 | 6830 | -0.108101E-01 | 0.857966 | 0.736106 | 8.00500 | -4.59002 |
| X_21 | 21 UNKNOWN_1 | 6830 | 0.224860E-01 | 0.696755 | 0.485467 | 5.32000 | -3.71996 |
| X_22 | 22 OVARIAN_1 | 6830 | 0.212042E-01 | 0.772627 | 0.596953 | 6.45000 | -4.34000 |
| X_23 | 23 MELANOMA_1 | 6830 | 0.225574E-01 | 0.699865 | 0.489811 | 6.83000 | -4.43000 |
| X_24 | 24 PROSTATE_1 | 6830 | 0.285893E-01 | 0.630123 | 0.397055 | 5.26000 | -3.60501 |
| X_25 | 25 OVARIAN_2 | 6830 | 0.553568E-01 | 0.811066 | 0.657828 | 7.91500 | -5.34000 |
| X_26 | 26 OVARIAN_3 | 6830 | 0.760168E-01 | 0.768507 | 0.590603 | 6.13999 | -4.23000 |
| X_27 | 27 OVARIAN_4 | 6830 | 0.376961E-01 | 0.657908 | 0.432842 | 7.18000 | -4.05250 |
| X_28 | 28 OVARIAN_5 | 6830 | 0.305438E-01 | 0.715700 | 0.512227 | 5.69000 | -4.78000 |
| X_29 | 29 OVARIAN_6 | 6830 | 0.527455E-01 | 0.638504 | 0.407687 | 4.79000 | -3.39000 |
| X_30 | 30 PROSTATE_2 | 6830 | 0.294985E-01 | 0.575334 | 0.331010 | 5.47999 | -2.97000 |
| X_31 | 31 NSCLC_3 | 6830 | 0.658671E-01 | 0.679016 | 0.461062 | 5.82000 | -3.85500 |
| X_32 | 32 NSCLC_4 | 6830 | 0.453370E-01 | 0.608540 | 0.370321 | 7.51000 | -2.93001 |
| X_33 | 33 NSCLC_5 | 6830 | 0.356831E-01 | 0.717497 | 0.514802 | 7.10500 | -3.71000 |
| X_34 | 34 LEUKEMIA_1 | 6830 | 0.868396E-02 | 0.816411 | 0.666527 | 5.44000 | -4.30996 |
| X_35 | 35 K562B-repro_1 | 6830 | -0.250114E-01 | 0.887368 | 0.787422 | 8.45000 | -4.70000 |
| X_36 | 36 K562A-repro_2 | 6830 | -0.671798E-01 | 0.943257 | 0.889734 | 8.17500 | -4.96500 |
| X_37 | 37 LEUKEMIA_1 | 6830 | -0.732477E-01 | 0.967338 | 0.935742 | 8.36002 | -5.38498 |
| X_38 | 38 LEUKEMIA_2 | 6830 | -0.593643E-01 | 0.900940 | 0.811693 | 6.20000 | -4.78500 |
| X_39 | 39 LEUKEMIA_3 | 6830 | -0.142086 | 1.08688 | 1.18131 | 7.05000 | -5.86000 |
| X_40 | 40 LEUKEMIA_4 | 6830 | -0.112343 | 1.03577 | 1.07281 | 7.59998 | -6.16000 |
| X_41 | 41 LEUKEMIA_5 | 6830 | -0.796739E-01 | 0.978570 | 0.957600 | 7.15000 | -5.68000 |
| X_42 | 42 COLON_1 | 6830 | 0.168402E-01 | 0.562942 | 0.316904 | 5.19996 | -3.75004 |
| X_43 | 43 COLON_2 | 6830 | -0.227442E-01 | 0.788790 | 0.622190 | 5.11000 | -5.22000 |

```
X_44      44 COLON_3                                          6830 -0.452907E-01  0.670674      0.449804       5.64000       -4.58000
X_45      45 COLON_4                                          6830  0.134815E-01  0.797377      0.635810       6.64002       -4.55498
X_46      46 COLON_5                                          6830  0.897246E-02  0.722299      0.521715       7.17000       -5.37000
X_47      47 COLON_6                                          6830  0.942966E-02  0.948977      0.900558       8.04000       -5.21500
X_48      48 COLON_7                                          6830  0.535021E-02  0.809488      0.655270       5.76998       -5.71500
X_49      49 MCF7A-repro_1                                    6830 -0.705726E-01  0.856800      0.734106       5.88000       -5.27500
X_50      50 BREAST_5                                         6830 -0.452117E-01  0.878386      0.771562       5.97998       -6.93998
X_51      51 MCF7D-repro_1                                    6830 -0.158876E-01  0.769246      0.591740       6.09002       -4.12994
X_52      52 BREAST_6                                         6830  0.141013E-01  0.849140      0.721039       6.61000       -4.83996
X_53      53 NSCLC_6                                          6830  0.502490E-01  0.731371      0.534903       6.51002       -4.01998
X_54      54 NSCLC_7                                          6830 -0.332816E-01  0.714942      0.511142       6.65000       -5.12500
X_55      55 NSCLC_8                                          6830 -0.438745E-01  0.901086      0.811956       8.18000       -5.55000
X_56      56 MELANOMA_2                                       6830  0.162778E-01  0.888810      0.789983       6.15000       -5.89996
X_57      57 BREAST_7                                         6830 -0.449605E-02  0.794938      0.631926       6.50000       -6.10002
X_58      58 BREAST_8                                         6830 -0.192574E-01  0.831060      0.690661       5.99998       -6.12004
X_59      59 MELANOMA_3                                       6830  0.406317E-01  0.756366      0.572089       6.20000       -4.86500
X_60      60 MELANOMA_4                                       6830  0.656020E-01  0.833817      0.695250       5.60000       -5.13001
X_61      61 MELANOMA_5                                       6830  0.406848E-01  0.688268      0.473713       5.35999       -3.57002
X_62      62 MELANOMA_6                                       6830  0.722298E-01  0.741078      0.549197       5.49500       -3.40001
X_63      63 MELANOMA_7                                       6830  0.216980E-01  0.816991      0.667474       5.79000       -4.96000
X_64      64 MELANOMA_8                                       6830  0.398451E-01  0.721280      0.520245       6.22500       -4.42000
CONSTANT 65                                                   6830   1.00000       0.00000       0.00000       1.00000        1.00000

Number of observations in data file   6830
Current missing variable code         1.000000000000000E+31

B34S(r) Matrix Command. d/m/y 31/ 7/09. h:m:s  7: 3:15.

=>  CALL LOADDATA$

=>  CALL ECHOOFF$
```

The first test is whether specific genes are similar to each other. A k-means model with 10 classes is attempted and a diagnostic test is performed by estimating how the sum of squares varies as the number of estimated classes varies between 2 and 16. It appears that models above 14 vary very little since for a 14 class model the total sum of squares was .2082E+6. For a 16 class model this fell only to .2081E+6.

```
Testing which Genes are most similar across samples

K_MEAN Cluster Analysis
Number of classes  (iclass)                10
Number of Observations                     6830
Number of variables                        64


Total sum of squares for all clusters      216216.0735239908

Number of observations in each cluster

          1     2     3     4     5     6     7     8     9    10

       1148   191    77  1781   291  1795   284  1039   163    61

Sum of squares within each cluster

             1              2              3              4              5              6              7              8
             9             10
         27961.1        18215.1        6333.17        33590.4        20088.2        30119.4        19554.7        35233.3
         18036.9        7083.77

Average sum of squares within each cluster

             1              2              3              4              5              6              7              8
             9             10
         24.3564        95.3668        82.2490        18.8604        69.0317        16.7796        68.8547        33.9107
         110.656        116.127


Obs      NCLASS      SAVE_TSS
     1      2.000      0.2511E+06
     2      3.000      0.2442E+06
     3      4.000      0.2367E+06
     4      5.000      0.2334E+06
     5      6.000      0.2273E+06
     6      7.000      0.2239E+06
     7      8.000      0.2206E+06
     8      9.000      0.2182E+06
     9      10.00      0.2162E+06
    10      11.00      0.2148E+06
    11      12.00      0.2129E+06
    12      13.00      0.2112E+06
    13      14.00      0.2100E+06
    14      15.00      0.2082E+06
    15      16.00      0.2081E+06
```

The next test is whether the tumors of a similar type exhibit similar gene patterns.  The numbers of tumors in each class found by the k-means cluster procedure shown next were 9, 34 and 21, the exact numbers found by Hastie-Tibshirani-Friedman (2009, 513). Finally the sum of squares for the classes is investigated for 2-12 classes and listed and plotted in Figure 17.11. The reordered gene intensity matrix is plotted in Figure 17.10 and the tumor types are listed. In class 1 there are 9 tumors, 7 of which are melanomas. These are listed as 1-9. Class 2 had 34 tumors and is listed as 10-43 while class 3 is listed as 44-64. It was noted that two breast cancers were in fact melanomas that had metastasized.

```
Testing which tumors are most similar across genes

K_MEAN Cluster Analysis
Number of classes  (iclass)                     3
Number of Observations                          64
Number of variables                             6830


Total sum of squares for all clusters           215746.3208514056

Number of observations in each cluster

          1     2     3

          9    34    21

Sum of squares within each cluster

             1              2              3
         19620.4        113624.        82502.2

Average sum of squares within each cluster

             1              2              3
         2180.04        3341.88        3928.67


%ICLUS  = Array  of       64    elements

       2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2     2
       2     2     2     2     2     2     2     2     2     2     2     2     2     3     3     3     3     3     3     3
       3     3     3     3     3     3     3     3     3     3     3     3     2     3     3     1     1     1     1     1
       1     1     1     1


Obs               II       CLASS     NEWLIST
     1           62          1  MELANOMA
     2           61          1  MELANOMA
     3           57          1  BREAST_7
     4           58          1  BREAST_8
     5           60          1  MELANOMA
     6           64          1  MELANOMA
     7           63          1  MELANOMA
     8           59          1  MELANOMA
     9           56          1  MELANOMA
    10            7          2  CNS_5
    11           13          2  RENAL_4
    12           12          2  RENAL_3
    13           14          2  RENAL_5
    14           15          2  RENAL_6
    15           23          2  MELANOMA
    16           11          2  RENAL_2
    17           53          2  NSCLC_6
    18           21          2  UNKNOWN_
    19           20          2  RENAL_9
    20            5          2  BREAST_2
    21           22          2  OVARIAN_
    22           27          2  OVARIAN_
    23            6          2  CNS_4
    24           25          2  OVARIAN_
    25           26          2  OVARIAN_
    26           24          2  PROSTATE
    27           28          2  OVARIAN_
    28           29          2  OVARIAN_
    29           30          2  PROSTATE
    30           31          2  NSCLC_3
    31           19          2  NSCLC_2
    32           18          2  BREAST_4
    33            1          2  CNS_1
    34            8          2  BREAST_3
    35           17          2  RENAL_8
    36            9          2  NSCLC_1
    37            4          2  RENAL_1
    38           33          2  NSCLC_5
    39           32          2  NSCLC_4
    40           10          2  NSCLC_2
    41           16          2  RENAL_7
    42            2          2  CNS_2
    43            3          2  CNS_3
```

```
    44           54          3  NSCLC_7
    45           52          3  BREAST_6
    46           51          3  MCF7D-re
    47           50          3  BREAST_5
    48           49          3  MCF7A-re
    49           48          3  COLON_7
    50           47          3  COLON_6
    51           46          3  COLON_5
    52           45          3  COLON_4
    53           44          3  COLON_3
    54           55          3  NSCLC_8
    55           43          3  COLON_2
    56           42          3  COLON_1
    57           41          3  LEUKEMIA
    58           40          3  LEUKEMIA
    59           39          3  LEUKEMIA
    60           38          3  LEUKEMIA
    61           37          3  LEUKEMIA
    62           36          3  K562A-re
    63           35          3  K562B-re
    64           34          3  LEUKEMIA

B34S Matrix Command Ending. Last Command reached.

Space available in allocator  119874394, peak  space  used    2681351
Number variables used                94, peak number  used        101
Number temp variables used            4, # user temp clean          0

B34S normal exit on Date (D:M:Y) 31/ 7/09   at Time (H:M:S)  7:41:38
```

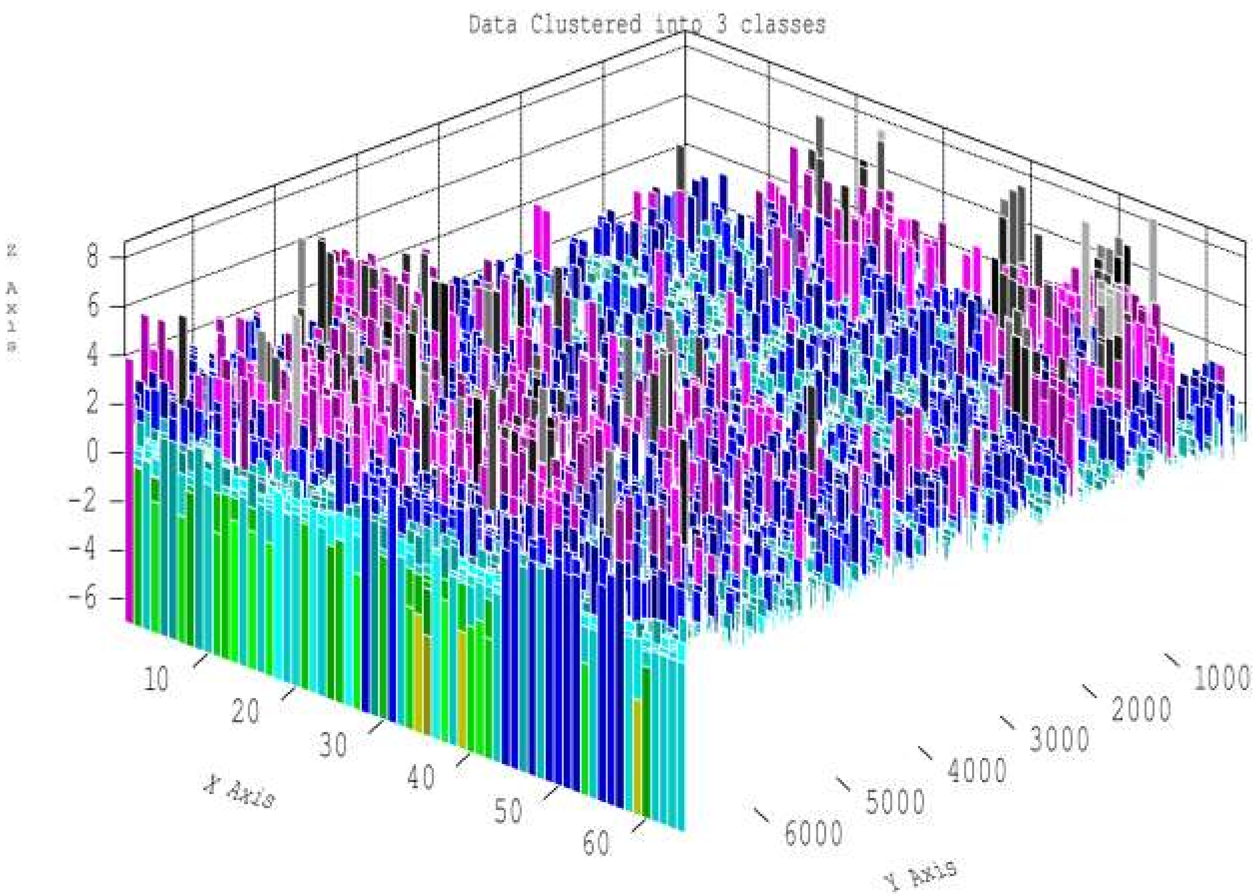



Figure 17.10  Human Tumor Data - Numbers in class 9,  34, 21

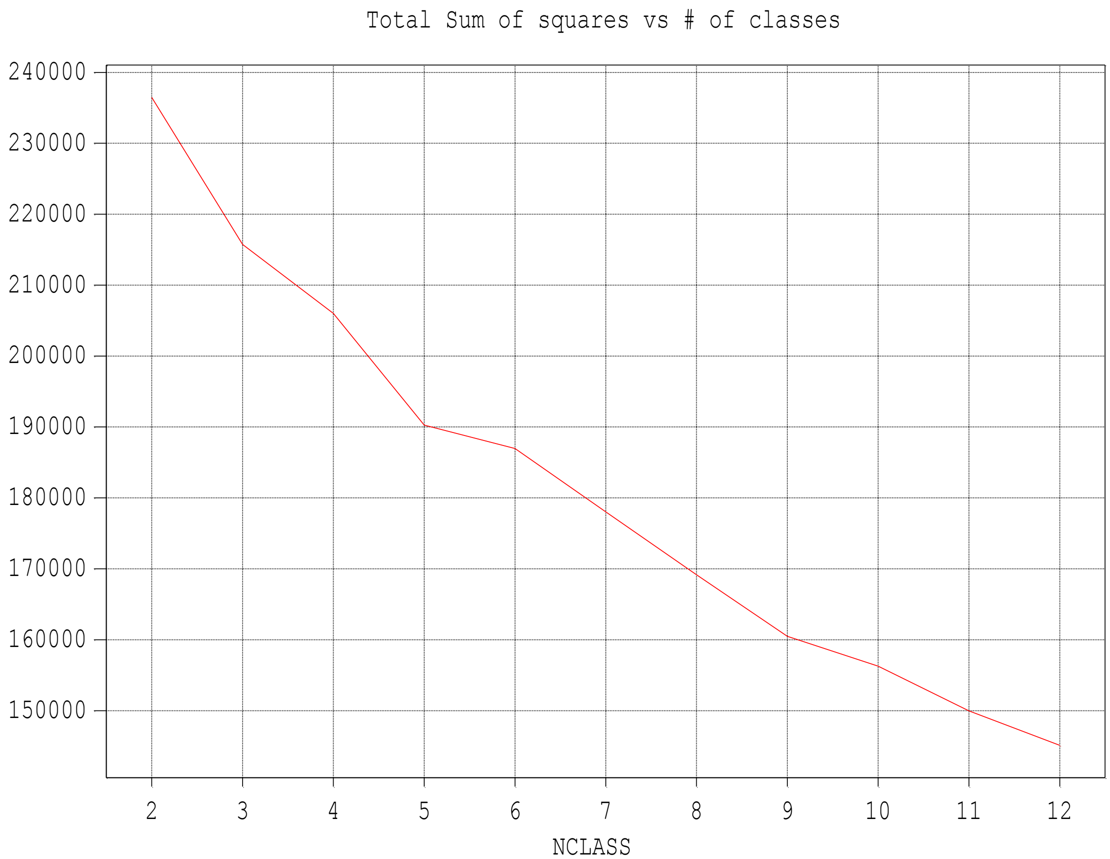



Figure 17.11 Analysis of the sum of squares in the range $k$=2 to $k$=12

## 17.7 Examples

The first example listed in Table 17.9 compares the performance of a number of estimation techniques for a 0-1 left-hand side variable using a dataset that was studied in Chapter 3 and in Chapter 14.

Table 17.9 Murder Data Estimated with Alternative Estimation Methods

```
/;
/; Murder Data estimated with:
/; OLS - PROBIT - RCOVER - RDA - PPREG - RANFOREST
/;
b34sexec options ginclude('b34sdata.mac') macro(murder)$
b34seend$
b34sexec matrix;
call loaddata;
call load(mvconfus :wbsuppl);
call echooff;

cases=dfloat(integers(0,1));

call olsq(d1 t y lf nw :print       );
call mvconf2(%y,%yhat,cases,
     'Tests on Murder Data using OLS Model',cfm,1);

call probit(d1 t y lf nw :print       );
call tabulate(%names,%lag,%coef,%se,%t);
call mvconf2(%y,%yhat,cases,
     'Tests on Murder Data using probit Model',cfm,1);

call rcover(d1 t y lf nw :print );
call mvconf2(%y,%yhat,cases,
     'Tests on Murder Data using rcover Model kd=10',
             cfm,1);

/; recode d1 1-2

d11=d1+1.;
call rda(d11 t y lf nw  :nk 2 :print );
%yhat_1=%yhat-1.;
call mvconf2(d1,%yhat_1,cases,'Tests on Murder Data using rda Model',
             cfm,1);

call ppreg(d11 t y lf nw  :class 2 :print);
%yhat_1=%yhat-1.;
call mvconf2(d1,%yhat_1,cases,'Tests on Murder Data using ppreg Model',
             cfm,1);

call print('Tests on Murder Data using Random Forest Model':);
call ranforest(d11 t y lf nw  :class 2
:print :maxtree 20 :vote_yhat);
b34srun;
```

Edited results of the script in Table 17.9 follow. After each estimation method a confusion matrix is displayed.

```
B34S 8.11D        (D:M:Y)   7/ 8/09 (H:M:S) 20:40:43   DATA STEP                  MCMANUS JPE 1985               PAGE    1

Variable    Label                                   # Cases      Mean        Std. Dev.    Variance      Maximum       Minimum
```

```
N         1 Observation Number                               44   22.5000      12.8452      165.000      44.0000      1.00000
M         2 Murder rate per 100,000 FBI est 1950             44   5.40364      4.46347      19.9225      19.2500      0.810000
PC        3 # convictions / Number of Murders 1950           44  0.260477     0.141703     0.200797E-01  0.757000     0.108000
PX        4 Av. # executions 46-50/# convictions             44  0.603409E-01  0.686291E-01  0.470995E-02  0.400000    0.00000
D1        5 Dummy. 1=> state has capital punish.             44  0.795455     0.408032     0.166490     1.00000      0.00000
T         6 Median time served convicted murders             44   136.523      61.6043      3795.09      298.000      34.0000
Y         7 Median family income (49) * 1000                 44   1.78091     0.396079     0.156878     2.39000      0.760000
LF        8 Labor force participation in 1950 in %           44   53.0659      2.48005      6.15067      58.8000      47.0000
NW        9 Proportion of population nonwhite                44  0.105591     0.113993     0.129944E-01  0.454000     0.300000E-
02
D2       10 Dummy variable. 1 for southern states            44  0.340909     0.479495     0.229915     1.00000      0.00000
D3       11 0=>d1=0,d3=1,1=>d1=1, 2+>PX GE .1                44  0.954545     0.608259     0.369979     2.00000      0.00000
CONSTANT 12                                                  44   1.00000      0.00000      0.00000      1.00000      1.00000

Number of observations in data file   44
Current missing variable code         1.000000000000000E+31

B34S(r) Matrix Command. d/m/y  7/ 8/09. h:m:s 20:40:43.

=>  CALL LOADDATA$

=>  CALL LOAD(MVCONFUS :WBSUPPL)$

=>  CALL ECHOOFF$

Ordinary Least Squares Estimation
Dependent variable                       D1
Centered R**2                            0.2801719877295442
Adjusted R**2                            0.2063434736505231
Residual Sum of Squares                  5.153314178754400
Residual Variance                        0.1321362609937025
Standard Error                           0.3635055171434163
Total Sum of Squares                     7.159090909090909
Log Likelihood                           -15.25320434099896
Mean of the Dependent Variable           0.7954545454545454
Std. Error of Dependent Variable         0.4080324573583922
Sum Absolute Residuals                   12.60627846049724
F( 4,       39)                          3.794902162458083
F Significance                           0.9893728105795514
1/Condition XPX                          6.859692723937969E-08
Maximum Absolute Residual                0.7625563180191952
Number of Observations                   44

Variable   Lag    Coefficient         SE              t
T           0      0.83418138E-03      0.96310865E-03   0.86613425
Y           0      0.53532854          0.23744998        2.2544897
LF          0     -0.61771228E-01      0.28786454E-01   -2.1458436
NW          0       2.5831207          0.70497968        3.6641067
CONSTANT    0       2.7333907           1.2982535        2.1054367

       --------------------------------------------------------------------------------
                                Confusion Matrix
                                Tests on Murder Data using OLS Model
       --------------------------------------------------------------------------------
                0        1  Total(A)    True%
       --------------------------------------------------------------------------------
       0        2        7        9   0.2222
       1        0       35       35   1.0000
       --------------------------------------------------------------------------------
Total(P)        2       42
   True%   1.0000   0.8333
```

The probit model is the base case from which to compare against.

```
Total Number of Cases         :        44
Number of Discrete Categories :         2
Accuracy Rate                 :    0.9646
Precision Rate                :    0.8674
Percent Correct               :    0.8409

Multivariate Probit Analysis (December 2004).

Dependent variable is D1

The iteration has converged.

1/ Cond of variance covariance of coef     7.669784078717216E-09
# of Iterations                            8
Log of likelihood function                 -9.016657779839512
Convergence tolerance                      9.999999747378752E-06

  **Summary of results**
                                                                    Partial Derivatives
Variable Lag  Max Likelihood       Est. Std. Error     t score     At Max Den.   At X Mean
T         0      0.94213373E-02     0.50703706E-02   1.8581161      0.37585698E-02  0.61991342E-05
Y         0       5.5510568          2.8121218        1.9739746       2.2145513      0.36525331E-02
LF        0     -0.43654090         0.25642135      -1.7024359     -0.17415462     -0.28723902E-03
NW        0       50.248507          20.088909        2.5013060       20.046254      0.33062954E-01
CONSTANT  0       10.267272          10.497241       0.97809244       4.0960490      0.67557499E-02
```

```
At point of means, E(dependent variable)   0.9998280530021426

# of observations                          44
# limits(=0)                               9
# nonlimits(=1)                            35
(-2.0) times the log likelihood ratio      26.55096552881705
Distributed as Chi squared with DF         4
Significance of Chi squared statistic      0.9999755024927519

Obs      %NAMES      %LAG        %COEF       %SE         %T
     1    T                   0   0.9421E-02  0.5070E-02   1.858
     2    Y                   0    5.551       2.812       1.974
     3    LF                  0  -0.4365      0.2564      -1.702
     4    NW                  0    50.25       20.09       2.501
     5    CONSTANT            0    10.27       10.50      0.9781

       --------------------------------------------------------------------------------
                                Confusion Matrix
                                Tests on Murder Data using probit Model
       --------------------------------------------------------------------------------
                0        1  Total(A)    True%
       --------------------------------------------------------------------------------
       0        7        2        9   0.7778
       1        1       34       35   0.9714
       --------------------------------------------------------------------------------
Total(P)        8       36
   True%   0.8750   0.9444


Total Number of Cases         :        44
Number of Discrete Categories :         2
Accuracy Rate                 :    0.9362
Precision Rate                :    0.9302
Percent Correct               :    0.9318
```

The recursive covering model for this example shows a confusion matrix that is inferior to the probit model but for this example is better than the regularized discriminate analysis approach.

```
Recursive Covering Program called:
Cover Tree Dimension (mxm)                        10000
Local Terminal Model Dimension (mxt)              10000
Internak itrm work array (mxi)                    10000
Internal rtrm work array (mxr)                    10000
Internal mxd work array                           10000
Number of Observations                            44
Number of right hand side variables               4
Approvimate method used to calculate yhat
lin=2 =>local constant model- linear combination splits
# training obs. in each terminal region (kd)      10
Trimming factor for splits (dtf)                  0.2500000000000000
Ridge weight decay factor (rdg)                   1.000000000000000E-02
Number of Terminal Nodes in the cover Tree        25
Terminal node storage requirement (mxt)           10000

Residual sum of squares                           5.077755102040816
Total    sum of squares                           7.159090909090909
Mean of the Dependent Variable                    0.7954545454545454
Std. Error of Dependent Variable                  0.4080324573583922
Sum Absolute Residuals                            12.47142857142857
Maximum Absolute Residual                         0.7000000000000000
Residual Variance                                 0.1269438775510204

       --------------------------------------------------------------------------------
                                Confusion Matrix
                                Tests on Murder Data using rcover Model kd=10
       --------------------------------------------------------------------------------
                0        1  Total(A)    True%
       --------------------------------------------------------------------------------
       0        6        3        9   0.6667
       1        3       32       35   0.9143
       --------------------------------------------------------------------------------
Total(P)        9       35
   True%   0.6667   0.9143


Total Number of Cases         :        44
Number of Discrete Categories :         2
Accuracy Rate                 :    0.8636
Precision Rate                :    0.8636
Percent Correct               :    0.8636

Regularized Discriminate Analysis
Number of classes  (nk)                           2
Number of Observations                            44
Number of right hand side variables               4
Lower search limit              (lamda(1))        0.000
Upper search limit              (lamda(2))        1.000
Number of search locations      (lamda(3))        5.000
Exp for power transformation    (lamda(4))        1.000
Covariance Shrinkage lower limit (gamma(1))       0.000
```

```
Covariance Shrinkage upper limit (gamma(2))     1.000
Number of Locations              (gamma(3))     10.00
Exp for power transformation     (gamma(4))     1.000

Left Hand Side Variable                         D11

Series           Mean          Max           Min
D11             1.795         2.000         1.000

Right Hand Side Variables

Series   Lag      Mean          Max           Min
T         0     136.5         298.0         34.00
Y         0     1.781         2.390        0.7600
LF        0     53.07         58.80         47.00
NW        0    0.1056        0.4540        0.3000E-02

Miss classification Loss Matrix

             1              2
    1    0.00000       0.500000
    2   0.500000        0.00000

Prior probability of Each Class

             1              2
        0.500000       0.500000


Size of %sp array                               398
Size of %dp array                               76
Covariance mixing parameter  (lambda).          1.000000000000000
Covariance shrinkage parameter (gamma).         0.3333333333333333
Cross-validated est. of misclassification risk  0.3071428571428573
Residual sum of squares                         22.00000000000000
Total    sum of squares                         7.159090909090909
Mean of the Dependent Variable                  1.795454545454545
Std. Error of Dependent Variable                0.4080324573583922
Sum Absolute Residuals                          22.00000000000000
Maximum Absolute Residual                       1.000000000000000
Residual Variance                               0.5500000000000000

       --------------------------------------------------------------------------------
                                Confusion Matrix
                                Tests on Murder Data using rda Model
       --------------------------------------------------------------------------------
                0        1  Total(A)    True%
       --------------------------------------------------------------------------------
       0        3        6        9   0.3333
       1       16       19       35   0.5429
       --------------------------------------------------------------------------------
Total(P)       19       25
   True%   0.1579   0.7600


Total Number of Cases         :        44
Number of Discrete Categories :         2
Accuracy Rate                 :    0.4524
Precision Rate                :    0.6368
Percent Correct               :    0.5000
```

The Projection Pursuit Model produces superior accuracy with only one error. There were 9 states with no capital punishment. All 9 were identified correctly. There were 35 states with capital punishment, one of which was incorrectly identified as not having capital punishment.

```
Projection Pursuit Regression
Classification option called.

Number of Observations                          44
Number of right hand side variables             5
Maximum number of trees                         20
Minimum number of trees                         20
Number of classes in left hand side variable    2
Level of fit                                    2
Max number of Primary   Iterations (maxit)      200
Max number of Secondary Iterations (mitone)     200
Number of cj Iterations (mitcj)                 10
Smoother tone control (alpha)                   0.000000000000000E+00
Span                                            0.000000000000000E+00
Convergence (CONV) set as                       5.000000000000000E-03
Left Hand Side Variable                         D11

Series           Mean          Max           Min
D11             1.795         2.000         1.000

Right Hand Side Variables

Series   Lag      Mean          Max           Min
T         0     136.5         298.0         34.00
Y         0     1.781         2.390        0.7600
LF        0     53.07         58.80         47.00
NW        0    0.1056        0.4540        0.3000E-02
CONSTANT  0     1.000         1.000         1.000
```

```
Given # of trees                                 20
# primary   iterations used                      1
# secondary iterations used                      4
# cj iterations used                             10
Number of miss-Classifications                   1.000000000000000
Error Rate                                       2.272727272727273E-02
Mean of the Dependent Variable                   1.795454545454545
Std. Error of Dependent Variable                 0.4080324573583922

Variable Importance for Model with # Trees  20
 Series Number       Importance
         4             1.00000
         3            0.969093
         2            0.722533
         1            0.566720
         5             0.00000


       --------------------------------------------------------------------------------
                                Confusion Matrix
                                Tests on Murder Data using ppreg Model
       --------------------------------------------------------------------------------
                0        1  Total(A)    True%
       --------------------------------------------------------------------------------
       0        9        0        9   1.0000
       1        1       34       35   0.9714
       --------------------------------------------------------------------------------
Total(P)       10       34
   True%   0.9000   1.0000


Total Number of Cases         :        44
Number of Discrete Categories :         2
Accuracy Rate                 :    0.9779
Precision Rate                :    0.9795
Percent Correct               :    0.9773
Tests on Murder Data using Random Forest Model
```

The random forest approach used 20 trees and using voting provided 100% accuracy.

```
Random Forest Analysis Ver. 3.1 - 30 May 2009 build
Classification option called.

Number of Observations                             44
Number of right hand side variables                4
Maximum number of trees (maxtree)                  20
Maximum number of nodes (nrnodes)                  89
Number of Variables to select at each node (mtry)  2
Number left hand variable classes (nclass)         2
Minimum node size (ndsize)                         1
Left Hand Side Variable                          D11

Series            Mean          Max           Min
D11              1.795         2.000         1.000

Right Hand Side Variables

  # Series   Lag      Mean          Max           Min
  1 T         0     136.5          298.0        34.00
  2 Y         0     1.781          2.390       0.7600
  3 LF        0     53.07          58.80        47.00
  4 NW        0    0.1056         0.4540       0.3000E-02

Error rate for bagged dataset as a f of tree used (errtr)

             1             2             3             4             5             6             7             8
             9            10            11            12            13            14            15            16
            17            18            19            20
       4.54545       13.6364       18.1818       18.1818       18.1818       22.7273       25.0000       22.7273
       22.7273       20.4545       20.4545       20.4545       22.7273       25.0000       20.4545       20.4545
       20.4545       22.7273       20.4545       20.4545

Error rate for out-of-bag dataset as a f of tree used (errc)

             1             2             3             4             5             6             7             8
             9            10            11            12            13            14            15            16
            17            18            19            20
       15.3846       33.3333       21.4286       21.4286       23.5294       25.0000       25.0000       11.7647
       17.6471       11.7647       33.3333       6.25000       37.5000       26.3158       16.6667       37.5000
       12.5000       33.3333       35.2941       13.3333


Confusion Matrix calculated using votes from # Trees =   20
Element x(i,j) => classify i as j.

          1     2

    1     9     0
    2     0    35

# Miss Classified obs from Confusion matrix       0
# Correctly Classified obs from Confusion matrix  44
100.* (# Miss classified / Total # Obs)           0.000000000000000E+00
```

```
# of miss-classified obs in bagged dataset        6
# of observations in bagged dataset               29
(# miss-classified / # of observations) in %      20.45454545454546
Error rate for out of bag sample data in %        13.33333333333333


Last tree Confusion Matrix. Element x(i,j) => classify i as j.

          1     2

    1     8     1
    2     1    34


Total Correctly Classified = x(i,i)/nob

             1             2
        0.181818       0.772727


Class Error= 1.-((# correct- # wrong)/# correct)

             1             2
        0.125000       0.294118E-01

# Miss Classified obs from Confusion matrix       2
# Correctly Classified obs from Confusion matrix  42
100.* (# Miss classified / Total # Obs)           4.545454545454546

B34S Matrix Command Ending. Last Command reached.

Space available in allocator    8856736, peak  space  used      67102
Number variables used               163, peak number  used        163
Number temp variables used         1856, # user temp clean          0
```

The next example uses the famous Thurber data that is a part of the NIST nonlinear test suite of difficult non-linear problems. The B34S files `stattest.mac` and `stattest2.mac` contain all these datasets with the certified answers and as well as the B34S and in many cases RATS solutions. In the example code in Table 17.10 the exact answers as supplied by NIST/ITL are given in the program script to aid in the evaluation of the results. Edited output from running this example is given below.

Table 17.10 Analysis of the Nonlinear Thurber Data

```
b34sexec options copyf(4,6,1,999999,1,80,0,1);
datacards;
NIST/ITL StRD
Dataset Name:  Thurber          (Thurber.dat)

File Format:   ASCII
               Starting Values   (lines 41 to 47)
               Certified Values  (lines 41 to 52)
               Data              (lines 61 to 97)

Procedure:     Nonlinear Least Squares Regression

Description:   These data are the result of a NIST study involving
               semiconductor electron mobility.  The response
               variable is a measure of electron mobility, and the
               predictor variable is the natural log of the density.


Reference:     Thurber, R., NIST (197?).
               Semiconductor electron mobility modeling.

Data:          1 Response Variable  (y = electron mobility)
               1 Predictor Variable (x = log[density])
               37 Observations
               Higher Level of Difficulty
               Observed Data
```

```
Model:         Rational Class (cubic/cubic)
               7 Parameters (b1 to b7)

               y = (b1 + b2*x + b3*x**2 + b4*x**3) /
                   (1 + b5*x + b6*x**2 + b7*x**3)  +  e


          Starting Values                  Certified Values

        Start 1     Start 2           Parameter     Standard Deviation
  b1 =   1000        1300         1.2881396800E+03  4.6647963344E+00
  b2 =   1000        1500         1.4910792535E+03  3.9571156086E+01
  b3 =    400         500         5.8323836877E+02  2.8698696102E+01
  b4 =     40          75         7.5416644291E+01  5.5675370270E+00
  b5 =      0.7         1         9.6629502864E-01  3.1333340687E-02
  b6 =      0.3         0.4       3.9797285797E-01  1.4984928198E-02
  b7 =      0.03        0.05      4.9727297349E-02  6.5842344623E-03

Residual Sum of Squares:                    5.6427082397E+03
Residual Standard Deviation:                1.3714600784E+01
Degrees of Freedom:                                30
Number of Observations:                            37
b34sreturn;
b34seend;

b34sexec data heading('Thurber Data');
input y x;
datacards;
  80.574E0  -3.067E0    84.248E0  -2.981E0
  87.264E0  -2.921E0    87.195E0  -2.912E0
  89.076E0  -2.840E0    89.608E0  -2.797E0
  89.868E0  -2.702E0    90.101E0  -2.699E0
  92.405E0  -2.633E0    95.854E0  -2.481E0
 100.696E0  -2.363E0   101.060E0  -2.322E0
 401.672E0  -1.501E0   390.724E0  -1.460E0
 567.534E0  -1.274E0   635.316E0  -1.212E0
 733.054E0  -1.100E0   759.087E0  -1.046E0
 894.206E0  -0.915E0   990.785E0  -0.714E0
1090.109E0  -0.566E0  1080.914E0  -0.545E0
1122.643E0  -0.400E0  1178.351E0  -0.309E0
1260.531E0  -0.109E0  1273.514E0  -0.103E0
1288.339E0   0.010E0  1327.543E0   0.119E0
1353.863E0   0.377E0  1414.509E0   0.790E0
1425.208E0   0.963E0  1421.384E0   1.006E0
1442.962E0   1.115E0  1464.350E0   1.572E0
1468.705E0   1.841E0  1447.894E0   2.047E0
1457.628E0   2.200E0
b34sreturn;
b34seend;
/$ Illustrates Nonlinear Estimation using NLLSQ Command under matrix

b34sexec matrix;
call loaddata;

*          Starting Values              Certified Values               ;
*      Start 1     Start 2           Parameter     Standard Deviation ;
*  b1 =   1000        1300       1.2881396800E+03  4.6647963344E+00   ;
*  b2 =   1000        1500       1.4910792535E+03  3.9571156086E+01   ;
*  b3 =    400         500       5.8323836877E+02  2.8698696102E+01   ;
*  b4 =     40          75       7.5416644291E+01  5.5675370270E+00   ;
*  b5 =      0.7         1       9.6629502864E-01  3.1333340687E-02   ;
*  b6 =      0.3         0.4     3.9797285797E-01  1.4984928198E-02   ;
*  b7 =      0.03        0.05    4.9727297349E-02  6.5842344623E-03   ;
```

```
ans=matrix(7,2:  1.2881396800E+03,  4.6647963344E+00,
                 1.4910792535E+03,  3.9571156086E+01,
                 5.8323836877E+02,  2.8698696102E+01,
                 7.5416644291E+01,  5.5675370270E+00,
                 9.6629502864E-01,  3.1333340687E-02,
                 3.9797285797E-01,  1.4984928198E-02,
                 4.9727297349E-02,  6.5842344623E-03);
testss =  5.6427082397E+03;

program test;
call echooff;
yhat = (b1 + b2*x + b3*(x**2.) + b4*(x**3.)) /
       (1. + b5*x + b6*(x**2.) + b7*(x**3.))     ;
r=y-yhat;
call outstring(3, 1,'b1 b2 b3');
call outdouble(14,1,b1);
call outdouble(34,1,b2);
call outdouble(54,1,b3);
call outstring(3, 2,'b4 b5 b6');
call outdouble(14,2,b4);
call outdouble(34,2,b5);
call outdouble(54,2,b6);
call outstring(3, 3,'b7');
call outdouble(14,3,b7);
return;
end;

call print(test);
call cls(-1);

/; Try OLS as if teh model was not known

call olsq(y x :print);

call nllsq(y,yhat :name test :parms b1 b2 b3 b4 b5 b6 b7
         :ivalue array(: 1000. 1000. 400. 40. .7 .3 .03)
/$       :ivalue array(: 1300. 1500. 500. 75. 1. .4 .05)
/$       :diff   array(: .0001 .0001 .0001 .0001 .0001 .0001 .0001)
         :maxit 500  :eps2 .0001
         :print result);
call print('NLLSQ on THURBER start # 1':);
call lre(ans(,1),15,%coef,lretest,bits :print);
call print('SE ':);
call lre(ans(,2),15,%se,  lretest,bits :print);
call print('Residual sum of squares':);
call lre(testss,15,%fss,  lretest,bits :print);
call print('  ':);

/; Alternative Nonlinear least squares program used

call nl2sol(r :name test :parms b1 b2 b3 b4 b5 b6 b7
         :ivalue array(: 1000. 1000. 400. 40. .7 .3 .03)
/$       :ivalue array(: 1300. 1500. 500. 75. 1. .4 .05)
         :maxit 500 :print);
call print('NL2SOL on THURBER start # 1':);
call lre(ans(,1),15,%coef,lretest,bits :print);
call print('SE ':);
call lre(ans(,2),15,%se,  lretest,bits :print);
call print('Residual sum of squares':);
call lre(testss,15,%fss,  lretest,bits :print);
call print('  ':);
/; Now do Exploritory PP

call load(ppexp_p);
```

```
/; sets number of solutions
mm=5;

/; sets order of legenre
jj=2;

fei=.1e-4;
nei = 1;
trm=.1;
itype=0;
call ppexp(y x
           :mm mm
           :jj jj
/;         :fei fei
/;         :nei nei
/;         :trm trm
/;         :print
           );
call print('ppexp Index of Thurber Data',%ppindex);
ppi_m1=%ppindex;

call ppexp_p(%xpa,%mm,%nob,itype,'a',%ppindex);

/; Now do Projection Pursuit to see if we get close to certified e'e

call ppreg(y x :mu 2 :m 10 :print);

b34srun$
```

```
Variable      # Cases       Mean          Std Deviation       Variance          Maximum           Minimum

Y          1       37   783.2101081       564.3487382       318489.4984       1468.705000       80.57400000
X          2       37 -0.8630270270       1.608668160       2.587813249       2.200000000      -3.067000000
CONSTANT   3       37   1.000000000       0.000000000       0.000000000       1.000000000       1.000000000

Number of observations in data file   37
Current missing variable code         1.000000000000000E+31

B34S(r) Matrix Command. d/m/y  8/ 8/09. h:m:s 12: 4:51.
```

The true model is

$$y = \frac{(\beta_1 + \beta_2 x + \beta_3 x^2 + \beta_4 x^3)}{(1 + \beta_5 x + \beta_6 x^2 + \beta_7 x^3)} + e \qquad (17.7\text{-}1)$$

Assuming no knowledge of the model the experimenter estimates an OLS model $y = a + bx$ and receives finds large and significant $t$ values and an adjusted R**2 of .919. Without further tests, many researchers might be led to select this model.

```
=>  CALL OLSQ(Y X :PRINT)$


Ordinary Least Squares Estimation
Dependent variable                    Y
Centered R**2                         0.9210137417351627
Adjusted R**2                         0.9187569914990245
Residual Sum of Squares               905626.5757730267
Residual Variance                     25875.04502208648
Standard Error                        160.8572193657670
Total Sum of Squares                  11465621.94067357
Log Likelihood                        -239.4518174873871
Mean of the Dependent Variable        783.2101081081081
Std. Error of Dependent Variable      564.3487382390817
Sum Absolute Residuals                4686.054831692326
F( 1,       35)                       408.1150527810371
F Significance                        1.000000000000000
1/Condition XPX                       0.1783517554098133
```

```
Maximum Absolute Residual             356.8345175380143
Number of Observations                37

Variable   Lag    Coefficient         SE             t
X           0      336.67754          16.665672       20.201858
CONSTANT    0      1073.7719          30.103058       35.669862
```

Using full knowledge of the correct model (17.7-1) nonlinear modeling is attempted using two nonlinear solvers. NLLSQ does best with SE's and is close on the coefficients and $e'e$.

```
=>  CALL NLLSQ(Y,YHAT :NAME TEST :PARMS B1 B2 B3 B4 B5 B6 B7
=>           :IVALUE ARRAY(: 1000. 1000. 400. 40. .7 .3 .03)
=>           :MAXIT 500  :EPS2 .0001
=>           :PRINT RESULT)$


Nonlinear Estimation using NLLSQ
# of observations                                37
# parameters                                     7
Max iterations                                   500
Starting Lamda (FLAM)                            1.000000000000000E-02
Starting FLU                                     10.00000000000000
Maximum relative change in sum squares (eps1)    0.000000000000000E+00
Maximum relative change in each parm.  (eps2)    1.000000000000000E-04

Initial Parameter Values (TH)
       1           2           3           4           5           6           7
   1000.       1000.       400.0       40.00       0.7000      0.3000      0.3000E-01

Proportions used in calculating difference quotients
       1           2           3           4           5           6           7
  0.1000E-01  0.1000E-01  0.1000E-01  0.1000E-01  0.1000E-01  0.1000E-01  0.1000E-01

Sign restriction vector (GT 0.0 means restricted
       1           2           3           4           5           6           7
   0.000       0.000       0.000       0.000       0.000       0.000       0.000


=>  CALL ECHOOFF$

Initial sum of squares                           4528124.603575195
Number of observations =                         37

Iteration stops - % change in each parm. LE      1.000000000000000E-04

Correlation Matrix of Estimated Parameters.
       1           2           3           4           5           6           7
   1  1.0000
   2 -0.0250      1.0000
   3 -0.0722      0.9955      1.0000
   4 -0.0760      0.9892      0.9977      1.0000
   5 -0.1534      0.9754      0.9799      0.9688      1.0000
   6  0.0329      0.9611      0.9700      0.9687      0.9583      1.0000
   7  0.1150      0.8445      0.8264      0.8395      0.7298      0.7748      1.0000

Normalizing Elements
       1           2           3           4           5           6           7
  0.3395       3.145       2.272      0.4420      0.2488E-02  0.1243E-02  0.4880E-03

Variance of residuals         188.2630298810194
Sum of squared residuals      5647.890896430582
Standard Error of Estimate    13.72089756105698
Adjusted R Square             0.9994088877942440
Degrees of freedom            30
Number of Iterations          19
1/Condition of Hessian        7.896837694340554E-11
Durbin Watson                 1.749061959524842

#   Name      Coefficient         Standard Error      T Value
  1 B1          1288.1840           4.6588644           276.50172
  2 B2          1485.2143           43.152713           34.417634
  3 B3          578.90243           31.179495           18.566768
  4 B4          74.581515           6.0640092           12.299044
  5 B5         0.96146230          0.34135009E-01       28.166458
  6 B6         0.39561751          0.17052001E-01       23.200650
  7 B7         0.49041610E-01      0.66957065E-02       7.3243368

Note: Confidence limits for each parameter on linear hypothesis.

NLLSQ on THURBER start # 1
Test    1288.139680000000      Ans:    1288.184022734966      LRE       4.46 # Bits   14.83
Test    1491.079253500000      Ans:    1485.214282093723      LRE       2.41 # Bits    7.99
Test    583.2383687700000      Ans:    578.9024324896610      LRE       2.13 # Bits    7.07
Test    75.41664429100000      Ans:    74.58151528866100      LRE       1.96 # Bits    6.50
Test   0.9662950286400001      Ans:   0.9614622967528016      LRE       2.30 # Bits    7.64
Test   0.3979728579700000      Ans:   0.3956175116035232      LRE       2.23 # Bits    7.40
Test   0.4972729734900000E-01  Ans:   0.4904160975572151E-01  LRE       1.86 # Bits    6.18

Mean      LRE    2.477433775001469
Variance  LRE    0.8026881336939621
Minimum   LRE    1.860468564728016
Maximum   LRE    4.463140482526512
```

```
SE
Test    4.664796334400000      Ans:    4.658864406877001      LRE       2.90 # Bits    9.62
Test    39.57115608600000      Ans:    43.15271309684226      LRE       1.04 # Bits    3.47
Test    28.69869610200000      Ans:    31.17949477446251      LRE       1.06 # Bits    3.53
Test    5.567537027000000      Ans:    6.064009152401098      LRE       1.05 # Bits    3.49
Test   0.3133334068700000E-01  Ans:   0.3413500929329785E-01  LRE       1.05 # Bits    3.48
Test   0.1498492819800000E-01  Ans:   0.1705200130440376E-01  LRE       0.00 # Bits    0.00
Test   0.6584234462300000E-02  Ans:   0.6695706500614951E-02  LRE       1.77 # Bits    5.88

Mean      LRE    1.267415984543630
Variance  LRE    0.7830901357081754
Minimum   LRE    0.000000000000000E+00
Maximum   LRE    2.895636851017459
Residual sum of squares
Test    5642.708239700000      Ans:    5647.890896430582      LRE       3.04 # Bits   10.09

Nonlinear Estimation using NL2SNO - Analytic Jacobian
Sum of squared Residuals      5642.708239667100
Residual Variance             188.0902746555700
Residual Standard Error       13.71460078367468
# of parameters               7
# of residuals                37
# of iterations               13
# of function evaluations     51
# of gradiant evaluations     14
# of Covariance evaluations   0
Relative Function Tolerance   1.000000000000000E-10
Finite-Difference factor      1.489370874967368E-08
Absolute Function Tolerance   9.999999999999999E-21
False Convergence Tolerance   2.220446049250313E-14
X-Convergence Tolerance       1.489370874967368E-08
2-norm of scaled gradiant     6.649898982036056E-07
2-norm of scaled step size    1.083346056658166E-02
1. / Condition of Hessian     3.524355883023316E-03

*** relative function convergence ***

 #  Name      Coefficient        Standard Error      T Value
  1 B1          1288.1397          4.7875346          269.06117
  2 B2          1491.0793          31.690627          47.051113
  3 B3          583.23838          25.858797          22.554738
  4 B4          75.416646          4.8400104          15.581918
  5 B5         0.96629504         0.35874037E-01      26.935777
  6 B6         0.39797286         0.17357346E-01      22.928209
  7 B7         0.49727298E-01     0.29431521E-02      16.895932

SE calculated as sqrt |diagonal(Covariance Matrix)|

Hessian Matrix

            1             2             3             4             5             6             7
    1    22.9205       -28.2417       1004.30       -30.5894       814.284       668.677       -5.87818
    2   -28.2417        1004.30      -30.5894        814.284       668.677      -5.87818        150.806
    3    1004.30       -30.5894       814.284        668.677      -5.87818       150.806        124.761
    4   -30.5894        814.284       668.677       -5.87818       150.806       124.761        23.4257
    5    814.284        668.677      -5.87818        150.806       124.761       23.4257     -0.543527E-01
    6    668.677       -5.87818       150.806        124.761       23.4257    -0.543527E-01    1.11903
    7   -5.87818        150.806       124.761        23.4257    -0.543527E-01    1.11903       0.914679


Gradiant Vector

    -0.807272E-06  -0.801616E-06  -0.242463E-06  -0.141157E-05  -0.731236E-03   0.233414E-02  -0.141092E-01

Scale Vector

      14.4391        31.7693        83.2829        229.047        6992.26        10701.8        22891.2


NL2SOL on THURBER start # 1
Test    1288.139680000000      Ans:    1288.139679308184      LRE       9.27 # Bits   30.79
Test    1491.079253500000      Ans:    1491.079265167933      LRE       8.11 # Bits   26.93
Test    583.2383687700000      Ans:    583.2383776783812      LRE       7.82 # Bits   25.96
Test    75.41664429100000      Ans:    75.41664597800754      LRE       7.65 # Bits   25.41
Test   0.9662950286400001      Ans:   0.9662950395611042      LRE       7.95 # Bits   26.40
Test   0.3979728579700000      Ans:   0.3979728626426217      LRE       7.93 # Bits   26.34
Test   0.4972729734900000E-01  Ans:   0.4972729790975419E-01  LRE       7.95 # Bits   26.40

Mean      LRE    8.095405018883538
Variance  LRE    0.2878266408888578
Minimum   LRE    7.650350180682473
Maximum   LRE    9.269972491440374
SE
Test    4.664796334400000      Ans:    4.787534626326338      LRE       1.58 # Bits    5.25
Test    39.57115608600000      Ans:    31.69062669410578      LRE       0.00 # Bits    0.00
Test    28.69869610200000      Ans:    25.85879652014536      LRE       1.00 # Bits    3.34
Test    5.567537027000000      Ans:    4.840010436044693      LRE       0.00 # Bits    0.00
Test   0.3133334068700000E-01  Ans:   0.3587403650289666E-01  LRE       0.00 # Bits    0.00
Test   0.1498492819800000E-01  Ans:   0.1735734631550366E-01  LRE       0.00 # Bits    0.00
Test   0.6584234462300000E-02  Ans:   0.2943152053455832E-02  LRE       0.00 # Bits    0.00

Mean      LRE    0.3692016848023297
Variance  LRE    0.4251507060910087
Minimum   LRE    0.000000000000000E+00
Maximum   LRE    1.579852611964675
Residual sum of squares
Test    5642.708239700000      Ans:    5642.708239667101      LRE      11.23 # Bits   37.32
```

NL2SOL finds 11 digits of the answer to $e'e$ of `5.6427082397E+03`, and roughly 8-9 digits of the answer to the estimated coefficients. Exploratory projection pursuit analysis finds a large drop off in the dimensionality of the nonlinearity between 1 to 2 which is shown in `%ppindex` falling from `.032283` to `.00013391`. The projection pursuit estimation results list $e'e$ for the # of trees going from 2 to 10 and suggest that there are no gains after 5. What is most interesting is that for the projection pursuit model $e'e$ was `4951.736522135388` which is less than the NIST answers. Assuming only continuous derivatives, projection pursuit by being a universal approximator is able to map/model a nonlinear function without assumptions on the error distribution or the functional form. Information on the model is contained not in estimated mathematical equations but in the leverage plots over relevant ranges of the explanatory variables.

```
         ppexp Index of Thurber Data

%PPINDEX= Vector of        5   elements

     0.322830E-01   0.133914E-03   0.133914E-03   0.133914E-03   0.133914E-03


Projection Pursuit Regression

Number of Observations                          37
Number of right hand side variables             2
Maximum number of trees                         10
Minimum number of trees                         2
Number of left hand side variables              1
Level of fit                                    2
Max number of Primary   Iterations (maxit)      200
Max number of Secondary Iterations (mitone)     200
Number of cj Iterations (mitcj)                 10
Smoother tone control (alpha)                   0.000000000000000E+00
Span                                            0.000000000000000E+00
Convergence (CONV) set as                       5.000000000000000E-03
Left Hand Side Variable                         Y

Series           Mean         Max          Min
Y               783.2        1469.        80.57

Right Hand Side Variables

Series   Lag      Mean         Max          Min
X          0   -0.8630        2.200       -3.067
CONSTANT   0     1.000        1.000        1.000

  MU       RSS         SUMARES          MAXRES
   2    5509.60       342.253         40.8920
   3    5010.58       298.260         42.6239
   4    4965.24       296.773         42.1087
   5    4951.74       297.999         41.8779
   6    4951.74       297.999         41.8779
   7    4951.74       297.999         41.8779
   8    4951.74       297.999         41.8779
   9    4951.74       297.999         41.8779
  10    4951.74       297.999         41.8779

Given # of trees                                10
# primary   iterations used                     1
# secondary iterations used                     2
# cj iterations used                            2
Residual sum of squares                         4951.736522135388
Total    sum of squares                         11465621.94067357
Mean of the Dependent Variable                  783.2101081081081
Std. Error of Dependent Variable                564.3487382390817
Sum Absolute Residuals                          297.9987391195669
Maximum Absolute Residual                       41.87785025241669
Residual Variance                               141.4781863467254
B34S Matrix Command Ending. Last Command reached.

Space available in allocator    8856412, peak  space  used      14139
Number variables used                96, peak number  used        123
Number temp variables used         8507, # user temp clean         0B34S 8.11D         (D:M:Y)   8/ 8/09 (H:M:S) 12: 4:56   DATA
```

Table 17.11 Testing OLS, MARS, GAM, PPEXP and PPREG

---

```
/; Illustrates Nonlinear Modeling
/; one nonlinear series. One linear series
/; Experiment with settings

/; Suggested use:
/; First   try mod=1,  mod=2, mod=3
/; Next experiment with bend setting
/; Finally experiment with Coef and noise settings
/;
b34sexec matrix;
call load(contrib);
call echooff;
call contribi;

program nonltest;
x=rn(array(n:));
z=rn(array(n:));
/;
/; alternative models
/;
if(mod.eq.1)y=coef1*cos(x**bend) +coef2*z+coef3+noise*rn(array(n:));
if(mod.eq.2)y=coef1*dlog(abs(x**bend))
                                 +coef2*z+coef3+noise*rn(array(n:));
if(mod.eq.3)y=coef1*(x**bend)    +coef2*z+coef3+noise*rn(array(n:));
/;
/; specific settings
/;
_mi=3;
_m=10;
iols=3;
isave=0;
call character(fsv_info,'nonltest Model');
call character(l_hand_s,'y');
call character(_args,   'x z');
call character(_argsg,  'x[predictor,3] z[predictor,3]');
call contribl;
call contribd;
return;
end;

bend=2.;
coef1=10.;
coef2=10.;
coef3=10.;
n=1000;
noise=1.;
mod=2;
do_ppexp=1;

/; fit case

call nonltest;

/; perfect fit case

noise=0.;
/; call nonltest;
b34srun;
```

Table 17.11 contains a testing setup to allow experimentatioin with various models. Three setups are shown, but more could easily be generated. The model is $y = f(x,z) + e$ where $x$ is usually nonlinear and $z$ is always linear. The script generates leverage plots. If mod=1 then

$$y = \beta_0 + \beta_1 \cos(x^{\gamma}) + \beta_2 z + \delta u \tag{17.7-2}$$

where $\delta$ amplifies the noise and $\gamma$ builds nonlinearity into $x$. If mod=2 then

$$y = \beta_0 + \beta_1 \ln|(x^{\gamma})| + \beta_2 z + \delta u\ ) \tag{17.7-3}$$

while if mod=3

$$y = \beta_0 + \beta_1 (x^{\gamma}) + \beta_2 z + \delta u \tag{17.7-4}$$

Figures 17.12 and 17.13 shows the leverage plots for $x$ and $z$ respectively for mod=2. The reader is invited to experiment with other settings.

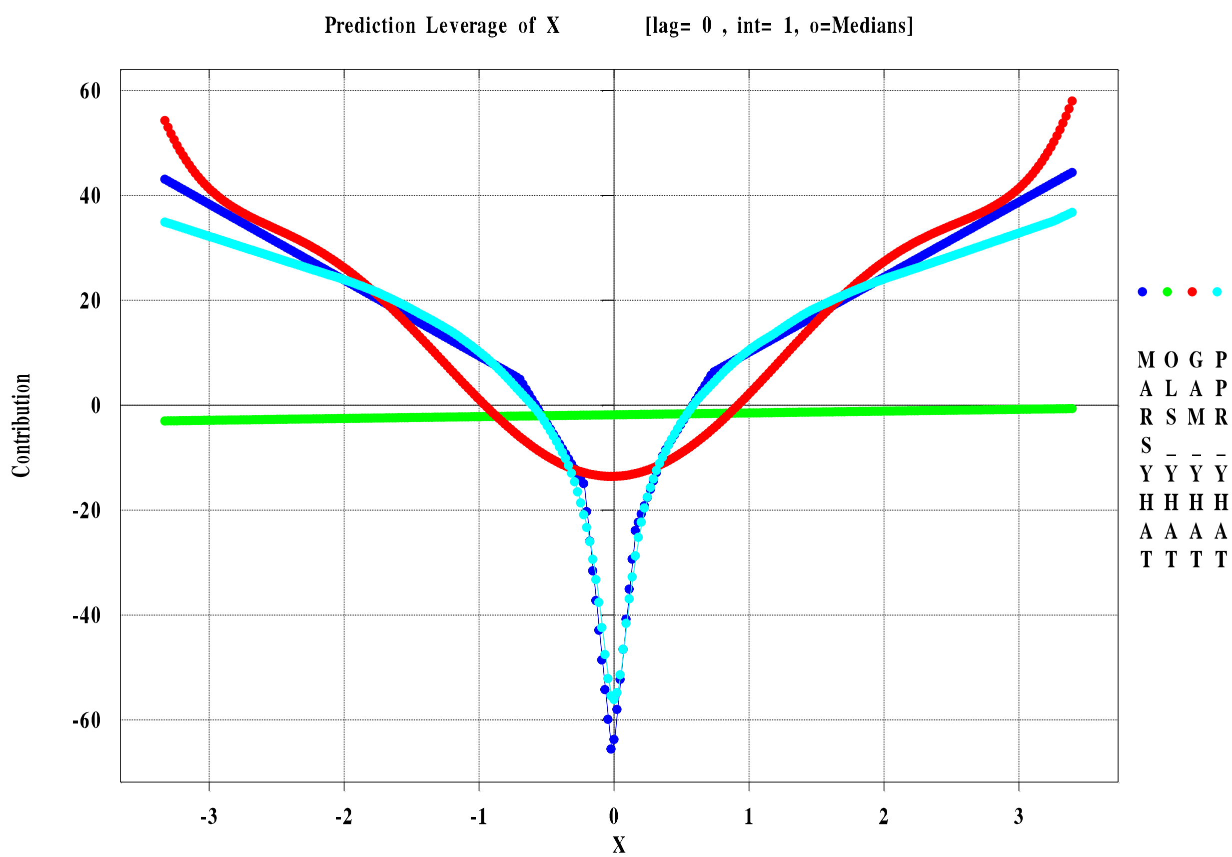



Figure 17.12 Leverage Plot of Nonlinear term for medians of data

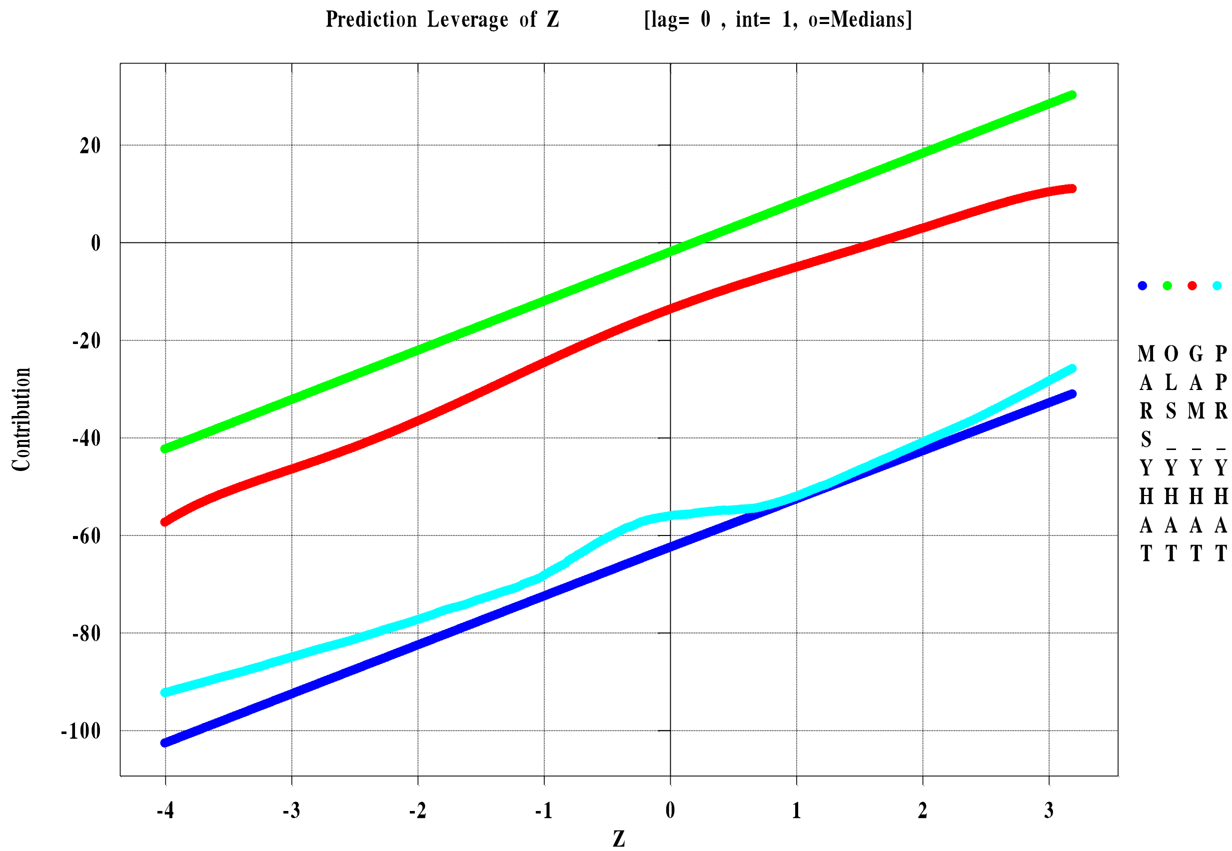



Figure 17.13 Leverage Plot of linear term for medians of data

The nonlinear techniques MARS, GAM and PPREG were clearly able to detect the nonlinearity for the *x* variable while finding the linear term *z*. Exploritory projection pursuit results are not shown due to space limits but should be consulted.

The file ch17.mac, distributed with `B34S`, contains a large number of further examples to illustrate the methods discussed in this chapter. The file stattest2.mac shows using **ppreg**, **marspline**, and **gamfit** on a number of well known nonlinear models.

## 17.8 Conclusions

Exploratory projection pursuit was shown to be able to both detect the complexity of nonlinearity in the data and delineate where it is occurring in terms of the observations of a dataset. Using time series data, such analysis should be useful in modeling a dataset that may have a changing structure. For supervised learning models, projection pursuit and random forest models were shown to be useful in both categorical and continuous variable situations. For unsupervised models, where there is no left-hand-side variable to validate or score the result, the clustering approach was shown to have use.