FKR: Fast Kernel Regression
John R. Wolberg

Introduction:

The FKR program is used to model data based upon the kernel regression method. Kernel regression (KR) is one of the most powerful methods for modeling data in which there is very little knowledge regarding the basic underlying principles. KR is applicable to problems in which there are a large number of candidate predictors and a large number of data records. This memo includes a description of the KR method and details regarding usage of the program for KR analyses. A complete discussion of the method is included in my book: *Expert Trading Systems: Modeling Financial Markets with Kernel Regression*, John Wiley & Sons, 2000.

Document Version and Date:	FKR Version 3.39	Dec 10, 2015
	Document update:	Dec 10, 2015

History:

Version 3.01, Aug 1, 1998:	Multiple best models using **BEST_MODELS** parameter
Version 3.02, Aug 4, 1998:	Cross sectional analysis using **DATECOL** & **TIMECOL**
Version 3.03, Aug 13, 1998:	Time weighting using **HALF_LIFE** parameter
Version 3.04, Aug 18, 1998:	Evaluation data set: **NEVL**, **INCLUDE_TEST, INLUDE_ALL**
Version 3.05, Aug 30, 1998:	Rolling forward Evaluation data set: **NFOLDS**
Version 3.06, Sep 2, 1998:	Terminating File **fkr_exit**
Version 3.07, Sep 3, 1998:	**MOD_CRITERION** parm for CorCoeff & CC_Origin
Version 3.08, Sep 17, 1998:	**NLRNDAYS**, **STARTLRNDATE**, etc.
Version 3.09, Nov 10, 1998:	Bug fix for cross-sectional feature
Version 3.10, Nov 24, 1998:	Alternative **DATES** for **DAYS** & **GAP** & **GAPDATES**
Version 3.11, Jan 11, 1999:	**MOD_CRITERION**=4 (separation)
Version 3.12, May 4, 1999:	**DMAX** <= **TREEHEIGHT**
Version 3.13, June 9, 1999:	Ascii data files may be space, comma, tab or new-line delimited
Version 3.14, Aug 24, 1999:	**NUMNN**=0 for **NUMCELL**>1 (all points in cells) & **GROUP_PCNT_TOP** & **GROUP_PCNT_BOTTOM**
Version 3.15, Mar 6, 2000:	**IDCOL** (refers to datafile). Contents included on **ycf** file
Version 3.16, May 22, 2000:	**DELTA_DIM** parameter
Version 3.17, June 18, 2000:	**NUM_MODELS** parameter
Version 3.18, June 20, 2000:	**YCFBIN** parameter
Version 3.19, June 28, 2000:	**YYMM** format for time weighting
Version 3.20, July 19, 2000:	Check for missing values: **CHECK_NODATA, NLRN_MIN, NLRN_MIN_PCT**
Version 3.21, Oct 31, 2000:	**NUM_MODELS** bug fix
Version 3.22, Feb 17, 2001:	**YACTCOL**
Version 3.23, Feb 25, 2001:	**NUM_OUTCOLS, OUTCOL(i)**
Version 3.24, Jan 17, 2002	Bug fix, Order=2
Version 3.25, Dec 18, 2002	Bug fix, SORT_BOX_ORDER
Version 3.26, Apr 6, 2003	**LINUX** compatibility (professional version only)
Version 3.27 May 31, 2005	dmax <= treeheight message
Version 3.28 Oct 21, 2008	TIME_BOMB problem & INC_ALL bug
Version 3.29 Jan 9, 2009	BEST_MODELS bug when MK > 1
Version 3.30 Oct 8, 2009	**GROUP_PCNT_xxx** = 0
Version 3.31 Jun 10, 2010	eps = 1e-13 in Crout reduction
Version 3.32 Feb 26, 2014	**MOD_CRITERION** = 5, **YSCALE**, comment character !
Version 3.33 Mar 17, 2014	**MOD_CRITERION** = 6 & 7, **SIGY** and **SIGYCOL**
Version 3.37 Oct 8, 2015	-999 bug in **F_calc**
Version 3.38 Oct 22, 2015	**PRINT_CELLS** option
Version 3.39 Dec 10, 2015	F-analysis with nevl > 0

Syntax: FKR datafile [parmsfile]

Input files:	*Dataname.Datatype*		(default *Datatype* is **pri**)
		or	
	Dataname.**lrn**	and	*Dataname*.**tst**
	Parmsname.**fpa**		(default *Parmsname* is *Dataname*)
Output files:	*Parmsname*.**fkr**		
	Parmsname.**ycf**		(if **BEST_MODEL** is specified)
	Parmsname.**cells**		(if **PRINT_CELLS=1** and **Print=1**)
	fkr_exit		

The data file (or data files) may be ascii or PRISM-type (binary) files. The **fpa** file contains the parameter values for the **FKR** run. The output **fkr** file is a report. The optional output **ycf** file is an ascii file which includes computed values of *y* and optionally computed values of *sigma_y*. The first two columns of the **ycf** file are the dates and the actual values of *y*. If **YCFBIN**='y' then the **ycf** file is created as a binary file (without the model details included in the file). The **cells** file contains the dimensions of all of the cells included in the learning tree. It also contains the average value and standard deviation of the *y* values included in each of the leaf cells. The file **fkr_exit** is created when the program terminates. The file includes the exit code which is zero if the program terminates successfully. Otherwise, the code is greater than zero.

Overview:

The program requires a learning data set and usually requires a test data set. These can be provided as separate data files or contained in a single file. A further option is to define a third data set: the evaluation set. This data set may also be included in the single data file or can be included at the end of the test data file. The program uses a full binary tree representation of the learning data. The leaves of the tree are equally (or approximately equally) populated. Each learning point is placed within one of the leaf cells of the tree. Three alternative methods of kernel regression are available: order zero, one or two. The order refers to the degree of polynomial used to fit the data. For each method, an attempt is made to find the leaf in which each test point resides. (It is possible for a test point to be outside of the limits of all leaves. Such points are disregarded.)

Learning points are used for estimating the value of a dependent variable Y for each test point. The zeroth order method (**FIT_ORDER**=0) is based upon a zeroth order polynomial which is just a constant. Thus this method computes the weighted average value of learning point Y’s as the predicted value of Y for a test point. However, if **FAST**=1 (or **FAST0**=1), then all points in the cell in which the test point resides are equally weighted and the predicted value of Y is just the cell average. Typically (unless **FAST** is specified) the program uses a nearest neighbor approach to kernel regression modeling. For each test point, once its cell location is determined, a search is made in **NUMCELL** cells for the **NUMNN** nearest neighbors. If **NUMCELL**=1, then only the cell in which the test point resides is included in the search. If **NUMCELL**>1, then the nearest **NUMCELL** - 1 cells are also included in the search. Regardless of the value of **NUMCELL**, the search for the **NUMNN** nearest neighbors is limited to only cells adjacent to the cell in which the test point resides. Thus if **NUMCELL** is set to a high number, the search is limited to the test cell and all adjacent cells.

Once the **NUMNN** points are located, these points are used to make a prediction. The coordinates of the test point are introduced and a weight is determined for each of the **NUMNN** learning points. The weights are based upon an exponential kernel with an exponent based upon the Cartesian squared distance from the test point to the learning point. If the parameter **FIT_ORDER** is specified as zero, then the weights are used to compute a weighted average:

$$\boldsymbol{Ycalc}(\boldsymbol{i}) = \frac{\sum_{\boldsymbol{i}=1}^{\boldsymbol{NUMNN}} \boldsymbol{w}(\boldsymbol{i}) * \boldsymbol{Y}(\boldsymbol{i})}{\sum_{\boldsymbol{i}=1}^{\boldsymbol{NUMNN}} \boldsymbol{w}(\boldsymbol{i})}$$

The weights *w(i)* are computed as follows:

$$\boldsymbol{w}(\boldsymbol{i}) = \exp(-\boldsymbol{f}(\boldsymbol{k}) * \boldsymbol{dist}(\boldsymbol{i}))$$

where *dist(i)* is the Cartesian squared distance from the test point to the i-th nearest neighbor. The points are pre-sorted so that the 1st point is the closest and the NUMNN-th point is the furthest. The Cartesian squared distance is computed as follows:

$$\boldsymbol{dist}(\boldsymbol{i}) = \sum_{\boldsymbol{j}=1}^{\boldsymbol{d}} (\dim(\boldsymbol{j}) / \boldsymbol{range}(\boldsymbol{j}))^2$$

where dim*(j)* is the distance of the i-th learning point to the test point in the j-th dimension and *range(j)* is the range of the j-th dimension for all points in the learning data set. The constant *d* is the dimensionality of the space. Note that the normalization by the *range(j)* frees the user from worrying about the magnitude of the various X's. The function *f(K)* is computed as follows:

$$\boldsymbol{f}(\boldsymbol{k}) = \ln(\boldsymbol{k}) / \boldsymbol{dist}(\boldsymbol{NUMNN})$$

In other words, the value of *w(NUMNN)* (the weight of the furthest point from the test point is *exp(-ln(k))* = 1 / *K*. Note that values of *K* < 1 cause weight to increase with distance which is meaningless and are therefore treated as errors. If K=1, *w(i)* is set to 1 (i.e., all points are equally weighted). If, for example, K=2, then the weight of the furthest point from the test point is 0.5. Setting the parameter **FAST**=1 (or **FAST0**=1) automatically sets **FIT_ORDER**=0, **NUMK**=1, **K(1)=1, NUMCELL**=1 and **NUMNN**=0 (i.e., all points in the test cell for each test point). If **FAST** has not been set and **NUMNN**=0 and **NUMCELL**>1 then all points in the **NUMCELL** cells are used. This takes more time than the **FAST** option but is much faster than if a full nearest neighbor search is initiated (i.e., if **NUMNN** is specified).

For **FIT_ORDER**=1, the **NUMNN** points are used to determine the coefficients of a hyperplane of the form:

$$\boldsymbol{Y} = \boldsymbol{a}(1) * \boldsymbol{x}(1) + \boldsymbol{a}(2) * \boldsymbol{x}(2) + ... + \boldsymbol{a}(\boldsymbol{d}) * \boldsymbol{x}(\boldsymbol{d}) + \boldsymbol{a}(\boldsymbol{d}+1)$$

where *d* is the dimensionality of the surface. The a's are determined using a weighted least squares calculation using the weighting function described above. A minimum of *d*+2 points are required to determine the *d+1* unknown values of *a*(k). Once the a's have been determined for a given test point, the value of *Ycalc* is determined using the coordinates of the test point and the equation for the hyperplane. It should be emphasized that typically a set of a's must be determined for every test point. The only exception to this rule is if **NUMCELL**=1, **NUMNN**=0 (i.e., use only the points in the test cell) and **K**=1. For this special case, one hyperplane is used for each cell. If several test points fall within the same cell, then this single hyperplane can be used for each of the points. This special case results in a very fast calculation and can be specified by setting the parameter **FAST1** to 1. Note that if **NUMCELL**>1 and **NUMNN**=0 the coefficients must always be recalculated.

For **FIT_ORDER**=2 the surface is a full multinomial of order 2 (including all cross product terms). This method is theoretically the most accurate and is also the most time consuming. However, for very noisy problems the order 2 method yields results that are often worse than the lower order methods. The number of coefficients that are needed to describe the surface is:

$$\boldsymbol{NumberCoeff} = 1 + \boldsymbol{d} + \boldsymbol{d} * (\boldsymbol{d}+1) / 2$$

where *d* is the dimensionality of the space under consideration. The program checks that the value of **NUMNN** is greater than *NumberCoeff* for the largest value of dim (i.e., **DMAX**). For example if *d*=3, the number of coefficients is 10 and therefore **NUMNN** must be at least 11. Similarly to **FAST1**, there is a parameter **FAST2**. If **FAST2**=1 then a single multinomial of order 2 is computed and saved for each leaf cell in which a test point resides. As the ratio of test points to leaf cells becomes large, the resulting calculation is vastly quicker than if a separate multinomial is computed for each test point.

The program includes 7different modeling criteria. Regardless of **FIT_ORDER**, once *Ycalc* (the predicted value of *Ytest*) for each test point) has been determined, the value of **MOD_CRITERION** is used to select which of the 7 is used. If the value is 1, then Variance reduction (VR) is used to measure the predictive power of a given space. Each value of *Ycalc* is compared to the actual value of *Ytest* for each test point. The VR for the space is computed as follows:

$$\boldsymbol{VR} = 100 * \left(1 - \frac{\sum_{\boldsymbol{i}=1}^{\boldsymbol{NTST}} (\boldsymbol{Ycalc}(\boldsymbol{i}) - \boldsymbol{Ytest}(\boldsymbol{i}))^2}{\sum_{\boldsymbol{i}=1}^{\boldsymbol{NTST}} (\boldsymbol{Ytest}(\boldsymbol{i}) - \boldsymbol{Yavg})^2}\right)$$

where *Yavg* is the average value of *Ytest* for all test points included in the analysis. The summations are made over all **NTST** Test Points. Note that for a perfect model in which all values of *Ycalc* are exactly equal to *Ytest*, the value of VR is 100.

A value of **MOD_CRITERION =** 2 invokes the Correlation Coefficient (CC) modeling criterion. A value of 3 invokes the Correlation Coefficient through the Origin (CCO) criterion. Both of these criteria are measures of the deviation from a straight line formed using the values of *Y* and *Ycalc.* The line used to determine CC is the least squares line created using the data. The line used to determine CCO is the least squares line which passes through the origin of the *Y*, *Ycalc* plane. Both values are expressed as percentages and therefore may vary from 100 to –100.

A value of **MOD_CRITERION =** 4 invokes the Separation (SEP) criterion. The SEP criterion uses a parameter **GROUP_PCNT**. The values of *Ycalc* are sorted and then two groups of the corresponding values of *Y* are analyzed: G1 is the bottom group with the lowest values of *Ycalc* and G2 is the top group with the highest values. The numbers *n1* and *n2* in each group is **NTST** * **GROUP_PCNT /** 100. Alternatively the values of *n1* and *n2* can be specified separately by using the parameters **GROUP_PCNT_TOP** and **GROUP_PCNT_BOTTOM**. The average values and σ's of each group are computed and separation is then computed as:

$$SEP = (AVG(G2) - AVG(G1)) / \sqrt{\sigma(G1)^2 / n1 + \sigma(G2)^2 / n2}$$

If the values of *Y* and *Ycalc* are unrelated (i.e., the model is useless) we expect SEP to be distributed normally with a mean of zero and a standard deviation of 1. In other words, SEP measures the separation between the means of the two groups in units of standard deviations. The values of FracSS included in the output report for this criterion only refer to the records in the two groups.

A value of **MOD_CRITERION =** 5 invokes the Weighted Sum-of-Squares (WSS) criterion. The WSS criterion minimizes the weighted sum of the squares of the differences between the actual and calculated values of *Y*. The values of WSS are the weighted sum of the squares divided by the number of test or evaluation points. This is useful when the *Y* data is based upon numbers of counts of an observable quantity. As examples, consider demographic data or radiation dosage data. Typically such data is Poisson Statistics distributed and the standard deviation of the values of *Y* can be assumed to be $Y^{1/2}$. By weighting the data points as $\sigma_y{}^2$ the resulting differences between the actual values of *Y* and the calculated values (i.e., the residuals) are approximately proportional to σ_y. For such data this is a more sensible criterion than trying to make the residuals approximately constant. Sometimes it is more convenient to use scaled values of the *Y* data. For example, if *Y* represents thousands of people, then the value of **YSCALE** would be set equal to 1000. The default value of **YSCALE** is one. If the model is good, then one can expect a value of **WSS** of approximately one. If scaled data is used, the resulting value of **WSS** should still be approximately one if **YSCALE** has been set properly.

A value of **MOD_CRITERION =** 6 invokes the Constant Fractional Error (**CFE**) criterion. When it can be assumed that the fractional errors in the values of Y are approximately constant, this modeling criterion is applicable. The values of CFE are the weighted sum of the squares divided by the number of test or evaluation points.. An example of experiments in this class are those in which the Y variable represents something determined by a measuring instrument that exhibits constant fractional error. This modeling criterion is similar to WSS with the only difference being the calculation of the weights for each value of Y. For experiments in this class the values of σ_y are assumed to be SIGY*Y and the weights are thus $1/(SIGY*Y)^2$. The default value of SIGY (an optional input parameter) is 1. However, if a good estimate of SIGY is available, then the value of CFE is a measure of the quality of the model. If one has achieved a good model, then this ratio should be close to one. For problems in which it can be assumed that the Y values exhibit constant fractional error but the modeler has no idea what the best estimate of SIGY should be, the value of CFE can be used to estimate SIGY. For example, assume that this ratio is about 0.01 for the best model obtained using FKR using the default value of SIGY = 1. To increase this value to about 1, SIGY would have to be reduced to a value of 0.1 (i.e., 10% uncertainty in the values of Y). A counting experiment might fall within this category if the values of Y are count rates and a constant number of counts is required to measure each value of Y. For example, to achieve a value of SIGY = 0.01 (i.e., 1% accuracy in the values of Y, one would require 10000 counts for each data point. To do this one would run the counter for each point for the amount of time required to reach 10000 points. Since the value of σ_y for each point is $Y^{1/2}$ = 100, a value of SIGY = 100/10000 = 0.01 should be specified.

A value of **MOD_CRITERION =** 7 invokes the Sigma Y's as input (**SIG**) criterion. The values of SIG are the weighted sum of the squares divided by the number of test or evaluation points. For some problems, the value of σ_y might vary from point to point in a way known by the experimenter. For example, different types of measuring instruments with differing inherent measuring errors might be used for the data points. For some measuring instruments the measuring error might vary depending upon the value of the quantity being measured. For count rate experiments with a wide range of differing count rates, achieving a constant fixed number of counts might not be practical. To cover problems falling within this SIG class, MOD_CRITERION=7 can be used. An additional data column (SIGYCOL) is specified and the values of σ_y are taken directly from the input data file. Once again, if the specified values of σ_y are reasonable estimates of the uncertainties in the values of Y, then a value of SIG close to one will be achieved for a good model.

The modeling process looks at various spaces from dim = **DMIN** to **DMAX**. A variety of potential independent variables can be tested and each combination comprises a space. For example, if we have 5 variables (i.e., X(1), X(2), .. X(5)), and if we want to consider all spaces from dim = 1 to dim = 3, then we would have to look at each X individually, then all pairs (i.e., (X(1),X(2)), (X(1),X(3)), ... up to (X(4),X(5)), and finally all triplets (i.e., (X(1),X(2),X(3)), ... (X(3),X(4),X(5)). If the total number of X columns is large, then the potential number of spaces to be examined becomes immense. The program includes a feature which allows specification of **SURVIVENUM**(i) for i = **DMIN** to **DMAX**. Only the best spaces survive to the next dimension. This strategy reduces the number of spaces to be examined. By selecting the values of **SURVIVENUM**(i), the total number of spaces examined can be controlled. Another parameter useful in limiting the number of spaces to be examined is **DELTA**. A more detailed explanation of the use of **DELTA** is included below in the section on the Survivor Concept.

Regardless of the choice of **FIT_ORDER**, each space examined must be pre-processed so that all the learning points are distributed in the full binary tree of specified height. The number of points per leaf is very close to being constant. The number of leaves and the average number of points per leaf are controlled by two parameters: **TREEHEIGHT** and **NLRN**:

$$\boldsymbol{AvgPerLeaf} = \frac{\boldsymbol{NLRN}}{2^{\boldsymbol{TREEHEIGHT}}}$$

For example, if the number of learning point **NLRN** = 20480, and tree height is specified as 8, then the points per leaf is 20480 / 256 = 80. An alternative parameter is **BUCKETSIZE**. If **BUCKETSIZE** is specified, the **TREEHEIGHT** is computed so that the average number of point per leaf is at least **BUCKETSIZE**.

Once the tree has been generated, if **NUMCELL**>1 an adjacency matrix is created which includes a list for all leaves of the tree. Each list contains the index of all adjacent leaves to the given leaf. The leaf in which each test point is located is determined by working down thru the tree. The search starts at the root cell and goes down level by level until it falls into a leaf (or is rejected because it is outside the range of all learning points). The number of cells examined is just the tree height. For each test point the **NUMNN** nearest neighbors (in the learning data set) are determined by measuring the distance of all points in the leaf plus all points in the (**NUMCELL**-1) closest adjacent leaves. If **NUMCELL** is greater than one, then for each test point, the distance to the center of each adjacent leaf is measured and the closest leaves are chosen for the search. If **NUMCELL** is set to an artificially high value then all adjacent leaves will be searched for all test points. Regardless of the value of **NUMCELL** we are not insured that the true **NUMNN** nearest neighbors will be discovered. However, for kernel regression based upon a number of test points **NTST** >> 1, extensive testing of the program has shown that this has a very small effect on the computed values of *Ycalc*. In other words, if a few of the nearest points to some test points happen to lie in non-adjacent cells, the average differences caused by disregarding these points in not significant.

By examining all specified spaces for a given level of dimensionality saving the survivors and then proceeding up to the next level of dimensionality, the program attempts to determine if a model exists which can be used for predicting values of Y. Clearly, if no model exists, then any modeling method will fail. If, however, a model does exist, this program is a very useful tool for determining a model even if the number of variables and the number of data points are large.

The program allows three possible options for selecting the learning data set: **LTYPE** = '**S**' (static), '**M**' (moving) or '**G**' (growing). The default option is a static learning set. For this option once the **NLRN** data points have been read, then these and only these points are used for all the test points. If the moving option is selected, then the original data set is only used for the first test point. For each subsequent point, one new data point is added to the set and the oldest data point is discarded. The growing option is similar to the moving option except that no points are discarded. If **LTYPE**='**M**' or '**G**', then the parameter **GROWTH_FACTOR** must be greater than 1. This is necessary for the 'M' option even though the total number of points does not change. The sum of the changes in the number of points per leaf is 0, but some leaves will grow while others exhibit a decrease in number of points. If the growth factor isn't large enough to receive all added points for a particular leaf, a warning message is issued but processing continues (without adding the new point to the leaf). The warning messages can be suppressed by setting the parameter **WARNING**=0.

The **BEST_MODEL** mode of operation is used to create an ascii file (*parmsname*.**ycf**). The value of **BEST_MODEL** specifies the number of models to be included in the file. The file includes the *y_calc* values of each of the best models. If **SIGCOL**=1 it also includes the values of *sigma_y_calc* (i.e., the computed standard deviation of the value of *y_calc*) for each of the best models. The initial columns are the dates (and times if **TIMECOL** is specified) and the actual values of Y. If **IDCOL** (identifier column) is specified, the values of this column in the datafile are then included in the next column of the **ycf** file. The number of rows of data is **NEVL** unless **NEVL** is not specified. If it is not specified then the number of rows is **NTST**. If **NEVL** is specified and **INCLUDE_TEST** is specified then the number of rows is **NTST** + **NEVL**. If **INCLUDE_ALL** is specified then the number of rows is **NLRN + NTST + NEVL**. If any of the values are indeterminate, then the **NODATA** value (default –999) is used.

Once the best models have been determined, they can be using to determine predicted values of Y for an additional data set (the evaluation set) which was not used in the modeling process. Comparing the predicted values of Y and the actual values of Y for this data set provide an independent out-of-sample evaluation of the models. The parameter used to indicate the size of the evaluation set is **NEVL**. If the parameter **NFOLDS** > 1, then all three data sets are rolled forward **NEVL** records for each fold and the analysis is repeated. For such cases, after the first fold **INCLUDE_ALL** or **INCLUDE_TEST** are set to zero so that the final **ycf** file will only have a maximum of **NLRN + NTST + NFOLDS*NEVL** records. If **INCLUDE_ALL** and **INCLUDE_TEST** are specified as zero then the maximum number of records in the **ycf** file is **NFOLDS*NEVL**. If **NFOLDS** is specified so that the total number of records exceeds the number of records in the file, then the value is reduced. Furthermore, if there are not enough records to have **NEVL** records in the final fold, the number of evaluation data points in this final fold is reduced appropriately.

Program Output (*.fkr* and *.ycf* files):

The program output (on the ***.fkr*** file) includes survivors from all dimensions considered. For each surviving space several output columns are included:

- **Measure of Performance:** The measure of performance depends upon the value of the MOD_CRITERION (1 to 7: see above).

- **FracSS:** The fraction of the test point for which the sign of the calculated value of Y is the same as the actual value of Y. If MOD_CRIERION=5, then FracSS is not included in the output because all values of Y are greater or equal to zero.

- **F:** The F statistic is a measure of how far the space differs from the null hypothesis (i.e., there is no significant difference in the distribution of the Y values from cell to cell). If the null hypothesis is true then one would expect an F value near 1. If F significantly exceeds one, then it can be concluded that the space contains some meaningful information. The 2σ limit for F is included in the program output.

The output also includes the best models for each fold of data.

The program output (on the ***.ycf*** file) includes a record for each evaluation data record. The columns included are:

1) The date column.
2) The IDCOL (if specified)
3) The YACTCOL (if specified)
4) OUTCOL(i) , i = 1 to NUM_OUTCOLS (if NUM_OUTCOLS specified)
5) The YCOL
6) Y-Model i , i = 1 to BEST_MODELS (if SIGCOL=0) or
7) Y-Model i, Sigma Model i , i = 1 to BEST_MODELS (if SIGCOL=1)

If YCFBIN=’Y’ then this file is created in binary format. Otherwise the file is created as an ascii file. If the file is ascii, then the model dimensions and columns are included as a separate table for each fold. For the ascii file only the date column and the IDCOL are in integer format. All other columns are in %15.6e format. If the binary file is created, then the model details can be extracted from the ***.fkr*** file.

F Statistic Analysis:

A very fast method for examining the spaces is to use the F Statistic of the Learning Spaces as a Measure of Performance instead of Variance Reduction. To activate this method set **NTST** = 0. There is no need for test points as the F Statistic is based upon only the learning points. The computed 2 Sigma limit for the F statistic is included in the output report and may be used to determine the significance of the listed values of F. It should be emphasized that a non-significant value does not necessarily imply that a space is useless. If for example a small portion of the space is highly predictive, then the F statistic may be close to one but the space might still exhibit useful predictive power. The separation criterion (MOD_CRITERION=4) may be useful for such spaces.

Defining the Analysis:

The parameters required to define the analysis are included on the **fpa** file. For example, consider the file **memo.fpa**:

```
 ! This is a comment.  The comment is terminated by a new line or end of file
NLRN=2048          NTST=1000      NEVL=500      NFOLDS=3
YCOL=19            FAST1=1        DELTA=1        TREEHEIGHT=6
NVAR=10            DMIN=1         DMAX=5          LTYPE='G'
GROWTH_FACTOR=3                   SURVIVE_NUM=3
SURVIVENUM(1)=10                  SURVIVENUM(2)=5
```

The interpretation of these parameters is as follows: the analysis includes the first 2048 records of the data file as the initial **NLRN** Learning data points, the next 1000 records as **NTST** Test points and the next 500 records are the **NEVL** Evaluation points. (By default the Test data points immediately follow the Learning points and the Evaluation points start immediately after the Test points.) The parameter **NFOLDS**=3 causes the analysis to be repeated 3 times and for each fold, all three data sets are rolled forward **NEVL** (i.e., 500) records. All the learning and test points are combined into a new tree (of height **TREEHEIGHT**) for the **BEST_MODEL** (or **BEST_MODELS**) the **MOD_CRITERION** (default is Variance Reduction) is computed for the model (or models). Regardless of the value of **LTYPE**, all learning and test points are used to make estimates for the values of ***y*** in the Evaluation data set.

The growing option is to be used (**LTYPE**='G'), and the **GROWTH_FACTOR** is set as 3. This factor is the maximum ratio of number of learning points a cell can grow. Any attempts to add additional points are ignored. Since **FAST1** is specified, only the test cells are used to determine calculated values of *y*. **FAST1** automatically sets **FIT_ORDER**=1, **NUMCELL=**1**, NUMK**=1 and **K(1)=**1.

The number of data columns per record is taken from the file header but must be at least 19 since the dependent Y variable is included in **YCOL**=19. If the data file is an ascii file, then the parameter **NCOL** must be specified. Specification of **NVAR**=10 means that 10 columns of data will be used as independent variables in an attempt to discover a model of dimensionality **DMIN**=1 to **DMAX**=5. Since the parameter **STARTVAR** wasn't specified and the individual **X(**i**),** i=1 to 10 weren't specified, the program defaults to columns 1 to 10.

The parameter **SURVIVENUM(**dim**),** dim = **DMIN** to **DMAX** is specified as 10 for dim=1, 5 for dim=2 but defaults to 3 for all other dimensions. After all analyses for a given dimension have been completed, only the best **SURVIVENUM**(dim) spaces are used to create spaces of the next higher dimension.

The **DELTA** parameter requires spaces to exhibit an increase of at least 1% in Variance Reduction to be considered as a candidate for survival. The improvement alone does not insure survival. The space must also rank within the top **SURVIVENUM(**dim**).** A more detailed explanation of the use of **DELTA** is included below in the section on the Survivor Concept.

Forced Variables:

A useful feature in modeling programs is the ability to force some variables into the analysis. Several parameters are available for this feature of the program. The parameters **NF** and **DIMF** are the number of forced variables and the dimension of the forced variables. For example, assume that **DIMF**=1 and **NF**=3 and we wish to force variables 1, 3 and 7 into all combinations of the higher dimensions. To specify this we would also set **XF(1)**=1 **XF(2)=**3 and **XF(3)=**7. Lets assume that we have a total of 8 variables (**NVAR**=8) and furthermore they are in columns 1 thru 8. Thus the pairs considered would be only pairs which contain at least one of the three specified forced variables. If we force at a dimensionality greater than 1 (e.g., **DIMF**=2), then we must specify a different set of parameters. Lets say we wish to force the following three pairs into the analysis: 1 & 3, 3 & 4, and 2 & 7. To specify this we would again use **NF**=3. The three pairs are specified as follows: **XFA(1,1)=**1 **XFA(1,2)=**3 **XFA(2,1)=**3 **XFA(2,2)=**4 **XFA(3,1)=**2 and **XFA(3,2)=**7. These pairs would then be forced into the 3 dimensional analysis.

The Survivor Concept:

The **FKR** program examines spaces starting from **DMIN** dimensions and continuing to **DMAX**. The parameter **SURVIVENUM**(dim) is the maximum number of spaces of dimensionality *dim* that are saved and used as the basis for dimensionality *dim*+1. The spaces saved at any value of *dim* are those with the largest values of the modeling criterion MC (e.g., VR, SEP, etc.). If the parameter **DELTA** is specified, then a space of size *dim*+1 is only saved if it exhibits an improvement of **DELTA** as compared to the space of size *dim* upon which it is based. For example, consider the space X5,X17 as a survivor at 2 dimensions with a value of MC = 5.3. Assume **DELTA** has been specified as 1. Consider two 3D spaces: X5,X12,X17 with MC = 6.7 and X5,X17,X23 with MC = 5.92. The first of these spaces is a candidate for survival if MC = 6.7 is enough to keep it in the maximum permissible survivors for 3 dimensions (i.e., **SURVIVENUM**(3)). The second space is immediately rejected even if 5.92 is enough to place it within the top **SURVIVNUM**(3) spaces. Clearly, for *dim* = **DMIN**, the **DELTA** parameter is meaningless. **DELTA** may be negative, zero or positive and is only applicable if *dim* > **DMIN**. For *dim* = **DMIN**, all possible spaces are considered. Thus if DMIN=1 and NVAR=50, all 50 1D spaces are examined. If **DMIN** is 2, then all 50*49/2=1225 2D spaces are examined. It is therefore clear that if **NVAR** is large, it is not a good idea to set **DMIN** too large. For example, if **NVAR**=100 and **DMIN**=4, then almost 4 million 4D spaces would have to be examined!

Using the **DELTA** parameter, it is possible that at a given value of *dim* no spaces survive. For such cases, the program ceases operation. For example, consider the case where **DMIN**=1, **DMAX**=5, **NVAR**=50 and **SURVIVE_NUM** is set as 10. Assume that **DELTA** is specified as 1. All 50 1D spaces are examined and the best 10 survive to the next level (i.e., 2 dimensions). The number of spaces examined at this level is thus 49 + 48 + ...+ 40 which is 445. Even though **SURVIVENUM(2)** is 10, assume only 5 of these 445 pairs exhibits an improvement of **DELTA** over the 1D space upon which it is based. The number of 3D spaces to be examined is thus less than or equal to 48 * 5 = 240. (It is exactly 48*5 only if all five spaces are made from completely different variables.) If none of these spaces exhibit an improvement of at least **DELTA** then no 4D or 5D spaces are examined even though **DMAX**=5.

There is a second parameter similar to **DELTA**: the **DELTA_DIM** parameter. When saving **BEST_MODELS**, a space must show at least an improvement of **DELTA_DIM** over the best space at all lower dimensions. In other words, **DELTA_DIM** acts as a penalty function for higher dimensional spaces. If for example, **DELTA_DIM**=1 and the best 1D space exhibits a value of MC=3.5. Regardless of the number of survivors at the 2D level, only those with MC>=4.5 will be included in the best model list. Lets assume that at least one exceeds this limit and has a value of MC=4.65. Then only 3D survivors with MC>=5.65 will be included in the best model list.

Cross Sectional Analysis:

When there are multiple records for a given date (or date/time), then the growing and moving options must be modified. Records cannot be added to the learning data set immediately upon usage. Only when all records for the given time slice have been processed, they can then be added to the learning data set. The FKR program includes two methods for specifying multiple records:

1) Specifying a date column. (**DATECOL**)
2) Specifying a date column and time column. (**DATECOL** and **TIMECOL**)

Both methods can be used even if the number of multiple records is not constant. If only a date column is specified, then this column is used as a key column to test whether or not there are multiple records. If a time column is also specified (HHMM format), then a combined key based upon date and time is used. If the input data files (or file) are PRISM format files, then the date column is taken directly from the file header. The format for dates in the date column may be short (YYMM or YYMMDD) or long (YYYYMMDD). The year 2000 problem is handled as follows: if date>=20000: if YY<**EARLY_DATE**, YY = YY + 100. If date<20000 then Jan 2000 must be 10001.

One of the problems encountered with cross sectional data is that the number of records may change from day to day. This causes a problem when specifying the values of **NLRN**, **NTST**, and **NEVL**. The more natural way to treat such problem is to specify **NLRNDATES, NTSTDATES**, and **NEVLDATES.** To make the specification complete we also need to indicate starting dates. The following parameters are available: **STARTDATE, STARTEVLDATE, STARTLRNDATE, STARTTSTDATE** and **ENDDATE.** If only **STARTDATE** is specified, then **STARTLRNDATE** defaults to this date, **STARTTSTDATE** is the date immediately following the **NLRNDATES** and if **NEVL** is specified, then **STARTEVLDATE** immediately follows the **NTSTDATES.** If no value is specified for any of the starting dates, then **STARTDATE** defaults to the first date in the data file. If the value of **NFOLDS** is greater than one, then on subsequent folds, all dates are advanced by **NEVLDATES.** When using this method of specifying the sizes of the various data sets, the values of **NLRN, NTST** and **NEVL** are computed and may vary from fold to fold.

GAP & GAPDATES:

Sometimes it is necessary to maintain a gap between the learning and test data and between the test data and the evaluation data. If for example, we are trying to model the change over two days and our database consists of daily records, then the Y value for record *i-1* shouldn't be used for making a prediction of Y for record *i.* The record at *i-1* contains information from the "future" (i.e., from *i+1).* The parameters **GAP** or **GAPDATES** are available for creating gaps between the data sets. For operations in which **NLRN**, **NTST**, and **NEVL** are specified, **GAP** is assumed to be a number of records included in the **GAP.** If **NLRNDATES, NTSTDATES** and **NEVLDATES** are specified, then **GAP** or **GAPDATES** are assumed to be a number of dates included in the gap.

Time Weighting:

Time weighting of data allows the analyst to reduce the weight progressively for earlier data records. The weighting function described above in the Overview Section is modified:

$$w(i) = \exp(-(f(k) * dist(i) + \alpha t))$$

The time parameter t is the time difference measured in days from the i-th learning point to test point under analysis. The decay constant α is determined from the **HALF_LIFE** parameter from the following equation:

$$0.5 = \exp(-\alpha * \text{ HALF\_LIFE})$$

As an example, assume that analyst wants data a year old to be given half the weight of current points at the same distance from a test point. The value of α would be computed as:

α = -ln(0.5)/365 = 0.693/365 = 0.0019

The time differences *t* are determined from the values in the **DATECOL** of the data file. The format for dates in the date column may be short (YYMM or YYMMDD) or long (YYYYMMDD). The year 2000 problem is handled as follows: if date>=20000: if YY<**EARLY_DATE**, YY = YY + 100. If date<20000 then Jan 2000 must be 10001. Time differences are taken by first converting dates to Julian dates.

NUM_MODELS:

If **NUM_MODELS** is specified, the program read the parameters associated with the models and then considers only these spaces. This feature is only applicable if **NFOLDS**=1. If **BEST_MODELS** is specified then a *.ycf* file is created. **BEST_MODELS** may be less than **NUM_MODELS** but it cant be greater. The models are specified using the arrays **MX** and **MK.** For example, assume NUM_MODELS=2, the first model includes variables 1, 3 and 7 with K=8 and the 2nd includes variables 2 and 10 with K=4. The specification would be as follows:

```
NUM_MODELS=2   MX(1,1)=1  MX(1,2)=3  MX(1,3)=7  MK(1)=8
               MX(2,1)=2  MX(2,2)=10             MK(2)=4
```

If MK(*i*) isn't specified, then the default value of 1 is used. As an example of the usage of this feature, assume that the following parameters were used to create the models: **NLRNDATES** = 24, **NTSTDATES** = 12 and **BEST_MODELS** = 15. After one month these models are to be examined and the best 5 are to be selected for usage in the following month. The correct choice of parameters is **NUM_MODELS** = 15, **NLRNDATES** = 36, **NTSTDATES** = 1, **NEVLDATES** = 0 and of course the specification of the 15 models. The following month the values of **NLRNDATES** would be increased to 37. A *.ycf* file is created so the results of the models for the **NTSTDATES** are available.

If a variable specified in a model is not a recognized variable, the program will issue a message: MX(*n,m*) not in X(i), i=1 to NVAR. For example, if the value of **NVAR** = 9 and the **X**(i) values are not specified explicitly, then the program will issue the message MX(2,2) not in X(i), i=1 to NVAR. To correct this either specify **NVAR** = 10 or explicitly specify each of the model variables. For the above example, the specification could be: **NVAR** = 5 **X**(1)=1 **X**(2)=2 **X**(3)=3 **X**(4)=7 **X**(5)=10.

Missing Values:

If the parameter **CHECK_NODATA** is specified as ‘Y’, then the program tests all input values for the **NODATA** code (by default –999). For every space, the input variables for the learning data are first examined for this code and if any of the variables in a given record equals the **NODATA** code, the record is eliminated for the space. For any record in the test or evaluation data set, if there is a missing value in the input variables of a space, the output value for that record is the **NODATA** code. If the remaining number of learning data points is less than. **NLRN_MIN** then the space is not examined. An alternative method for specifying **NLRN_MIN** is **NLRN_MIN_PCT**.

List of Parameters:

Parameter	**Optional**	**Default**	**Limitation**
BEST_MODEL	Yes	1 if **NEVL**>0	If greater than 0, generate **ycf** file
BEST_MODELS	Yes	else 0	Synonym of BEST_MODEL
BUCKETSIZE	*		Specify this or TREEHEIGHT
CHECK_NODATA	Yes	N	If Y then invokes missing values check
CLIP	Yes	Not specified	Y clipped: -CLIP<=Y<=CLIP
DATECOL	Yes	Not specified	See Cross Sectional Analysis
DELTA	Yes	Not specified	See: The Survivor Concept
DELTA_DIM	Yes	Not specified	See: The Survivor Concept
DMAX	Yes	DIM	1 to 10
DIM	Yes	1	1 to 10
DIMF	Yes	0	Must be < DMIN

DMIN	Yes	DIM	1 to 10
EARLY_DATE	Yes	0	If date < early_date, date += 1000000
ENDDATE	Yes	99999999	See Cross Sectional Analysis
GAP	Yes	0	See GAP & GAPDATES
GAPDATES	Yes	0	See GAP & GAPDATES
GROUP_PCNT	Yes	10	Used when MOD_CRITERION=4
GROUP_PCNT_TOP	Yes	GROUP_PCNT	G_P_TOP + G_P_BOTTOM <= 100
GROUP_PCNT_BOTTOM		GROUP_PCNT	G_P_TOP + G_P_BOTTOM <= 100
GROWTH_FACTOR	Yes	2	Must be > 1 if LTYPE is G or M
HALF_LIFE	Yes	Not specified	See Time Weighting
IDCOL	Yes	Not specified	Include values of this column in **ycf** file
INCLUDE_ALL	Yes	0	If 1 include all recs in **ycf** file
INCLUDE_TEST	Yes	0	If 1 include test and eval recs in **ycf** file
FAST	Yes	0	If 1, then fast mode. See **
FAST0	Yes	0	Alternative to FAST. See **
FAST1	Yes	0	If 1, then fast mode. See **
FAST2	Yes	0	If 1, then fast mode. See **
FIT_ORDER	Yes	1	0 to 2.
K(i)	Yes	1	If 1 all points same weight
LTYPE	Yes	S	G, M or S
MK(i)	Yes	Not specified	See NUM_MODELS discussion
MOD_CRITERION	Yes	1	1=VR, 2=CC, 3=CCO, 4=SEP, 5=WSS, 6=CFE, 7=SIG
MX(i,j)	Yes	Not specified	See NUM_MODELS discussion
NCP	Yes	DMAX	Synonym for NVAR
NEVL	Yes	0	NEVL<=NREC-NLRN-NTST
NEVLDATES	Yes	Not specified	See Cross Sectional Analysis
NF	Yes	0	NF < NVAR
NFOLDS	Yes	1	Limited by NREC
NLRN	Yes	NREC-NTST	At least 2 ^ TREEHEIGHT
NLRN_MIN	Yes	Not specified	See Missing Values
NLRN_MIN_PCT	Yes	Not specified	See Missing Values
NLRNDATES	Yes	Not specified	See Cross Sectional Analysis
NLRNDAYS	Yes	Not specified	Synonym for NLRNDATES
NODATA	Yes	-999	For Ycalc (column 3) of **ycf** file
NREC	Yes	from file	up to num records of file
NTST	No	Not specified	Either 0 or 1<NTST<=NREC-NLRN
NTSTDATES	Yes	Not specified	See Cross Sectional Analysis
NUMBOX	Yes	1	Synonym for NUMCELL
NUMCELL	Yes	1	Increase if NUMNN too large
NUMCELLS	Yes	1	Synonym for NUMCELL
NUM_MODELS	Yes	Not specified	See NUM_MODELS discussion
NUMNN	Yes	computed	Defaults to NLRN / TREEHEIGHT^2
NUM_OUTCOLS	Yes	Not specified	Include these columns in **ycf** file
NVAR	Yes	DMAX	Must be <= num columns of file
NUMK	Yes	1	Must be 1 to 20
OUTCOLS(i)	Yes	Not specified	ii = 1 to NUM_OUTCOLS
PRINT	Yes	0	0 to 5 (increasing levels of output)
PRINT_CELLS	Yes	0	1 creates **cells** file if **print**=1
PRINTY	Yes	0	If 1 then print Y, Ycalc table
SIGCOL	Yes	0	If 1 then include sigmas in **ycf** file
SIGYCOL	Yes	0	Col with Sig Y data (mod_criterion=7)
SIGY	Yes	1	Constant frac error (mod_criterion=6)
STARTDATE	Yes	Not specified	See Cross Sectional Analysis
STARTEVL	Yes	See below *	After test records
STARTEVLDATE	Yes	Not specified	See Cross Sectional Analysis
STARTLRN	Yes	1	STARTLRN+NLRN-1 <= NREC
STARTLRNDATE	Yes	Not specified	See Cross Sectional Analysis
STARTTST	Yes	See below *	STARTTST+NTST-1 <= NREC of test
STARTTSTDATE	Yes	Not specified	See Cross Sectional Analysis

STARTVAR	Yes	1	X(i) = column (i+STARTVAR-1)
SURVIVE_NUM	Yes	0	0 means all spaces survive
SURVIVENUM(i)	Yes	SURVIVE_NUM	See above for explanation
TIMECOL	Yes	Not specified	See Cross Sectional Analysis
TIMING	Yes	0	If 1, then output timing report
TREEHEIGHT	*	0	If not specified, computed
WARNING	Yes	1	If 0 suppress some warnings
X(i)	Yes	from STARTVAL	1 to num columns of datafile
XF(i)	Yes	Not specified	Must equal one of the X vals
XFA(i,j)	Yes	Not specified	i=1 to NF, j=1 to DIMF (DIMF>1)
YACTCOL	Yes	Not specified	Include values of this column in **ycf** file
YCFBIN	Yes	'N'	If 'Y' the ycf file is PRISM format
YCOL	No	Not specified	Must be a valid column
YSCALE	Yes	1	Used for WSS modeling criterion

* If there is a separate **.tst** file then the default value of **STARTTST** is 1, else it is **NREC** - **NTST – NEVL** + 1. If **NEVL** is specified then the default value of **STARTEVL** is **STARTTST + NTST**.

** If FAST is specified, **FIT_ORDER** is set to 0, **NUMK**=1, **K(1)**=1, **NUMCELL**=1. If **FAST1** is specified, then same as **FAST** but **FIT_ORDER**=1. If FAST2 is specified, then same as **FAST** but **FIT_ORDER**=2.

Demonstration of FKR:

This demonstration of FKR includes 3 files: the file **wssdemo.fpa**, **wssdemo.fkr** and **wssdemo.ycf.** The ycf file has been shortened to show only the first few records as this is a long file and includes a record for each of the 1000 data points in the evaluation data set. The wssdemo.fpa is the parameter file that defines the analysis:

```
!  This is a demostration of the new WSS feature of FKR
nlrn=7000  ntst=2000 nevl=1000 treeheight=8
ltype='s' best_models=1 fit_order=1  mod_criterion=5
dmin=1 dmax=2 nvar=5 numnn=10 numcell=100
ycol=10 NCOL=10
numk=2 k(1)=1 k(2)=5
yscale=10000  ! scale up y values by a factor of 10000
```

The data for this analysis is in **wssdemo.txt** and includes 10000 records that are used as follows: the first 7000 records **(`nlrn=7000`)** are the learning data set, the next 2000 (**`ntst=2000`**) are the test data set and the final 1000 **(`nevl=1000`)** are the evaluation data set. The first 5 columns are the X variables (i.e., the candidate predictors). The Y variable (i.e., the variable to be modeled) is located in column 10. All values of Y are to be scaled up (**`yscale=10000`**) by a factor of 10000. One and two dimensional combination of the 5 X variables are to be considered as possible models. The results shown in the **wssdemo.fkr** file show that an excellent 2D model is obtained with `X2 and X4.` Using all the data in both the learning and test data sets, the resulting value of **WSS** was 1.22 which is the weighted sum of the squares of the residuals divided by the number of evaluation data points. The analysis was based upon fitting a line to the 1D models and a plane to the 2D models (i.e., **fit_order=1**). The best 2D plane was determined by using 10 nearest neighbors with a value of **K=5**. The K parameter is used to determine distance weighting as explained above in the **Overview** section. The 10 nearest neighbors were determined using a search in all cells adjacent the cell in which the test or evaluate data points were located. This was achieved by setting **numcell** to a large number. The learning points were distributed in a tree of height 8 (**`treeheight=8`**) and one model (**`best_models=1`**) was saved in the **wssdemo.ycf** file. The values of WSS for all 1D models and the 2D models other than X2 and X4 were huge. This implies that the differences between the actual and calculated values of Y were on average very large.

```
Output report:  FKR Analysis Sat Mar 15 10:10:23 2014
                Version 3.32 (Professional) Feb 26, 2014

Parameters used in analysis    PARAMETER FILE: wssdemo.fpa
                               INPUT DATA FILE: wssdemo.txt

 DMIN   - min number of dimensions               :     1
 DMAX   - max number of dimensions               :     2
 NCP    - number of candidate predictors         :     5
 NCOL   - number of columns of data              :    10
 NREC   - total number of records                : 10000
 NFOLDS - number of evaluation folds             :     1
 LTYPE  - Learning set type (G, M or S)          :     S
          G is growing, M is moving and S is static
 NUMCELL- Number of cells searched for NUMNN     :   100
 NUMNN  - Number of nearest neighbors            :    10
 YSCALE - Multiplier for Y values                : 10000
 STARTVAR - starting variable                    :     1
 DELTA  - Min incremental change         :      -100.00
 NODATA - For Ycalc (columns of YCF file):       -999.00

 Kernel Regression parameters:
 MOD_CRITERION- 1=VR 2=CC 3=CC_O 4=Sep  5=WSS    :     5
 FIT_ORDER - 0 is average,  1 & 2 are surfaces   :     1
 NUMK - Number of smoothing parameters           :     2
 K(1) - Smoothing parameter 1                    :  1.00
 K(2) - Smoothing parameter 2                    :  5.00
 YCOL - Column of dependent variable             :    10

 Tree parameters:
 BUCKETSIZE - design number per cell (computed)  :    27
 TREEHEIGHT - tree parm                          :     8
 Computed number of cells                        :   511
 Computed number of leaf cells                   :   256
 Computed avg bucket size                        :  27.3
 Computed 2 Sigma limit for F Stat               :  1.18

 Data set parameters:
 NLRN   - number of learning records             :  7000
 NTST   - number of test records                 :  2000
 NEVL   - number of evaluation records           :  1000
 GAP    - gap records between data sets          :     0
 STARTLRN - starting learning record             :     1
 STARTTST - starting test record                 :  7001
 STARTEVL - starting evaluation record           :  9001

Ordered results for 1 dimensional models:
   Total number of combinations:       5
   Number tested             :       5
   survivenum( 1)            :       5
   Number of survivors       :       5
WSS:  30899.805   F: 20.43   K: 5.00   X:   4
WSS:  31203.477   F: 23.43   K: 1.00   X:   2
WSS:  61217.560   F:  1.01   K: 1.00   X:   1
WSS:  61634.920   F:  1.02   K: 5.00   X:   3
WSS:  63078.280   F:  1.12   K: 1.00   X:   5
```

```
Ordered results for 2 dimensional models:
   Total number of combinations:      10
   Number tested                :      10
   survivenum( 2)               :      10
   Number of survivors          :       5
WSS:       1.335   F: 1919.5   K: 5.00   X:   2    4
WSS:  27200.622   F: 22.61    K: 1.00   X:   2    3
WSS:  28797.320   F: 20.16    K: 1.00   X:   1    4
WSS:  30759.637   F: 22.55    K: 5.00   X:   1    2
WSS:  30773.303   F: 19.85    K: 1.00   X:   4    5

Best Model Report Fold 1 (Test Set Data):
Model: 1 WSS:   1.335   F: 1919.49   K:  5.00    X:   2    4

Evaluation Report Fold 1: (Evaluation Data)
Model: 1  Dim: 2   WSS:    1.220
```

The **wssdemo.ycf** file includes an output line for each of the 1000 records in the evaluation data set. The first and last 5 records of the file are included and also record 9751. The model value for this record is -999 the **NODATA** code. Whenever a value of one of the X's in an evaluation data point is out of range of the combined learning and test data points, a **NODATA** value is included in the ycf file and the point is excluded from the calculation of **WSS**.

```
Model  1:  X:   2    4    K:  5.00
  Record              Y       Model  1
    9001  2.603000e-001  2.701112e-001
    9002  5.075000e-001  5.076262e-001
    9003  1.357000e-001  1.366367e-001
    9004  9.000000e-004  8.867934e-004
    9005  7.029000e-001  7.007145e-001
                    ***
                    ***
    9751  2.131000e-001 -9.990000e+002
                    ***
    9996  1.060000e-002  1.127342e-002
    9997  1.314000e-001  1.363564e-001
    9998  2.385000e-001  2.366260e-001
    9999  5.906000e-001  5.866586e-001
   10000  2.790000e-002  2.591561e-002
```

As an additional analysis, consider the **memo**.**fpa** file shown above in the section **Defining the Analysis**.. The following results were obtained using this file plus an artificial data file **memo.pri**. The values in the first 10 columns (i.e., **X(1)** through **X(10)**) were created using a random number generator. The values in column 19 (i.e, **YCOL**) were created using a non-linear function based upon columns 1, 2 and 4 plus 75% noise. In other words, the upper limit for this artificial problem is a Variance Reduction of 25%. In the results below we see that the best model in each of the 3 folds was the combination of X1, X2 and X4. In fold 1 the value of VarRed is 17.3%, in fold 2 it is17.0% and in fold 3 it is 19.6%. The FracSS (i.e., Fraction Same Sign) for fold 1 is 0.63, which means that the value of Ycalc and Y for the 1000 test points had the same sign 63% of the time. The value of F=8.93 for fold 1 was way above the 2 sigma significance limit of 1.36 for the F statistic. Results for folds 2 and 3 are similar. Another interesting point to note is that even though **DMAX** was set as 5, no 4D (i.e., 4 dimensional) spaces survived (due to the **DELTA** criterion) and therefore the analysis was terminated. Finally, the results in the out-of-sample evaluation data set showed that the best model continued to perform well. The variance reduction was 23.3% and the FracSS was 0.65 in fold 1, 20.5% and 0.61 in fold 2 and 22.1% and 0.63 in fold 3 It should be noted that all learning and test points are used to calculate *y* in the evaluation set. The use of this additional data when going from the test to the evaluation data set improved values of VarRed for this artificial data set.

```
Output report:  FKR Analysis Tue Aug 24 09:01:54 1999
                 Version 3.14 (Professional) Aug 10, 1999

 Parameters used in analysis    PARAMETER FILE: memo.fpa
                               INPUT DATA FILE: memo.pri
 DMIN   - min number of dimensions               :     1
 DMAX   - max number of dimensions               :     5
 NCP    - number of candidate predictors         :    10
 NCOL   - number of columns of data              :    20
 NREC   - total number of records                : 25000
 NFOLDS - number of evaluation folds             :     3
 LTYPE  - Learning set type (G, M or S)          :     G
          G is growing, M is moving and S is static
 GROWTH_FACTOR - (max_num/leaf)/(num/leaf)       :  3.00
 NUMCELL- Number of cells searched for NUMNN     :     1
 NUMNN  - Irrelevant because FAST mode specified
 STARTVAR - starting variable                    :     1
 DELTA  - Minimum incremental change      :         1.00
 NODATA - For Ycalc (columns of YCF file):       -999.00

 Kernel Regression parameters:
 MOD_CRITERION- 1=VR, 2=CC, 3=CC_O, 4=Separation :     1
 FIT_ORDER - 0 is average,  1 & 2 are surfaces   :     1
   FAST option used: Ycalc from cell hyperplane
 NUMK - Number of smoothing parameters           :     1
 K(1) - Smoothing parameter 1                    :  1.00
 YCOL - Column of dependent variable             :    19

 Tree parameters:
 BUCKETSIZE - design number per cell (computed)  :    32
 TREEHEIGHT - tree parm                          :     6
 Computed number of cells                        :   127
 Computed number of leaf cells                   :    64
 Computed avg bucket size                        :  32.0
 Computed 2 Sigma limit for F Stat               :  1.36

 Data set parameters for fold 1 of 3 folds:
 NLRN   - number of learning records             :  2048
 NTST   - number of test records                 :  1000
 NEVL   - number of evaluation records           :   500
 GAP    - gap records between data sets          :     0
 STARTLRN - starting learning record             :     1
 STARTTST - starting test record                 :  2049
 STARTEVL - starting evaluation record           :  3049

Ordered results for 1 dimensional models:
   Total number of combinations:      10
   Number tested               :      10
   survivenum( 1)              :      10
   Number of survivors         :      10
Var Red:   4.127   FracSS: 0.575   F:  3.87   X:   4
Var Red:   3.855   FracSS: 0.577   F:  3.53   X:   2
Var Red:   2.955   FracSS: 0.575   F:  3.62   X:   1
Var Red:  -1.759   FracSS: 0.537   F:  0.70   X:   6
Var Red:  -3.041   FracSS: 0.515   F:  0.84   X:   3
Var Red:  -4.482   FracSS: 0.492   F:  0.54   X:  10
Var Red:  -5.450   FracSS: 0.481   F:  0.63   X:   7
Var Red:  -5.997   FracSS: 0.503   F:  0.58   X:   9
Var Red:  -6.368   FracSS: 0.486   F:  0.45   X:   5
Var Red:  -7.871   FracSS: 0.487   F:  0.54   X:   8
```

```
Ordered results for 2 dimensional models:
   Total number of combinations:      45
   Number tested               :      45
   survivenum( 2)              :       5
   Number of survivors         :       3
Var Red:  12.203   FracSS: 0.596   F:  6.37   X:   1    2
Var Red:  10.113   FracSS: 0.593   F:  6.53   X:   2    4
Var Red:   9.953   FracSS: 0.595   F:  7.18   X:   1    4

Ordered results for 3 dimensional models:
   Total number of combinations:     120
   Number tested               :      22
   survivenum( 3)              :       3
   Number of survivors         :       2
Var Red:  17.275   FracSS: 0.630   F:  8.93   X:   1    2    4
Var Red:  11.789   FracSS: 0.615   F:  5.79   X:   2    4    8

Ordered results for 4 dimensional models:
   Total number of combinations:     210
   Number tested               :      13
   survivenum( 4)              :       3
   Number of survivors         :       0

Best Model Report Fold 1 (Test Set Data):
Model: 1 Var Red:  17.275   FracSS: 0.630   F:  8.93   X:   1    2    4

Evaluation Report Fold 1: from 860909 to 881007 (Evaluation Data)
Model: 1  Dim: 3   Var Red:   23.341   FracSS: 0.648


 Data set parameters for fold 2 of 3 folds:
 NLRN   - number of learning records            :  2048
 NTST   - number of test records                :  1000
 NEVL   - number of evaluation records          :   500
 GAP    - gap records between data sets         :     0
 STARTLRN - starting learning record            :   501
 STARTTST - starting test record                :  2549
 STARTEVL - starting evaluation record          :  3549

Ordered results for 1 dimensional models:
   Total number of combinations:      10
   Number tested               :      10
   survivenum( 1)              :      10
   Number of survivors         :      10
Var Red:   2.839   FracSS: 0.555   F:  3.46   X:   2
Var Red:   2.790   FracSS: 0.555   F:  3.79   X:   1
Var Red:   1.203   FracSS: 0.566   F:  4.00   X:   4
Var Red:  -4.708   FracSS: 0.522   F:  0.77   X:   6
Var Red:  -4.813   FracSS: 0.515   F:  0.69   X:  10
Var Red:  -4.981   FracSS: 0.518   F:  0.61   X:   8
Var Red:  -5.527   FracSS: 0.502   F:  0.56   X:   9
Var Red:  -5.993   FracSS: 0.491   F:  0.57   X:   7
Var Red:  -7.198   FracSS: 0.496   F:  0.63   X:   3
Var Red:  -7.900   FracSS: 0.486   F:  0.60   X:   5
```

```
Ordered results for 2 dimensional models:
   Total number of combinations:      45
   Number tested               :      45
   survivenum( 2)              :       5
   Number of survivors         :       3
Var Red:  11.291   FracSS: 0.584   F:  6.54   X:   1    2
Var Red:   9.535   FracSS: 0.602   F:  6.99   X:   1    4
Var Red:   8.071   FracSS: 0.573   F:  6.48   X:   2    4

Ordered results for 3 dimensional models:
   Total number of combinations:     120
   Number tested               :      22
   survivenum( 3)              :       3
   Number of survivors         :       1
Var Red:  16.996   FracSS: 0.626   F:  9.05   X:   1    2    4

Ordered results for 4 dimensional models:
   Total number of combinations:     210
   Number tested               :       7
   survivenum( 4)              :       3
   Number of survivors         :       0

Best Model Report Fold 2 (Test Set Data):
Model: 1 Var Red:  16.996   FracSS: 0.626   F:  9.05   X:   1    2    4

Evaluation Report Fold 2: from 881009 to 901107 (Evaluation Data)
Model: 1  Dim: 3   Var Red:   20.476   FracSS: 0.606


 Data set parameters for fold 3 of 3 folds:
 NLRN   - number of learning records          :  2048
 NTST   - number of test records              :  1000
 NEVL   - number of evaluation records        :   500
 GAP    - gap records between data sets       :     0
 STARTLRN - starting learning record          :  1001
 STARTTST - starting test record              :  3049
 STARTEVL - starting evaluation record        :  4049

Ordered results for 1 dimensional models:
   Total number of combinations:      10
   Number tested               :      10
   survivenum( 1)              :      10
   Number of survivors         :      10
Var Red:   6.827   FracSS: 0.612   F:  4.30   X:   4
Var Red:   3.585   FracSS: 0.572   F:  3.68   X:   1
Var Red:   3.365   FracSS: 0.557   F:  3.37   X:   2
Var Red:  -3.845   FracSS: 0.516   F:  1.08   X:   8
Var Red:  -3.852   FracSS: 0.514   F:  0.67   X:   7
Var Red:  -3.902   FracSS: 0.491   F:  0.64   X:   9
Var Red:  -4.306   FracSS: 0.506   F:  0.81   X:   3
Var Red:  -4.652   FracSS: 0.516   F:  0.61   X:   5
Var Red:  -4.690   FracSS: 0.512   F:  0.74   X:   6
Var Red:  -5.088   FracSS: 0.503   F:  0.62   X:  10
```

```
Ordered results for 2 dimensional models:
   Total number of combinations:      45
   Number tested               :      45
   survivenum( 2)              :       5
   Number of survivors         :       3
Var Red:  13.586   FracSS: 0.607   F:  6.80   X:   2    4
Var Red:  10.276   FracSS: 0.603   F:  6.73   X:   1    4
Var Red:   9.436   FracSS: 0.580   F:  6.15   X:   1    2

Ordered results for 3 dimensional models:
   Total number of combinations:     120
   Number tested               :      22
   survivenum( 3)              :       3
   Number of survivors         :       2
Var Red:  19.594   FracSS: 0.608   F:  9.09   X:   1    2    4
Var Red:  12.195   FracSS: 0.601   F:  6.14   X:   1    4    8

Ordered results for 4 dimensional models:
   Total number of combinations:     210
   Number tested               :      13
   survivenum( 4)              :       3
   Number of survivors         :       0

Best Model Report Fold 3 (Test Set Data):
Model: 1 Var Red:  19.594   FracSS: 0.608   F:  9.09   X:   1    2    4

Evaluation Report Fold 3: from 901109 to 921207 (Evaluation Data)
Model: 1  Dim: 3   Var Red:   22.077   FracSS: 0.633
```