

User Manual for GUIDE version 11*

Wei-Yin Loh
Department of Statistics
University of Wisconsin–Madison

October 28, 2011

Contents

1	Warranty disclaimer	2
2	Introduction	3
3	What’s new in version 11	4
4	Program input files	6
5	Interactive use	8
6	Classification examples	9
6.1	Iris data	9
6.1.1	Interactive dialog	9
6.1.2	Contents of <code>irisout.txt</code>	11
6.2	Hepatitis data — unequal costs	15
6.2.1	Session log	17
6.2.2	Results	19

*Based upon work partially supported by grants from the U.S. Army Research Office, the National Science Foundation and the National Institutes of Health.

7	Regression examples	26
7.1	Baseball data	26
7.1.1	Stepwise linear regression	28
7.1.2	Best simple linear	36
7.2	Tuition data with missing values	44
7.2.1	Best simple ANCOVA	46
7.3	Quantile regression	54
7.4	Longitudinal data	67
7.4.1	Session log	69
7.4.2	Contents of output file <code>cd4out.txt</code>	73
7.5	Multi-response data	76
7.5.1	Contents of <code>multi.txt</code>	81
8	Other features	92
8.1	Pruning with test samples	92
8.2	Prediction of test samples	92
8.3	Least median of squares, Poisson, and relative risk regression	92
8.4	Unattended (batch) operation	92
8.5	Forests and tree ensembles	93
8.6	Importance scoring and ranking of variables	97
8.7	Automatic generation of powers and products	99
8.8	Data formatting functions	100

1 Warranty disclaimer

Redistribution and use in binary forms, with or without modification, are permitted provided that the following condition is met:

Redistributions in binary form must reproduce the above copyright notice, this condition and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY WEI-YIN LOH “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL WEI-YIN LOH BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER

CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The views and conclusions contained in the software and documentation are those of the author and should not be interpreted as representing official policies, either expressed or implied, of the University of Wisconsin.

2 Introduction

GUIDE stands for *Generalized, Unbiased, Interaction Detection and Estimation*. It is the only classification and regression tree algorithm with all these features:

1. Unbiased variable selection.
2. Kernel and nearest-neighbor node models for classification trees.
3. Weighted least squares, least median of squares, quantile, Poisson, and relative risk (proportional hazards) regression models.
4. Univariate, multivariate, and longitudinal response variables.
5. Pairwise interaction detection at each node.
6. Linear splits on two variables at a time for classification trees.
7. Categorical variables for splitting only, or for both splitting and fitting (via 0-1 dummy variables), in regression tree models.
8. Ranking and scoring of predictor variables.
9. Tree ensembles (bagging and forests).

Tables 1 and 2 compare the features of GUIDE with CRUISE (Kim and Loh, 2001, 2003), QUEST (Loh and Shih, 1997), C4.5 (Quinlan, 1993), RPART¹, and M5' (Quinlan, 1992; Witten and Frank, 2000).

GUIDE is available free from www.stat.wisc.edu/~loh/guide.html in the form of compiled executables for Linux, Mac OS X, and Windows on Intel and compatible

¹RPART is an implementation of CART (Breiman et al., 1984) in R. CART is a registered trademark of California Statistical Software, Inc.

Table 1: Comparison of GUIDE, QUEST, CRUISE, CART, and C4.5 classification tree algorithms. Node models: S = simple, K = kernel, L = linear discriminant, N = nearest-neighbor.

	GUIDE	QUEST	CRUISE	CART	C4.5
Unbiased splits	Yes	Yes	Yes	No	No
Splits per node	2	2	≥ 2	2	2
Interaction detection	Yes	No	Yes	No	No
Importance ranking	Yes	No	No	Yes	No
Class priors	Yes	Yes	Yes	Yes	No
Misclassification costs	Yes	Yes	Yes	Yes	No
Linear splits	Yes	Yes	Yes	Yes	No
Categorical splits	Subsets	Subsets	Subsets	Subsets	Atoms
Node models	S, K, N	S	S, L	S	S
Missing values	Special	Imputation	Surrogate	Surrogate	Weights
Tree diagrams	Text and L ^A T _E X			Proprietary	Text
Bagging	Yes	No	No	No	No
Forests	Yes	No	No	No	No

processors. This manual illustrates the use of the program and interpretation of the output.

The GUIDE algorithm is documented in Loh (2002) for regression trees and Loh (2009) for classification trees. Loh (2008a) and Loh (2010) give brief reviews of the subject. Extensions of the algorithm as well as other features and capabilities are reported in Chaudhuri and Loh (2002), Loh (2006b), Kim et al. (2007), Loh et al. (2007), and Loh (2008b). For a list of third-party applications of GUIDE, CRUISE, QUEST, and the logistic regression tree algorithm LOTUS (Chan and Loh, 2004; Loh, 2006a), see <http://www.stat.wisc.edu/~loh/apps.html>

3 What's new in version 11

Besides bug fixes, version 11 adds the ability to construct regression trees for longitudinal data with unbalanced and unequal time points. It also brings back the

Table 2: Comparison of GUIDE, CART and M5' regression tree algorithms

	GUIDE	CART	M5'
Unbiased splits	Yes	No	No
Pairwise interaction detection	Yes	No	No
Importance scores	Yes	Yes	No
Loss functions	Weighted least squares, least median of squares, quantile, Poisson, proportional hazards	Least squares, least absolute deviations	Least squares only
Survival, longitudinal and multi-response data	Yes, yes, yes	No, no, no	No, no, no
Node models	Constant, multiple, stepwise linear, polynomial, ANCOVA	Constant only	Constant and linear
Linear models	Multiple or stepwise (forward-backward and forward only)	N/A	Stepwise
Variable roles	Split only, fit only, both, neither, weight, censored, offset	Split only	Split and fit
Categorical variable splits	Subsets of categorical values	Subsets	0-1 variables
Tree selection	Pruning or stopping rules	Pruning only	Pruning only
Tree diagrams	Text and \LaTeX	Proprietary	Text
Operation modes	Interactive and batch	Interactive and batch	Interactive
Case weights	Yes	Yes	No
Transformations	Powers and products	No	No
Missing values	Missing values treated as a special category	Surrogate splits	Imputation
Bagging & forests	Yes, yes	No, no	No, no
Data conversions	ARFF, C4.5, Minitab, R, SAS, Statistica, Systat, CSV	No	No

pruning option of using the median of cross-validation estimates of error.

4 Program input files

The GUIDE program requires two text files for input.

Data file: This file contains the training sample. Each file record consists of observations on the response (i.e., dependent) variable, the predictor (i.e., X or independent) variables, and optional weight and survival time variables. The entries in each record must be comma, space, or tab delimited. A record can occupy more than one line in the file, but each record must begin on a new line. Data values can be numbers (including scientific notation) or character strings. Categorical variables can be coded as numbers or character strings. If a string contains characters other than numbers or alphabets, it must be surrounded by a matching pair of single or double quotation marks. Data values cannot exceed 21 characters in length.

Description file: This file is used to provide information to the program about the name and location of the data file, the names and column positions of the variables, and their roles in the analysis. Different analyses of the same dataset may be carried out by altering the roles of the variables in this file. The file `irisdsc.txt` included with the distribution is an example description file. Its contents are:

```
irisdata.txt
"?"
column, varname, vartype
1 sepallen n
2 sepalwid n
3 petallen n
4 petalwid n
5 class d
```

The data give the sepal lengths and widths and the petal lengths and widths of 150 iris flowers. The response variable is the type of iris flower.

The first line of the file `irisdsc.txt` gives the name of the training sample file. If the data file `irisdata.txt` is not in the folder where GUIDE is installed, its full path (such as `"c:\data\irisdata.txt"`) is needed. The second line gives

the code that denotes a missing value in the data. The missing value code can be up to 80 characters long. If it contains characters other than alphabets or numbers, it must be surrounded by quotation marks. A missing value code must appear in the second line of the file even if there are no missing values in the data (in that case any character string not present among the data values can be used). The third line contains three character strings to indicate the column headers of the subsequent lines. The position, name and role of each variable comes next (in that order), with one line for each variable. Only the first ten characters in a variable name are printed in the output. And only alphabets (lower or upper case) and numbers may be used in a variable name. The following roles for the variables are permitted. Lower and upper case letters are accepted.

- b** Categorical variable that is used both for splitting and for node modeling in regression. It is transformed to 0-1 dummy variables for node modeling. It is converted to “c” for classification.
- c** Categorical variable. It is used for splitting only.
- d** Dependent variable. Except for multi-response data (see Sec. 7.5), there can only be one such variable. In the case of relative risk models, this is the **d**eath indicator. The variable can take character string values for classification.
- f** Numerical variable used only for fitting the linear models in the nodes of the tree. It is not used for splitting the nodes and is disallowed in classification.
- n** Numerical variable used both for splitting the nodes and for fitting the node models. It is converted to type “s” in classification.
- r** Categorical treatment (**R**x) variable used only for fitting the linear models in the nodes of the tree. It is not used for splitting the nodes.
- s** Numerical-valued variable only used for splitting the nodes. It is not used as a regressor in the linear models. This role is suitable for ordinal categorical variables if they are given numerical values that reflect the orderings.
- t** Survival time or observation time variable. This variable type is only allowed for relative risk (proportional hazards) models or longitudinal data, respectively.
- w** Weight variable. It can be used in two ways: (1) to fit a weighted least squares regression model, and (2) to obtain predicted values for a test

sample. See section 8.2 for the latter. A record with a missing value in a **d**, **t**, or **z**-variable is automatically given 0 weight.

- x** Variable **ex**cluded from the analysis. The excluded variables may be categorical or numerical. This allows multiple applications of GUIDE to different subsets of variables without requiring reformats of the data file.
- z** **Offset** variable. It is only allowed if the Poisson regression option is chosen.

5 Interactive use

The GUIDE program is executed by typing its name in a shell window. Following is an example session log with annotations printed in *red italics*. Whenever you are prompted for a selection, there is usually range of permissible values given within square brackets and a default choice (indicated by the symbol `<cr>=`). The default may be selected by pressing the ENTER or RETURN key.

When the program starts, the user is asked to select one of five options:

```
Choose one of the following options:
1. Read the warranty disclaimer
2. Fit a model
3. Convert data to other formats
4. Create a batch input file
5. Rank and select regressor variables
```

The meanings of these options are:

1. Print the warranty disclaimer.
2. Construct a classification or regression tree model.
3. Convert the datafile into a format suitable for importation into database, spreadsheet, or statistics software. See Table 2 for the statistical packages supported and Section 8.8 for an example.
4. Create an input file for unattended (batch) mode operation.
5. Obtain the importance scores of the variables and identify the important ones.

6 Classification examples

6.1 Iris data

We first show how to obtain a classification tree from the data in the `irisdata.txt` file by selecting option 2.

6.1.1 Interactive dialog

```

Input your choice: 2
Input name of file to store results: irisout.txt
This is the name of the file to store the results.
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Choose 1 for a single tree.
You can fit a classification tree or a regression tree
Input 1 for classification, 2 for regression ([1:2], <cr>=1):
Choose 1 if the dependent variable is categorical.
Input 1 for simple, 2 for nearest-neighbor, 3 for kernel method ([1:3], <cr>=1):
Input 1 for linear and interaction splits,
    2 to skip linear splits, 3 to skip both
Input your choice ([1:3], <cr>=1):
Input 1 to prune by CV, 2 by test sample, 3 for no pruning ([1:3], <cr>=1):
CV stands for cross-validation. Choose option 3 if you want an unpruned tree.
Input name of data description file (max 100 chars; enclose within
quotes if it contains spaces): irisdesc.txt
The program starts to read the description file.
Reading data description file ...
Training sample file: irisdata.txt
Missing value code: ?
Length of longest data entry = 11
Total number of cases =      150
Number of classes =          3
Checking data ...
Class name      Num. cases  Proportion
Setosa          50  0.33333333
Versicolour     50  0.33333333
Virginica       50  0.33333333
  Total  #cases w/  #cases w/
  #cases  miss. D  miss. val  #X-var  #N-var  #F-var  #S-var  #B-var  #C-var
    150     0      0      0      0      0      4      0      0
Number of cases used for training =      150
Default number of cross-validations =      10
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Choose 2 if you want to change the number of cross-validation steps.
Best tree may be chosen based on mean or median CV estimate

```

Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
The default uses the mean; choose 2 if you want to use the median.
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50):
You can control the size of the pruned tree; small numbers produce bigger trees.
Choose 1 for estimated priors, 2 for equal priors,
3 to input the priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
If you choose option 3, you will be asked to give the name of a file containing the prior probability of each class.
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
If you choose option 2, you will be asked to give the name of a file containing the misclassification cost matrix.
Choose a split point selection method for numerical variables:
Choose 1 to use faster method based on sample quantiles
Choose 2 to use exhaustive search
Input 1 or 2 ([1:2], <cr>=2):
Choose option 1 if you want to shorten computation time.
Default max number of split levels = 10
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Choose 2 if you want to change the max. depth of the tree before pruning.
Smallest node sample size before pruning = 5
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Choose 2 to change the smallest permissible node size before pruning.
Input 2 for LaTeX tree diagrams, 1 to skip them ([1:2], <cr>=1):2
Input file name to store LaTeX code (use .tex as suffix): iris.tex
Input 1 for a vertical tree, 2 for a sideways tree ([1:2], <cr>=1):
Choose 2 if you want the tree diagram to grow sideways; see Figure 6 for an example.
Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1):
Input 1 to number all nodes, 2 to number leaves only ([1:2], <cr>=1):
Input 1 to color leaf nodes, 2 otherwise ([1:2], <cr>=1):
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Options 2 and 3 save some information in other files for further processing.
Input 2 to save predicted values and node IDs, 1 otherwise ([1:2], <cr>=1):
Choose 2 to save the predicted values in another file.
Input 2 to save terminal node IDs for importance scoring later; 1 otherwise ([1:2], <cr>=1):
This option is for setting up a follow-up run for importance scoring; see the section on importance ranking below.
Constructing main tree ...
Number of terminal nodes of largest tree = 4
Performing cross-validation:
Finished cross-validation iteration 1
Finished cross-validation iteration 2

```

Finished cross-validation iteration      3
Finished cross-validation iteration      4
Finished cross-validation iteration      5
Finished cross-validation iteration      6
Finished cross-validation iteration      7
Finished cross-validation iteration      8
Finished cross-validation iteration      9
Finished cross-validation iteration     10

```

Pruning main tree. Please wait.

Results of subtree sequence

Trees based on mean with naive SE are marked with * and **

Tree based on mean with bootstrap SE is marked with --

Trees based on median with finite bootstrap SE are marked with + and ++

Subtree	#Terminal nodes
0**	4
1	3
2	2
3	1

* tree, ** tree, + tree, and ++ tree all the same

The pruned tree is marked with two asterisks.

LaTeX code for tree is in file: iris.tex

Results are stored in file: irisout.txt

6.1.2 Contents of irisout.txt

Classification tree

Pruning by cross-validation

Data description file: irisdisc.txt

Training sample file: irisdata.txt

Missing value code: ?

Length of longest data entry = 11

Number of classes = 3

Class name	Num. cases	Proportion
Setosa	50	0.33333333
Versicolour	50	0.33333333
Virginica	50	0.33333333

Summary information (without x variables)

d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical,

n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight

For categorical variables, #categories include one for missing values

Column number	Variable name	Variable type	Minimum value	Maximum value	Number of categories	Number missing
1	sepalen	s	4.3000E+00	7.9000E+00		

```

2  sepalwid      s      2.0000E+00  4.4000E+00
3  petallen      s      1.0000E+00  6.9000E+00
4  petalwid      s      1.0000E-01  2.5000E+00
5  class         d

```

3

```

Total #cases w/ #cases w/
#cases miss. D miss. val #X-var #N-var #F-var #S-var #B-var #C-var
150      0      0      0      0      0      0      4      0      0
Number of cases used for training = 150

```

Interaction tests on all variables

Linear splits

Pruning by v-fold cross-validation, with v = 10

Selected tree is based on mean of CV estimates

Simple node models

Estimated priors

Unit misclassification costs

Split values for N and S variables based on exhaustive search

Max number of split levels = 10

Minimum node size = 5

Number of SE's for pruned tree = 5.0000E-01

Size and CV mean cost and SE of subtrees:

Tree	#Tnodes	Mean Cost	SE(Mean)	BSE(Mean)	Median Cost	BSE(Median)
0**	4	3.333E-02	1.466E-02	1.403E-02	0.000E+00	2.454E-02
1	3	4.667E-02	1.722E-02	1.579E-02	3.333E-02	3.111E-02
2	2	3.333E-01	3.849E-02	0.000E+00	3.333E-01	0.000E+00
3	1	6.667E-01	3.849E-02	0.000E+00	6.667E-01	0.000E+00

Column 2 gives the number of terminal nodes in each tree.

Column 3 gives the CV estimates of misclassification cost.

Column 4 gives naive estimates of standard errors (SE).

Column 5 gives bootstrap estimates of standard errors.

Column 6 gives median estimates of misclassification cost.

Column 7 gives bootstrap estimates of SE of the estimated median costs.

0-SE tree based on mean is marked with *

Selected-SE tree based on mean using naive SE is marked with **

Selected-SE tree based on mean using bootstrap SE is marked with --

0-SE tree based on median with finite bootstrap SE is marked with +

Selected-SE tree based on median and bootstrap SE is marked with ++

* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean with naive SE estimate.

Structure of final tree. Each terminal node is marked with a T.

Node cost is node misclassification cost divided by number of training cases

Node label	Total cases	Train cases	Predicted class	Node cost	Split variable	Interacting variable
1	150	150	Setosa	6.667E-01	petalwid	
2T	50	50	Setosa	0.000E+00		
3	100	100	Versicolour	5.000E-01	petalwid	
6	54	54	Versicolour	9.259E-02	petallen	
12T	48	48	Versicolour	2.083E-02		
13T	6	6	Virginica	3.333E-01		
7T	46	46	Virginica	2.174E-02		

Column 2 gives the sample size in each node.

Column 3 gives the number of cases used to provide the predicted class in Column 4. The entries in these two columns differ only if there are missing values or cases with 0 weights.

Column 5 gives the estimated misclassification cost at the node.

Column 6 gives split variable name.

If the split is due to an interaction, the interacting variable is in Column 7.

Number of terminal nodes of final tree: 4
Total number of nodes of final tree: 7

Classification tree:

```

Node 1: petalwid <= 0.80000
  Node 2: Setosa
Node 1: petalwid > 0.80000
  Node 3: petalwid <= 1.75000
    Node 6: petallen <= 4.95000
      Node 12: Versicolour
      Node 6: petallen > 4.95000
        Node 13: Virginica
    Node 3: petalwid > 1.75000
      Node 7: Virginica

```

This is the tree structure in tabbed text form.

The tree diagram in Figure 1 is obtained by LaTeX from the file irisout.tex.

Detailed information about the node compositions are given next.

```

Node 1: Intermediate node
A case goes into Node 2 if petalwid <= 8.0000000E-01
      petalwid mean = 1.1987E+00
ClassName      Number ClassPrior
Setosa          50      0.3333
Versicolou     50      0.3333
Virginica       50      0.3333

```

```

Number of training cases misclassified =          100
-----
Node 2: Terminal node
ClassName      Number ClassPrior
Setosa         50      1.0000
Versicolou    0       0.0000
Virginica     0       0.0000
Predicted class is Setosa
Number of training cases misclassified =          0
-----
Node 3: Intermediate node
A case goes into Node 6 if petalwid <= 1.7500000E+00
      petalwid mean = 1.6760E+00
ClassName      Number ClassPrior
Setosa         0       0.0000
Versicolou    50      0.5000
Virginica     50      0.5000
Number of training cases misclassified =          50
-----
Node 6: Intermediate node
A case goes into Node 12 if petallen <= 4.9500000E+00
      petallen mean = 4.3370E+00
ClassName      Number ClassPrior
Setosa         0       0.0000
Versicolou    49      0.9074
Virginica     5       0.0926
Number of training cases misclassified =          5
-----
Node 12: Terminal node
ClassName      Number ClassPrior
Setosa         0       0.0000
Versicolou    47      0.9792
Virginica     1       0.0208
Predicted class is Versicolour
Number of training cases misclassified =          1
-----
Node 13: Terminal node
ClassName      Number ClassPrior
Setosa         0       0.0000
Versicolou    2       0.3333
Virginica     4       0.6667
Predicted class is Virginica
Number of training cases misclassified =          2
-----
Node 7: Terminal node
ClassName      Number ClassPrior

```

```

Setosa          0    0.0000
Versicolou     1    0.0217
Virginica      45    0.9783
Predicted class is Virginica
Number of training cases misclassified =          1
-----

```

LaTeX code for tree is in file: iris.tex

A summary of the classification results follows.

Classification matrix for training sample:

Predicted class	True class		
	Setosa	Versicolo	Virginica
Setosa	50	0	0
Versicolou	0	47	1
Virginica	0	3	49
Total	50	50	50

Number of cases used for tree construction = 150

Number misclassified = 4

Resubstitution estimate of mean misclassification cost =
2.666666666666666E-002

The classification tree drawn by LaTeX using the file `iris.tex` is shown in Figure 1.

6.2 Hepatitis data — unequal costs

We illustrate the use of unequal misclassification costs with the hepatitis data set from <http://archive.ics.uci.edu/ml/datasets/Hepatitis>. The data consist of observations from 155 individuals, of whom 32 are labeled “die” and 123 labeled “live”. The contents of the description file `hepdsc.txt` are:

```

hep.dat
"?"
column, var, type
1 CLASS d
2 AGE n
3 SEX c
4 STEROID c
5 ANTIVIRALS c
6 FATIGUE c
7 MALAISE c
8 ANOREXIA c

```

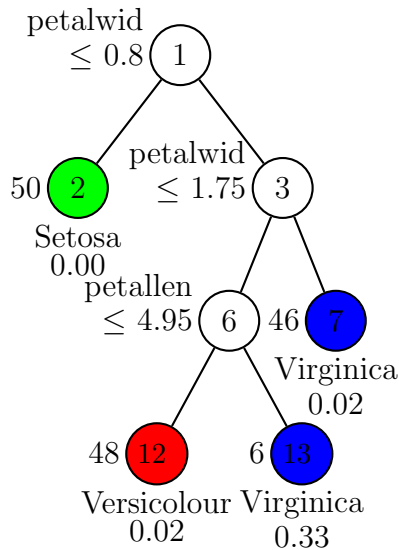


Figure 1: GUIDE classification tree model. At each intermediate node, a case goes to the left child node if and only if the condition is satisfied. Predicted class and mean misclassification cost beneath each leaf node; number of cases on its left.

```

9  BIGLIVER  c
10 FIRMLIVER c
11 SPLEEN   c
12 SPIDERS  c
13 ASCITES  c
14 VARICES  c
15 BILIRUBIN n
16 ALKPHOSPHATE n
17 SGOT     n
18 ALBUMIN  n
19 PROTIME  n
20 HISTOLOGY c
  
```

Since the number of “live” outnumber the number of “die” by almost 4 to 1 and since “die” is treated as class 1 and “live” as class 2 by GUIDE (because “die” precedes “live” alphabetically), we use the misclassification cost matrix

$$C = \begin{pmatrix} 0 & 1 \\ 4 & 0 \end{pmatrix}$$

where $C(i, j)$ denotes the cost of misclassifying a class j individual as class i . This matrix is saved in the text file `cost.txt` which has the following two lines:

```
0 1
4 0
```

6.2.1 Session log

Choose one of the following options:

1. Read the warranty disclaimer
2. Fit a model
3. Convert data to other formats
4. Create a batch input file
5. Rank and select regressor variables

Input your choice: 2

Input name of file to store results: hepout.txt

Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):

You can fit a classification tree or a regression tree

Input 1 for classification, 2 for regression ([1:2], <cr>=1):

Input 1 for simple, 2 for nearest-neighbor, 3 for kernel method ([1:3], <cr>=1):

Input 1 for linear and interaction splits,

2 to skip linear splits, 3 to skip both

Input your choice ([1:3], <cr>=1):

Input 1 to prune by CV, 2 by test sample, 3 for no pruning ([1:3], <cr>=1):

Input name of data description file (max 100 chars; enclose within quotes if it contains spaces): hepdesc.txt

Reading data description file ...

Training sample file: hep.dat

Missing value code: ?

Length of longest data entry = 6

Total number of cases = 155

Missing values found among categorical variables

Separate categories will be created for missing categorical variables

Number of classes = 2

Cat. var. in column	#levels (incl. missing)	#missing values
3	2	0
4	3	1
5	2	0
6	3	1
7	3	1
8	3	1
9	3	10
10	3	11
11	3	5
12	3	5
13	3	5

```

          14          3          5
          20          2          0
Checking data ...
Class name      Num. cases  Proportion
die             32         0.20645161
live           123         0.79354839
  Total #cases w/ #cases w/
  #cases miss. D miss. val #X-var #N-var #F-var #S-var #B-var #C-var
    155     0     72     0     0     0     6     0     13
Number of cases used for training =      155
Number of cases excluded due to zero weight or missing D-values =      0
Default number of cross-validations =      10
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Best tree may be chosen based on mean or median CV estimate
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50):
Choose 1 for estimated priors, 2 for equal priors,
3 to input the priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):2
Input the name of a file containing the cost matrix C(i|j),
where C(i|j) is the cost of classifying class j as class i
The rows of the matrix must be in alphabetical order of the class names
Input name of file: cost.txt
Choose a split point selection method for numerical variables:
Choose 1 to use faster method based on sample quantiles
Choose 2 to use exhaustive search
Input 1 or 2 ([1:2], <cr>=2):
Default max number of split levels =      10
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Smallest node sample size before pruning =      5
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Input 2 for LaTeX tree diagrams, 1 to skip them ([1:2], <cr>=1):2
Input file name to store LaTeX code (use .tex as suffix): hep.tex
Input 1 for a vertical tree, 2 for a sideways tree ([1:2], <cr>=1):
Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1):
Input 1 to number all nodes, 2 to number leaves only ([1:2], <cr>=1):
Input 1 to color leaf nodes, 2 otherwise ([1:2], <cr>=1):
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save predicted values and node IDs, 1 otherwise ([1:2], <cr>=1):
Input 2 to save terminal node IDs for importance scoring later; 1 otherwise ([1:2], <cr>=1):
Constructing main tree ...

```

```

Number of terminal nodes of largest tree =          13
Performing cross-validation:
Finished cross-validation iteration          1
Finished cross-validation iteration          2
Finished cross-validation iteration          3
Finished cross-validation iteration          4
Finished cross-validation iteration          5
Finished cross-validation iteration          6
Finished cross-validation iteration          7
Finished cross-validation iteration          8
Finished cross-validation iteration          9
Finished cross-validation iteration         10

```

Pruning main tree. Please wait.

Results of subtree sequence

Trees based on mean with naive SE are marked with * and **

Tree based on mean with bootstrap SE is marked with --

Trees based on median with finite bootstrap SE are marked with + and ++

Subtree	#Terminal nodes
1	12
2	11
3++	10
4**	9
5	6
6	4
7	3
8	2
9	1

* tree same as + tree

** tree same as -- tree

+ tree same as ++ tree

* tree same as ++ tree

LaTeX code for tree is in file: hep.tex

Results are stored in file: hepout.txt

6.2.2 Results

Classification tree

Pruning by cross-validation

Data description file: hepdsc.txt

Training sample file: hep.dat

Missing value code: ?

Warning: N variables changed to S

Dependent variable is CLASS

Length of longest data entry = 6

Missing values found among categorical variables

Separate categories will be created for missing categorical variables

Number of classes = 2

Class name	Num. cases	Proportion
die	32	0.20645161
live	123	0.79354839

Summary information (without x variables)

d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical, n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight

For categorical variables, #categories include one for missing values

Column number	Variable name	Variable type	Minimum value	Maximum value	Number of categories	Number missing
1	CLASS	d			2	
2	AGE	s	7.0000E+00	7.8000E+01		
3	SEX	c			2	
4	STEROID	c			3	1
5	ANTIVIRALS	c			2	
6	FATIGUE	c			3	1
7	MALAISE	c			3	1
8	ANOREXIA	c			3	1
9	BIGLIVER	c			3	10
10	FIRMLIVER	c			3	11
11	SPLEEN	c			3	5
12	SPIDERS	c			3	5
13	ASCITES	c			3	5
14	VARICES	c			3	5
15	BILIRUBIN	s	3.0000E-01	8.0000E+00		6
16	ALPKHOSPHATE	s	2.6000E+01	2.9500E+02		29
17	SGOT	s	1.4000E+01	6.4800E+02		4
18	ALBUMIN	s	2.1000E+00	6.4000E+00		16
19	PROTIME	s	0.0000E+00	1.0000E+02		67
20	HISTOLOGY	c			2	

Total #cases	#cases w/ miss. D	#cases w/ miss. val	#X-var	#N-var	#F-var	#S-var	#B-var	#C-var
155	0	72	0	0	0	6	0	13

Number of cases used for training = 155

Number of cases excluded due to zero weight or missing D-values = 0

Interaction tests on all variables

Linear splits

Pruning by v-fold cross-validation, with v = 10

Selected tree is based on mean of CV estimates

Simple node models

Estimated priors

Misclassification cost matrix from file cost.txt:

```

Predicted   True class
           die       live
die         0.000E+00 1.000E+00
live        4.000E+00 0.000E+00

```

Split values for N and S variables based on exhaustive search

Max number of split levels = 10

Minimum node size = 5

Number of SE's for pruned tree = 5.0000E-01

Size and CV mean cost and SE of subtrees:

Tree	#Tnodes	Mean Cost	SE(Mean)	BSE(Mean)	Median Cost	BSE(Median)
1	12	5.032E-01	8.722E-02	8.390E-02	4.479E-01	1.580E-01
2	11	5.032E-01	8.722E-02	8.390E-02	4.479E-01	1.580E-01
3++	10	5.032E-01	8.722E-02	8.390E-02	4.479E-01	1.580E-01
4**	9	5.226E-01	8.428E-02	7.747E-02	6.104E-01	1.642E-01
5	6	5.871E-01	9.250E-02	9.915E-02	5.333E-01	1.390E-01
6	4	5.935E-01	9.512E-02	9.737E-02	5.333E-01	1.308E-01
7	3	5.806E-01	8.980E-02	7.843E-02	5.667E-01	9.608E-02
8	2	6.516E-01	1.070E-01	6.011E-02	6.000E-01	6.399E-02
9	1	7.935E-01	3.251E-02	1.602E-02	7.750E-01	3.305E-02

0-SE tree based on mean is marked with *

0-SE tree based on median with bootstrap SE is marked with +

Selected-SE tree based on mean using naive SE is marked with **

Selected-SE tree based on mean using bootstrap SE is marked with --

Selected-SE tree based on median and bootstrap SE is marked with ++

* tree same as + tree

** tree same as -- tree

+ tree same as ++ tree

* tree same as ++ tree

Following tree is based on mean CV with naive SE estimate.

Structure of final tree. Each terminal node is marked with a T.

Node cost is node misclassification cost divided by number of training cases

Node label	Total cases	Train cases	Predicted class	Node cost	Split variable	Interacting variable
1	155	155	die	7.935E-01	ASCITES	
2T	20	20	die	3.000E-01	ALBUMIN	
3	135	135	live	5.333E-01	SPIDERS	
6	93	93	live	2.151E-01	MALAISE	
12T	71	71	live	5.634E-02		
13	22	22	live	7.273E-01	BILIRUBIN	+SGOT
26T	10	10	live	0.000E+00		

27T	12	12	die	6.667E-01	SGOT
7	42	42	die	6.905E-01	SEX
14T	7	7	live	0.000E+00	
15	35	35	die	6.286E-01	FIRMLIVER
30	21	21	die	7.619E-01	AGE
60	16	16	live	5.000E-01	BILIRUBIN
120T	11	11	live	0.000E+00	
121T	5	5	die	6.000E-01	
61T	5	5	die	4.000E-01	
31T	14	14	die	4.286E-01	HISTOLOGY

Number of terminal nodes of final tree: 9

Total number of nodes of final tree: 17

Classification tree:

Node 1: ASCITES = yes

Node 2: die

Node 1: ASCITES = ?, no

Node 3: SPIDERS = no

Node 6: MALAISE = no

Node 12: live

Node 6: MALAISE = ?, yes

Node 13: $-4.6048708E-03 * SGOT + BILIRUBIN \leq 6.2800075E-01$

Node 26: live

Node 13: $-4.6048708E-03 * SGOT + BILIRUBIN > 6.2800075E-01$

Node 27: die

Node 3: SPIDERS = ?, yes

Node 7: SEX = female

Node 14: live

Node 7: SEX = male

Node 15: FIRMLIVER = yes

Node 30: AGE ≤ 53.50000

Node 60: BILIRUBIN ≤ 1.85000

Node 120: live

Node 60: BILIRUBIN > 1.85000

Node 121: die

Node 30: AGE > 53.50000

Node 61: die

Node 15: FIRMLIVER = ?, no

Node 31: die

In the following the predictor node mean is mean of complete cases

Node 1: Intermediate node

```

A case goes into Node 2 if ASCITES = yes
      ASCITES mode = no
ClassName      Number ClassPrior
die            32      0.2065
live          123      0.7935
Number of training cases misclassified = 123
-----
Node 2: Terminal node
ClassName      Number ClassPrior
die            14      0.7000
live           6      0.3000
Predicted class is die
Number of training cases misclassified = 6
-----
Node 3: Intermediate node
A case goes into Node 6 if SPIDERS = no
      SPIDERS mode = no
ClassName      Number ClassPrior
die            18      0.1333
live          117      0.8667
Number of training cases misclassified = 18
-----
Node 6: Intermediate node
A case goes into Node 12 if MALAISE = no
      MALAISE mode = no
ClassName      Number ClassPrior
die             5      0.0538
live           88      0.9462
Number of training cases misclassified = 5
-----
Node 12: Terminal node
ClassName      Number ClassPrior
die             1      0.0141
live           70      0.9859
Predicted class is live
Number of training cases misclassified = 1
-----
Node 13: Intermediate node
A case goes into Node 26 if
-4.6048708E-03 * SGOT + BILIRUBIN <= 6.2800075E-01
      linear combination mean = 9.3485E-01
ClassName      Number ClassPrior
die             4      0.1818
live           18      0.8182
Number of training cases misclassified = 4
-----

```

```

Node 26: Terminal node
ClassName      Number ClassPrior
die            0      0.0000
live          10      1.0000
Predicted class is live
Number of training cases misclassified =  0
-----
Node 27: Terminal node
ClassName      Number ClassPrior
die            4      0.3333
live           8      0.6667
Predicted class is die
Number of training cases misclassified =  8
-----
Node 7: Intermediate node
A case goes into Node 14 if SEX = female
                SEX mode = male
ClassName      Number ClassPrior
die            13      0.3095
live           29      0.6905
Number of training cases misclassified =  29
-----
Node 14: Terminal node
ClassName      Number ClassPrior
die            0      0.0000
live           7      1.0000
Predicted class is live
Number of training cases misclassified =  0
-----
Node 15: Intermediate node
A case goes into Node 30 if FIRMLIVER = yes
                FIRMLIVER mode = yes
ClassName      Number ClassPrior
die            13      0.3714
live           22      0.6286
Number of training cases misclassified =  22
-----
Node 30: Intermediate node
A case goes into Node 60 if AGE <=  5.3500000E+01
                AGE mean =  4.5095E+01
ClassName      Number ClassPrior
die            5      0.2381
live           16      0.7619
Number of training cases misclassified =  16
-----
Node 60: Intermediate node

```

A case goes into Node 120 if BILIRUBIN \leq 1.8500000E+00

BILIRUBIN mean = 1.7438E+00

ClassName	Number	ClassPrior
die	2	0.1250
live	14	0.8750

Number of training cases misclassified = 2

Node 120: Terminal node

ClassName	Number	ClassPrior
die	0	0.0000
live	11	1.0000

Predicted class is live

Number of training cases misclassified = 0

Node 121: Terminal node

ClassName	Number	ClassPrior
die	2	0.4000
live	3	0.6000

Predicted class is die

Number of training cases misclassified = 3

Node 61: Terminal node

ClassName	Number	ClassPrior
die	3	0.6000
live	2	0.4000

Predicted class is die

Number of training cases misclassified = 2

Node 31: Terminal node

ClassName	Number	ClassPrior
die	8	0.5714
live	6	0.4286

Predicted class is die

Number of training cases misclassified = 6

LaTeX code for tree is in file: hep.tex

Classification matrix for training sample:

Predicted	True class	
	die	live
class		
die	31	25
live	1	98
Total	32	123

```
Number of cases used for tree construction = 155
Number misclassified = 26
Resubstitution estimate of mean misclassification cost =
0.187096774193548
```

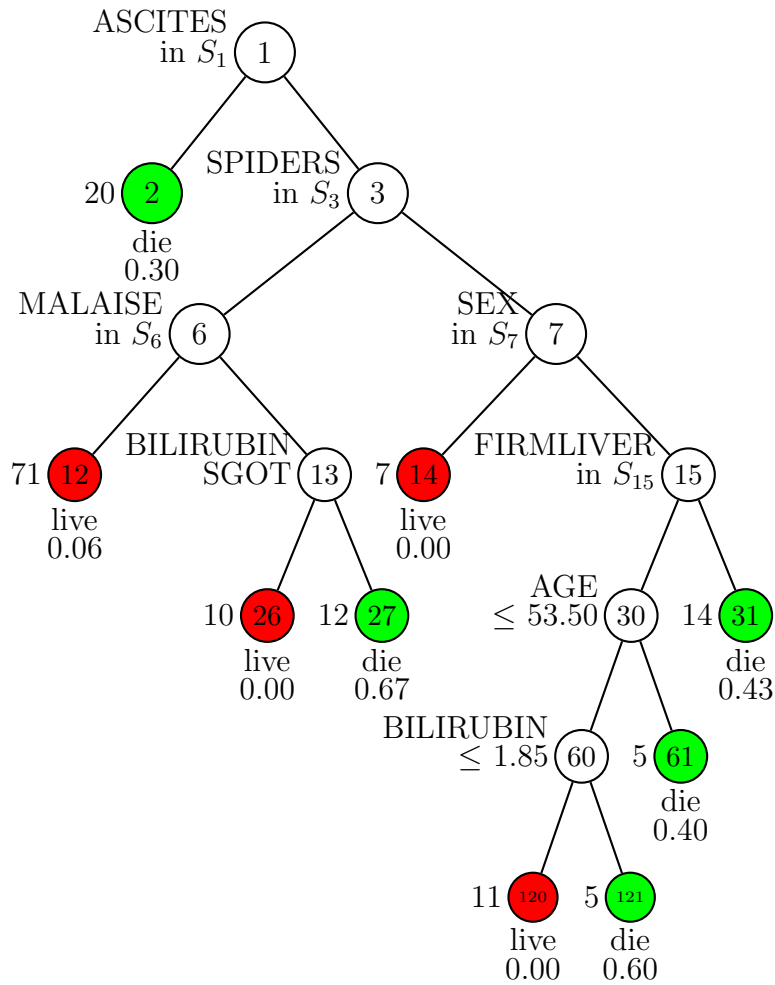
The tree is shown in Figure 2. Terminal nodes are color-coded to indicate the different predicted classes.

7 Regression examples

7.1 Baseball data

We use the baseball dataset `bbdat.txt` to show the results for regression trees when there are no missing values. The contents of the data description file `bbdsc.txt` are:

```
bbdat.txt
NA
column, varname, vartype
1 Id x
2 Name x
3 Bat86 n
4 Hit86 n
5 Hr86 n
6 Run86 n
7 Rb86 n
8 Wlk86 n
9 Yrs n
10 Batcr n
11 Hitcr n
12 Hrcr n
13 Runcr n
14 Rbcr n
15 Wlkcr n
16 Leag86 b
17 Div86 b
18 Team86 c
19 Pos86 b
20 Puto86 n
21 Asst86 n
22 Err86 n
23 Salary x
24 Leag87 b
25 Team87 c
26 Logsalary d
```



Notice that there are four variables having the “b” variable type. This means that 0-1 dummy variables will be created for them in fitting the node linear models.

7.1.1 Stepwise linear regression

Following is a session log for fitting a regression tree model using stepwise linear regression in each node.

```

Choose one of the following options:
1. Read the warranty disclaimer
2. Fit a model
3. Convert data to other formats
4. Create a batch input file
5. Rank and select regressor variables
Input your choice: 2
Input name of file to store results: bbout.txt

Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
You can fit a classification tree or a regression tree
Input 1 for classification, 2 for regression ([1:2], <cr>=1):2
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards, 5=multiresponse,
6=longitudinal data (requires T variables).
Input choice ([1:6], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
0: stepwise linear, 1: multiple linear, 2: best polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <cr>=0):
Option 1 uses all n, f, and b variables as regressors.
Option 2 selects a single n or f variable as regressor in each node.
Option 3 fits a constant to each node, and Option 4 fits a simple ANCOVA
(analysis of covariance) model to each node, using the best linear predictor
among the f and b variables and as many dummy variables as needed.
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
You can fit a classification tree or a regression tree
Input 1 for classification, 2 for regression ([1:2], <cr>=1):2
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards, 5=multiresponse,
6=longitudinal data (requires T variables).
Input choice ([1:6], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
0: stepwise linear, 1: multiple linear, 2: best polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <cr>=0):
Input 1 for forward+backward, 2 for forward, 3 for all subsets ([1:3], <cr>=1):

```

```

Input the maximum number of variables to be selected
0 indicates that the largest possible value is used
Input maximum number of variables to be selected ([0:], <cr>=0):
Input F-to-enter value ([0.01:], <cr>=4.00):
Input F-to-delete value ([0.01:], <cr>=3.99):
Choose larger values to reduce the number of selected regressors.
Choose a truncation method for predicted values:
0: none, 1: node range, 2: +10% node range, 3: global range,
4: 2-sided Winsorization
Input 0, 1, 2, 3, or 4 ([0:4], <cr>=0):
Truncation can reduce mean prediction error. Option 1 truncates the predicted
values to lie within the observed values in the node. Option 2 expands this to
110% of the node range. Option 3 truncates the predicted values to lie within
the range of the whole training sample. Option 4 uses Winsorization, i.e.,
predicted values are constant outside the smallest box containing the training
sample in the node.
Input 1 for interaction tests, 2 to skip them ([1:2], <cr>=1):
Skip to shorten computation time.
Input 1 to prune by CV, 2 for no pruning ([1:2], <cr>=1):

Input name of data description file (max 100 chars; enclose within
quotes if it contains spaces): bbdsc.txt
Reading data description file ...
Training sample file: bbdatt.txt
Missing value code: NA
Dependent variable is Logsalary
Length of longest data entry = 17
Total number of cases =      263
Cat. var. in column   #levels (incl. missing)   #missing values
                   16                   2                   0
                   17                   2                   0
                   18                  24                   0
                   19                  23                   0
                   24                   2                   0
                   25                  24                   0

Checking data ...
The program will try to create the variables in the desc. file.
If it is unsuccessful, please create the columns yourself...
Some b variables are found; GUIDE and will create dummy vectors for them.
Number of dummy variables created:      25
  Total #cases w/ #cases w/
  #cases  miss. D  miss. val  #X-var  #N-var  #F-var  #S-var  #B-var  #C-var
    263      0      0      3      16      0      0      4      2

No weight variable in data file
Number of cases used for training =      263
Default number of cross-validations =      10

```

Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
 Best tree may be chosen based on mean or median CV estimate
 Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
 Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50):
 Choose fraction of cases for splitting
 Larger values give more splits: 0 = median split and 1 = all possible splits
 Default fraction is 0.38023
If 0 is chosen, each node is split at the median value; this option is quickest.
If 1 is chosen, every possible split will be attempted; this is slowest.
 Choose 1 to accept default split fraction, 2 to change it
 Input 1 or 2 ([1:2], <cr>=1):
 Default max number of split levels = 10
 Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
 Smallest possible node sample size = 5
 Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
 Input 2 for LaTeX tree diagrams, 1 to skip them ([1:2], <cr>=1):2
 Input file name to store LaTeX code (use .tex as suffix): bb.tex
 A file by that name already exists
 Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):
 Input 1 for a vertical tree, 2 for a sideways tree ([1:2], <cr>=1):
 Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1):
 Input 1 to number all nodes, 2 to number leaves only ([1:2], <cr>=1):
 Choose a color for the leaf nodes:
 (1) white
 (2) lightgray
 (3) gray
 (4) darkgray
 (5) black
 (6) yellow
 (7) red
 (8) blue
 (9) green
 (10) magenta
 (11) cyan
 Input your choice ([1:11], <cr>=6):
 You can store the variables and/or values used to split and fit in a file
 Choose 1 to skip this step, 2 to store split and fit variables,
 3 to store split variables and their values
 Input your choice ([1:3], <cr>=1):
 Input 2 to save regression coefs in each node in a file, 1 otherwise ([1:2], <cr>=1):
 Input 2 to save predicted values and node IDs, 1 otherwise ([1:2], <cr>=1):2
 Input name of file to store node IDs and predicted values: bbfit.txt
 Input 2 to save terminal node IDs for importance scoring later;
 1 otherwise ([1:2], <cr>=1):
*This option is for setting up a follow-up run for importance scoring; see the
 section on importance ranking below.*

Constructing main tree ...

Number of terminal nodes of largest tree = 30

Performing cross-validation:

Finished cross-validation iteration	1
Finished cross-validation iteration	2
Finished cross-validation iteration	3
Finished cross-validation iteration	4
Finished cross-validation iteration	5
Finished cross-validation iteration	6
Finished cross-validation iteration	7
Finished cross-validation iteration	8
Finished cross-validation iteration	9
Finished cross-validation iteration	10

Pruning main tree. Please wait.

Results of subtree sequence

Trees based on mean with naive SE are marked with * and **

Tree based on mean with bootstrap SE is marked with --

Trees based on median with finite bootstrap SE are marked with + and ++

Subtree	#Terminal nodes
1	29
2	28
3	27
4	26
5	25
6	24
7	21
8	20
9	18
10	15
11	14
12	13
13	12
14	11
15	5
16	4
17**	2
18	1

* tree, ** tree, + tree, and ++ tree all the same

LaTeX code for tree is in file: bb.tex

Observed and predicted values are in file: bbfit.txt

Results are stored in file: bbout.txt

The contents from the file `bbout.txt` follow. They show a tree with two leaf nodes and give the regression coefficients, sample means of the dependent and predictor variables, MSE and R^2 values, and names of the split variables in each node.

```
Least squares regression tree
No truncation of predicted values
Pruning by cross-validation
Data description file: bbdsc.txt
Training sample file: bbdatt.txt
Missing value code: NA
Dependent variable is Logsalary
Piecewise forward and backward stepwise regression
F-to-enter and F-to-delete = 4.000000000000000 3.990000000000000
Using as many variables as needed
Length of longest data entry = 17
Number of dummy variables created = 25
```

Summary information (without x variables)

d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical, n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight
For categorical variables, #categories include one for missing values

Column number	Variable name	Variable type	Minimum value	Maximum value	Number of categories	Number missing
3	Bat86	n	1.2700E+02	6.8700E+02		
4	Hit86	n	3.2000E+01	2.3800E+02		
5	Hr86	n	0.0000E+00	4.0000E+01		
6	Run86	n	1.3000E+01	1.3000E+02		
7	Rb86	n	8.0000E+00	1.2100E+02		
8	Wlk86	n	3.0000E+00	1.0500E+02		
9	Yrs	n	1.0000E+00	2.4000E+01		
10	Batcr	n	1.8100E+02	1.4053E+04		
11	Hitcr	n	4.2000E+01	4.2560E+03		
12	Hrcr	n	0.0000E+00	5.4800E+02		
13	Runcr	n	1.8000E+01	2.1650E+03		
14	Rbcr	n	9.0000E+00	1.6590E+03		
15	Wlkcr	n	8.0000E+00	1.5660E+03		
16	Leag86	b			2	
17	Div86	b			2	
18	Team86	c			24	
19	Pos86	b			23	
20	Puto86	n	0.0000E+00	1.3770E+03		
21	Asst86	n	0.0000E+00	4.9200E+02		
22	Err86	n	0.0000E+00	3.2000E+01		
24	Leag87	b			2	
25	Team87	c			24	
26	Logsalary	d	4.2121E+00	7.8079E+00		

*Dummy variables constructed from the b categorical variables are given next.
For each variable, the first level in alphabetical order is set to zero.*

27	Leag8=N	f	0.0000E+00	1.0000E+00
28	Div86=W	f	0.0000E+00	1.0000E+00
29	Pos86=10	f	0.0000E+00	1.0000E+00
30	Pos86=23	f	0.0000E+00	1.0000E+00
31	Pos86=2B	f	0.0000E+00	1.0000E+00
32	Pos86=2S	f	0.0000E+00	1.0000E+00
33	Pos86=32	f	0.0000E+00	1.0000E+00
34	Pos86=3B	f	0.0000E+00	1.0000E+00
35	Pos86=30	f	0.0000E+00	1.0000E+00
36	Pos86=3S	f	0.0000E+00	1.0000E+00
37	Pos86=C	f	0.0000E+00	1.0000E+00
38	Pos86=CD	f	0.0000E+00	1.0000E+00
39	Pos86=CF	f	0.0000E+00	1.0000E+00
40	Pos86=DH	f	0.0000E+00	1.0000E+00
41	Pos86=D0	f	0.0000E+00	1.0000E+00
42	Pos86=LF	f	0.0000E+00	1.0000E+00
43	Pos86=01	f	0.0000E+00	1.0000E+00
44	Pos86=0D	f	0.0000E+00	1.0000E+00
45	Pos86=0F	f	0.0000E+00	1.0000E+00
46	Pos86=0S	f	0.0000E+00	1.0000E+00
47	Pos86=RF	f	0.0000E+00	1.0000E+00
48	Pos86=S3	f	0.0000E+00	1.0000E+00
49	Pos86=SS	f	0.0000E+00	1.0000E+00
50	Pos86=UT	f	0.0000E+00	1.0000E+00
51	Leag8=N	f	0.0000E+00	1.0000E+00

Total	#cases w/	#cases w/							
#cases	miss. D	miss. val	#X-var	#N-var	#F-var	#S-var	#B-var	#C-var	
263	0	0	3	16	0	0	4	2	

No weight variable in data file

Number of cases used for training = 263

Interaction tests on all variables

Pruning by v-fold cross-validation, with v = 10

Selected tree is based on mean of CV estimates

Fraction of cases used for splitting each node = 0.38023

Max number of split levels = 10

Minimum node size = 5

Number of SE's for pruned tree = 5.0000E-01

Size and CV MSE and SE of subtrees:

Tree	#Tnodes	Mean MSE	SE(Mean)	BSE(Mean)	Median MSE	BSE(Median)
1	29	2.804E-01	2.720E-02	2.430E-02	2.771E-01	2.005E-02
2	28	2.804E-01	2.720E-02	2.430E-02	2.771E-01	2.005E-02

3	27	2.804E-01	2.720E-02	2.430E-02	2.771E-01	2.005E-02
4	26	2.804E-01	2.720E-02	2.430E-02	2.771E-01	2.005E-02
5	25	2.951E-01	3.337E-02	3.747E-02	2.740E-01	3.350E-02
6	24	2.917E-01	3.306E-02	3.714E-02	2.722E-01	3.625E-02
7	21	2.539E-01	2.630E-02	2.433E-02	2.514E-01	5.287E-02
8	20	2.451E-01	2.627E-02	2.483E-02	2.345E-01	4.815E-02
9	18	2.405E-01	2.663E-02	2.669E-02	2.295E-01	4.929E-02
10	15	2.308E-01	2.538E-02	2.458E-02	2.585E-01	4.389E-02
11	14	2.165E-01	2.329E-02	2.288E-02	2.385E-01	4.467E-02
12	13	2.168E-01	2.483E-02	2.602E-02	2.446E-01	4.591E-02
13	12	2.000E-01	2.322E-02	2.304E-02	1.752E-01	3.981E-02
14	11	1.775E-01	2.176E-02	2.133E-02	1.706E-01	3.404E-02
15	5	1.775E-01	2.176E-02	2.133E-02	1.706E-01	3.404E-02
16	4	1.777E-01	2.098E-02	2.134E-02	1.716E-01	3.431E-02
17**	2	1.211E-01	1.441E-02	1.368E-02	1.166E-01	1.876E-02
18	1	3.478E-01	2.590E-02	2.268E-02	3.456E-01	3.879E-02

Column 3 gives the CV estimates of mean squared error.

Column 4 gives their standard errors (called naive estimate of SE).

0-SE tree based on mean is marked with *

0-SE tree based on median with bootstrap SE is marked with +

Selected-SE tree based on mean using naive SE is marked with **

Selected-SE tree based on mean using bootstrap SE is marked with --

Selected-SE tree based on median and bootstrap SE is marked with ++

* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate.

Structure of final tree. Each terminal node is marked with a T.

D-mean is mean of Logsalary in the node

Cases fit give the number of cases used to fit node

MSE and R² are based on all cases in node

Node label	Total cases	Cases fit	Matrix rank	Node D-mean	Node MSE	Node R ²	Split variable	Interacting variable
1	263	263	9	5.945E+00	2.808E-01	0.6391	Yrs	
2T	143	143	7	5.506E+00	7.928E-02	0.8907	Yrs	
3T	120	120	6	6.469E+00	1.195E-01	0.6456	Bat86	

The values in the 3rd column are the number of cases used in the regression models. If there are missing values, the 3rd column values may be less than those in the 2nd column. The 4th column gives the number of parameters fitted in each node. The second last column contains the names of the variables selected to split the nodes. If a split is due to an interaction effect between two variables, the names of the two variables will appear in the last and second last columns. There are no such splits in this example.

Number of terminal nodes of final tree: 2
 Total number of nodes of final tree: 3

Regression tree:

Node 1: Yrs <= 6.50000
 Node 2: Logsalary-mean = 5.50632
 Node 1: Yrs > 6.50000
 Node 3: Logsalary-mean = 6.46866

The next paragraphs give the estimated regression coefficients, t-statistics, and the minimum, mean, and maximum values of the selected regressor variables in the leaf nodes. At each intermediate node, cases with missing values in the split variable are sent to the left branch. Cases with missing values in any linear predictor in a terminal node are predicted by the node mean.

Node 1: Intermediate node
 A case goes into Node 2 if Yrs <= 6.500000E+00
 Yrs mean = 7.3802E+00

 Node 2: Terminal node
 Coefficients of least squares regression function:

Regressor	Coefficient	t-stat	p-val	Min	Mean	Max
Constant	4.1385E+00	39.36	0.0000			
Bat86	-1.8387E-03	-4.25	0.0000	1.5100E+02	4.1345E+02	6.8700E+02
Run86	1.5359E-02	6.46	0.0000	1.3000E+01	5.6699E+01	1.1900E+02
Yrs	1.1659E-01	4.86	0.0000	1.0000E+00	3.8042E+00	6.0000E+00
Batcr	5.0610E-04	5.44	0.0000	1.8100E+02	1.2046E+03	3.3740E+03
Rbcr	1.5954E-03	2.86	0.0049	9.0000E+00	1.4315E+02	4.7500E+02
Pos86=CF	-2.3212E-01	-2.74	0.0070	0.0000E+00	1.0490E-01	1.0000E+00

Predicted mean if regression function is inapplicable due to missing values =
 5.50631560540131

 Node 3: Terminal node
 Coefficients of least squares regression function:

Regressor	Coefficient	t-stat	p-val	Min	Mean	Max
Constant	6.2274E+00	33.81	0.0000			
Hit86	4.7082E-03	4.73	0.0000	3.2000E+01	1.0759E+02	2.0000E+02
Yrs	-1.0408E-01	-6.16	0.0000	7.0000E+00	1.1642E+01	2.4000E+01
Runcr	7.9765E-04	3.26	0.0015	6.7000E+01	6.1369E+02	2.1650E+03
Rbcr	5.9958E-04	2.56	0.0118	8.2000E+01	5.6998E+02	1.6590E+03
Puto86	4.1481E-04	3.43	0.0008	0.0000E+00	2.7723E+02	1.3140E+03

Predicted mean if regression function is inapplicable due to missing values =
 6.46866476618588

LaTeX code for tree is in file: `bb.tex`

Observed and predicted values are in file: `bbfit.txt`

Proportion of variance (R-squared) explained by tree model = 0.8745

The \LaTeX drawing produced by the file `bb.tex` is shown in Figure 3.

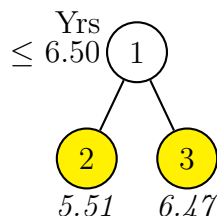


Figure 3: GUIDE piecewise linear least-squares regression tree model with stepwise variable selection. At each intermediate node, a case goes to the left child node if and only if the condition is satisfied. Number in italics beneath a leaf is the sample mean of `Logsalary`.

Following are the first few lines of the file `bbfit.txt` that was produced by GUIDE. In the first column, the symbol `y` refers to “yes” and the symbol `n` to “no”. A `y` indicates that the case is used to train the model. Since this dataset has no weight variable nor missing values, every row has a `y` in the 1st column. The leaf node number is given in the 2nd column. This facilitates the extraction of the observations in one or more nodes for further scrutiny. The observed and predicted values of the dependent variable are given in the last two columns.

train	node	observed	predicted
y	3	6.163315E+00	5.918191E+00
y	2	6.173786E+00	5.867504E+00
y	3	6.214608E+00	6.992343E+00
y	2	4.516339E+00	4.654253E+00
y	3	6.620073E+00	6.596362E+00
y	2	4.248495E+00	4.507433E+00
y	2	4.605170E+00	4.625577E+00

7.1.2 Piecewise best simple linear regression

For some purposes, it is useful to be able to visualize the fitted regression function and the data simultaneously. This can be accomplished by fitting a piecewise simple

linear model, where the best single regressor is selected to fit a straight line in each node. The following interactive dialog shows how this is done.

```

Choose one of the following options:
1. Read the warranty disclaimer
2. Fit a model
3. Convert data to other formats
4. Create a batch input file
5. Rank and select regressor variables
Input your choice: 2
Input name of file to store results: linout.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
You can fit a classification tree or a regression tree
Input 1 for classification, 2 for regression ([1:2], <cr>=1):2
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards, 5=multiresponse,
6=longitudinal data (requires T variables).
Input choice ([1:6], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
0: stepwise linear, 1: multiple linear, 2: best polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <cr>=0):2
Input degree of polynomial ([1:9], <cr>=1):
Choose 1 to use alpha-level to drop insignificant powers, 2 otherwise ([1:2], <cr>=1):
Input significance level ([0.00:1.00], <cr>=0.05):
A lower-order polynomial is fitted if the higher-order terms are not significant
at this level. Categorical variables are only allowed to split the nodes.
Length of longest data entry = 17
Total number of cases =      263
Cat. var. in column   #levels (incl. missing)   #missing values
                16                2                0
                17                2                0
                18               24                0
                19               23                0
                24                2                0
                25               24                0

Checking data ...
  Total #cases w/ #cases w/
#cases  miss. D  miss. val  #X-var  #N-var  #F-var  #S-var  #B-var  #C-var
   263      0      0      3      16      0      0      4      2

No weight variable in data file
Number of cases used for training =      263
Default number of cross-validations =      10
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Best tree may be chosen based on mean or median CV estimate
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):

```

```

Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50):
Choose fraction of cases for splitting
Larger values give more splits: 0 = median split and 1 = all possible splits
Default fraction is 0.38023
Choose 1 to accept default split fraction, 2 to change it
Input 1 or 2 ([1:2], <cr>=1):
Default max number of split levels =          10
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Smallest possible node sample size =          6
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Input 2 for LaTeX tree diagrams, 1 to skip them ([1:2], <cr>=1):2
Input file name to store LaTeX code (use .tex as suffix): lin.tex
A file by that name already exists
Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):
Input 1 for a vertical tree, 2 for a sideways tree ([1:2], <cr>=1):
Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1):
Input 1 to number all nodes, 2 to number leaves only ([1:2], <cr>=1):
Choose a color for the leaf nodes:
(1) white
(2) lightgray
(3) gray
(4) darkgray
(5) black
(6) yellow
(7) red
(8) blue
(9) green
(10) magenta
(11) cyan
Input your choice ([1:11], <cr>=6):
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save regression coefs in each node in a file, 1 otherwise ([1:2], <cr>=1):
Input 2 to save predicted values and node IDs, 1 otherwise ([1:2], <cr>=1):
Input 2 to save terminal node IDs for importance scoring later; 1 otherwise ([1:2], <cr>=1):
This option is for setting up a follow-up run for importance scoring; see the
section on importance ranking below.
Constructing main tree ...
Number of terminal nodes of largest tree =          30
Performing cross-validation:
Finished cross-validation iteration          1
Finished cross-validation iteration          2
Finished cross-validation iteration          3
Finished cross-validation iteration          4

```

```

Finished cross-validation iteration      5
Finished cross-validation iteration      6
Finished cross-validation iteration      7
Finished cross-validation iteration      8
Finished cross-validation iteration      9
Finished cross-validation iteration     10

```

Pruning main tree. Please wait.

Results of subtree sequence

Trees based on mean with naive SE are marked with * and **

Tree based on mean with bootstrap SE is marked with --

Trees based on median with finite bootstrap SE are marked with + and ++

Subtree	#Terminal nodes
1	29
2	28
3	27
4	26
5	25
6	24
7	23
8	20
9	19
10	18
11	17
12	14
13	13
14	11
15	10
16	8
17	6
18**	5
19	3
20	2
21	1

* tree, ** tree, + tree, and ++ tree all the same

LaTeX code for tree is in file: lin.tex

Results are stored in file: linout.txt

Contents of linout.txt

Powers are dropped if they are not significant at level 0.0500

Least squares regression tree

No truncation of predicted values

Pruning by cross-validation

Data description file: bbdsc.txt
 Training sample file: bbdatt.txt
 Missing value code: NA
 Dependent variable is Logsalary
 Warning: B variables changed to C
 Piecewise simple linear or constant model
 Length of longest data entry = 17

Summary information (without x variables)

d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical,
 n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight
 For categorical variables, #categories include one for missing values

Column number	Variable name	Variable type	Minimum value	Maximum value	Number of categories	Number missing
3	Bat86	n	1.2700E+02	6.8700E+02		
4	Hit86	n	3.2000E+01	2.3800E+02		
5	Hr86	n	0.0000E+00	4.0000E+01		
6	Run86	n	1.3000E+01	1.3000E+02		
7	Rb86	n	8.0000E+00	1.2100E+02		
8	Wlk86	n	3.0000E+00	1.0500E+02		
9	Yrs	n	1.0000E+00	2.4000E+01		
10	Batcr	n	1.8100E+02	1.4053E+04		
11	Hitcr	n	4.2000E+01	4.2560E+03		
12	Hrcr	n	0.0000E+00	5.4800E+02		
13	Runcr	n	1.8000E+01	2.1650E+03		
14	Rbcr	n	9.0000E+00	1.6590E+03		
15	Wlkcr	n	8.0000E+00	1.5660E+03		
16	Leag86	c			2	
17	Div86	c			2	
18	Team86	c			24	
19	Pos86	c			23	
20	Puto86	n	0.0000E+00	1.3770E+03		
21	Asst86	n	0.0000E+00	4.9200E+02		
22	Err86	n	0.0000E+00	3.2000E+01		
24	Leag87	c			2	
25	Team87	c			24	
26	Logsalary	d	4.2121E+00	7.8079E+00		

Total #cases	#cases w/ miss. D	#cases w/ miss. val	#X-var	#N-var	#F-var	#S-var	#B-var	#C-var
263	0	0	3	16	0	0	4	2

No weight variable in data file
 Number of cases used for training = 263

Interaction tests on all variables
 Pruning by v-fold cross-validation, with v = 10

Selected tree is based on mean of CV estimates
 Fraction of cases used for splitting each node = 0.38023
 Max number of split levels = 10
 Minimum node size = 6
 Number of SE's for pruned tree = 5.0000E-01

Size and CV MSE and SE of subtrees:

Tree	#Tnodes	Mean MSE	SE(Mean)	BSE(Mean)	Median MSE	BSE(Median)
1	29	2.487E-01	2.863E-02	2.849E-02	2.308E-01	3.133E-02
2	28	2.450E-01	2.827E-02	2.823E-02	2.308E-01	2.894E-02
3	27	2.439E-01	2.829E-02	2.762E-02	2.308E-01	2.836E-02
4	26	2.439E-01	2.829E-02	2.762E-02	2.308E-01	2.836E-02
5	25	2.439E-01	2.829E-02	2.762E-02	2.308E-01	2.836E-02
6	24	2.439E-01	2.829E-02	2.762E-02	2.308E-01	2.836E-02
7	23	2.435E-01	2.846E-02	2.751E-02	2.318E-01	2.827E-02
8	20	2.338E-01	2.784E-02	2.405E-02	2.293E-01	2.325E-02
9	19	2.255E-01	2.698E-02	2.561E-02	2.200E-01	3.142E-02
10	18	2.218E-01	2.680E-02	2.563E-02	2.045E-01	3.289E-02
11	17	2.213E-01	2.679E-02	2.579E-02	2.045E-01	3.349E-02
12	14	2.072E-01	2.228E-02	2.540E-02	1.982E-01	2.579E-02
13	13	1.889E-01	2.146E-02	2.635E-02	1.624E-01	1.975E-02
14	11	1.710E-01	1.900E-02	1.757E-02	1.481E-01	1.791E-02
15	10	1.628E-01	1.788E-02	1.669E-02	1.514E-01	1.323E-02
16	8	1.581E-01	1.742E-02	1.289E-02	1.514E-01	1.271E-02
17	6	1.521E-01	1.664E-02	1.316E-02	1.492E-01	1.290E-02
18**	5	1.475E-01	1.666E-02	1.427E-02	1.305E-01	2.393E-02
19	3	1.594E-01	1.766E-02	1.686E-02	1.602E-01	3.387E-02
20	2	1.744E-01	1.754E-02	1.664E-02	1.733E-01	3.672E-02
21	1	4.675E-01	4.141E-02	4.128E-02	4.668E-01	5.621E-02

0-SE tree based on mean is marked with *

0-SE tree based on median with bootstrap SE is marked with +

Selected-SE tree based on mean using naive SE is marked with **

Selected-SE tree based on mean using bootstrap SE is marked with --

Selected-SE tree based on median and bootstrap SE is marked with ++

* tree, ** tree, + tree, and ++ tree all the same

Following tree is based on mean CV with naive SE estimate.

Structure of final tree. Each terminal node is marked with a T.

D-mean is mean of Logsalary in the node

Cases fit give the number of cases used to fit node

MSE and R² are based on all cases in node

Node label	Total cases	Cases Matrix fit rank	Node D-mean	Node MSE	Node R ²	Split variable	Interact. variable	Fit variable
---------------	----------------	--------------------------	----------------	-------------	------------------------	-------------------	-----------------------	-----------------

1	263	263	2	5.945E+00	4.468E-01	0.4257	Yrs	+Hitcr
2	143	143	2	5.506E+00	1.266E-01	0.8254	Hitcr	+Batcr
4T	110	110	2	5.146E+00	9.042E-02	0.7371	Wlkr	+Hitcr
5T	33	33	2	6.706E+00	7.046E-02	0.4395	Wlkr86	+Rbcr
3	120	120	2	6.469E+00	1.911E-01	0.4331	Wlkr	+Hit86
6T	78	78	2	6.338E+00	1.344E-01	0.4658	Rb86	+Bat86
7	42	42	2	6.711E+00	2.169E-01	0.4654	Hrcr	+Hit86
14T	21	21	2	6.541E+00	1.371E-01	0.5709	Puto86	+Bat86
15T	21	21	2	6.880E+00	1.222E-01	0.7189	Rbcr	-Yrs

Number of terminal nodes of final tree: 5

Total number of nodes of final tree: 9

Regression tree:

```

Node 1: Yrs <= 6.50000
  Node 2: Hitcr <= 4.59500E+02
    Node 4: Logsalary-mean = 5.14642
    Node 2: Hitcr > 4.59500E+02
      Node 5: Logsalary-mean = 6.70595
  Node 1: Yrs > 6.50000
    Node 3: Wlkr <= 5.27000E+02
      Node 6: Logsalary-mean = 6.33838
    Node 3: Wlkr > 5.27000E+02
      Node 7: Hrcr <= 1.85500E+02
        Node 14: Logsalary-mean = 6.54147
        Node 7: Hrcr > 1.85500E+02
          Node 15: Logsalary-mean = 6.87978

```

Node 1: Intermediate node

A case goes into Node 2 if Yrs <= 6.500000E+00
 Yrs mean = 7.3802E+00

Node 2: Intermediate node

A case goes into Node 4 if Hitcr <= 4.595000E+02
 Hitcr mean = 3.2022E+02

Node 4: Terminal node

Coefficients of least squares regression function:

Regressor	Coefficient	t-stat	p-val	Min	Mean	Max
Constant	4.2487E+00	71.82	0.0000			
Hitcr	4.1935E-03	17.40	0.0000	4.2000E+01	2.1408E+02	4.5700E+02

Predicted mean if regression function is inapplicable due to missing values =
 5.14642453296224

```

-----
Node 5: Terminal node
Coefficients of least squares regression function:
Regressor    Coefficient    t-stat  p-val    Min        Mean        Max
Constant     5.9484E+00    36.98  0.0000
Rbcr         2.5219E-03    4.93  0.0000   1.0300E+02  3.0039E+02  4.7500E+02
Predicted mean if regression function is inapplicable due to missing values =
  6.70595251353155
-----
Node 3: Intermediate node
A case goes into Node 6 if Wlkr <=  5.2700000E+02
      Wlkr mean =  4.5207E+02
-----
Node 6: Terminal node
Coefficients of least squares regression function:
Regressor    Coefficient    t-stat  p-val    Min        Mean        Max
Constant     5.3482E+00    41.56  0.0000
Bat86        2.5225E-03    8.14  0.0000   1.2700E+02  3.9253E+02  6.3700E+02
Predicted mean if regression function is inapplicable due to missing values =
  6.33837853694061
-----
Node 7: Intermediate node
A case goes into Node 14 if Hrcr <=  1.8550000E+02
      Hrcr mean =  2.0974E+02
-----
Node 14: Terminal node
Coefficients of least squares regression function:
Regressor    Coefficient    t-stat  p-val    Min        Mean        Max
Constant     5.3107E+00    20.49  0.0000
Bat86        3.0220E-03    5.03  0.0001   1.5500E+02  4.0729E+02  6.0800E+02
Predicted mean if regression function is inapplicable due to missing values =
  6.54147427723514
-----
Node 15: Terminal node
Coefficients of least squares regression function:
Regressor    Coefficient    t-stat  p-val    Min        Mean        Max
Constant     9.9871E+00    22.05  0.0000
Yrs          -2.0392E-01   -6.97  0.0000   1.0000E+01  1.5238E+01  2.0000E+01
Predicted mean if regression function is inapplicable due to missing values =
  6.87977553519045
-----

```

LaTeX code for tree is in file: lin.tex

Proportion of variance (R-squared) explained by tree model = 0.8621

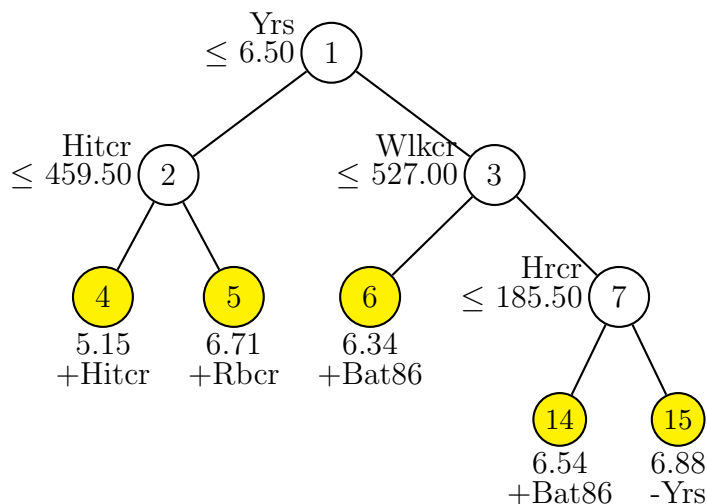


Figure 4: GUIDE piecewise simple linear least-squares regression tree model. At each intermediate node, a case goes to the left branch if and only if the condition is satisfied. Beneath each leaf node are sample mean of Logsalary and sign and name of the regressor.

The tree structure is shown in Figure 4.

7.2 Tuition data with missing values

If a data set contains missing values and a piecewise-linear model is desired, only the observations with non-missing regressors are used to estimate the linear model in each node. Observations with missing regressor values in the node are fitted with a constant equal to their own mean at the node. We demonstrate this with the data set `tuitiondat.txt`, which gives information on out-of-state tuition and other variables for U.S. colleges. The data description file `tuitiondsc.txt` is:

```
tuitiondat.dat
NA
col_num var_name var_type
1 FICE x
2 CollName x
3 State x
4 PubPriv b
5 MathSAT x
6 VerbsAT x
7 CombSAT n
8 ACT x
```

9 Q1MSAT x
10 Q3MSAT x
11 Q1VSAT x
12 Q3VSAT x
13 Q1ACT x
14 Q3ACT x
15 AppsRec n
16 AppsAcc n
17 NewEnrol n
18 Top10 n
19 Top25 n
20 FUgrad n
21 PUgrad x
22 InTuition x
23 OutTuition d
24 RnBcost n
25 RmCost x
26 BrdCost x
27 AddFees x
28 BookCost x
29 PerSpend x
30 PFacPhD n
31 PFacTerm x
32 StudFac n
33 PAIDonate x
34 InstExp n
35 GradRate n
36 Type c
37 FullPSal n
38 AssocPSal x
39 AsstPSal x
40 AveSal x
41 FullPComp x
42 AssocPComp x
43 AsstPComp x
44 AveComp x
45 NFullProf n
46 NAssocProf x
47 NAsstProf x
48 NInstr x
49 NAllFac x

7.2.1 Best simple ANCOVA

This section shows how to use GUIDE to fit a piecewise best simple ANCOVA (analysis of covariance) model. At each node, the best linear predictor is chosen and used to fit an ANCOVA model that includes the dummy variables from all “b” variables. Here is the session log.

```

Choose one of the following options:
1. Read the warranty disclaimer
2. Fit a model
3. Convert data to other formats
4. Create a batch input file
5. Rank and select regressor variables
Input your choice: 2
Input name of file to store results: ancova.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
You can fit a classification tree or a regression tree
Input 1 for classification, 2 for regression ([1:2], <cr>=1):2
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards, 5=multiresponse,
6=longitudinal data (requires T variables).
Input choice ([1:6], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
0: stepwise linear, 1: multiple linear, 2: best polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <cr>=0):4
Input the maximum number of variables to be selected
0 indicates that the largest possible value is used
Input maximum number of variables to be selected ([0:], <cr>=0):
Input F-to-enter value ([0.01:], <cr>=4.00):
Input F-to-delete value ([0.01:], <cr>=3.99):
Choose a truncation method for predicted values:
0: none, 1: node range, 2: +10% node range, 3: global range,
4: 2-sided Winsorization, 5: 1-sided Winsorization
Input 0, 1, 2, 3, 4, or 5 ([0:5], <cr>=0):
Input 1 for interaction tests, 2 to skip them ([1:2], <cr>=1):
Input 1 to prune by CV, 2 for no pruning ([1:2], <cr>=1):

Input name of data description file (max 100 chars; enclose within
quotes if it contains spaces): tuitiondsc.txt
Reading data description file ...
Training sample file: tuitiondat.txt
Missing value code: NA
Dependent variable is OutTuition
Length of longest data entry = 20
Total number of cases =      1134

```

```

Cat. var. in column      #levels (incl. missing)    #missing values
                        4                      2                      0
                        36                     3                      0

Checking data ...
The program will try to create the variables in the desc. file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created:                1
  Total #cases w/ #cases w/
#cases  miss. D  miss. val  #X-var  #N-var  #F-var  #S-var  #B-var  #C-var
  1134      13      621      32      14      0        0        1        1

No weight variable in data file
Number of cases used for training =                1121
Number of cases excluded due to zero weight or missing D-values =                13
Default number of cross-validations =              10
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Best tree may be chosen based on mean or median CV estimate
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50):
Choose fraction of cases for splitting
Larger values give more splits: 0 = median split and 1 = all possible splits
Default fraction is 0.19493
Choose 1 to accept default split fraction, 2 to change it
Input 1 or 2 ([1:2], <cr>=1):
Default max number of split levels =                10
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Smallest possible node sample size =                10
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Input 2 for LaTeX tree diagrams, 1 to skip them ([1:2], <cr>=1):2
Input file name to store LaTeX code (use .tex as suffix): ancova.tex
Input 1 for a vertical tree, 2 for a sideways tree ([1:2], <cr>=1):
Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1):
Input 1 to number all nodes, 2 to number leaves only ([1:2], <cr>=1):
Choose a color for the leaf nodes:
(1) white
(2) lightgray
(3) gray
(4) darkgray
(5) black
(6) yellow
(7) red
(8) blue
(9) green
(10) magenta
(11) cyan
Input your choice ([1:11], <cr>=6):
You can store the variables and/or values used to split and fit in a file

```

```

Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save regression coefs in each node in a file, 1 otherwise ([1:2], <cr>=1):
Input 2 to save predicted values and node IDs, 1 otherwise ([1:2], <cr>=1):
Input 2 to save terminal node IDs for importance scoring later; 1 otherwise ([1:2], <cr>=1):
Constructing main tree ...
Number of terminal nodes of largest tree =          71
Performing cross-validation:
Finished cross-validation iteration          1
Finished cross-validation iteration          2
Finished cross-validation iteration          3
Finished cross-validation iteration          4
Finished cross-validation iteration          5
Finished cross-validation iteration          6
Finished cross-validation iteration          7
Finished cross-validation iteration          8
Finished cross-validation iteration          9
Finished cross-validation iteration         10

```

Pruning main tree. Please wait.

Results of subtree sequence

Trees based on mean with naive SE are marked with * and **

Tree based on mean with bootstrap SE is marked with --

Trees based on median with finite bootstrap SE are marked with + and ++

Subtree	#Terminal nodes
1	70
2	68
3	67
4	65
5	64
6	63
7	60
8	59
9	57
10	56
11	55
12	54
13	51
14	50
15	48
16	47
17	46
18	45
19	43
20	42

21	40
22	35
23	32
24	30
25	29
26	28
27	26
28	25
29	22
30	18
31	17
32	16
33	15
34	14
35+	11
36	9
37++	7
38**	5
39	4
40	3
41	2
42	1

++ tree same as -- tree

* tree same as ++ tree

* tree same as -- tree

LaTeX code for tree is in file: ancova.tex

Results are stored in file: ancova.txt

Contents of ancova.txt

```
Least squares regression tree
No truncation of predicted values
Pruning by cross-validation
Data description file: tuitiondsc.txt
Training sample file: tuitiondat.txt
Missing value code: NA
Dependent variable is OutTuition
Piecewise simple linear ANCOVA model
F-to-enter and F-to-delete = 4.000 3.990
Length of longest data entry = 20
Number of dummy variables created = 1
```

Summary information (without x variables)

d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical,

n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight
 For categorical variables, #categories include one for missing values

Column number	Variable name	Variable type	Minimum value	Maximum value	Number of categories	Number missing
4	PubPriv	b			2	
7	CombSAT	n	6.0000E+02	1.4100E+03		471
15	AppsRec	n	5.7000E+01	4.8094E+04		9
16	AppsAcc	n	4.4000E+01	2.6330E+04		9
17	NewEnrol	n	2.1000E+01	7.4250E+03		5
18	Top10	n	1.0000E+00	9.8000E+01		183
19	Top25	n	1.1000E+01	1.0000E+02		155
20	FUgrad	n	1.1800E+02	3.1643E+04		3
23	OutTuition	d	1.0440E+03	2.5750E+04		13
24	RnBcost	n	1.3060E+03	8.7000E+03		57
30	PFacPhD	n	8.0000E+00	1.0500E+02		29
32	StudFac	n	2.5000E+00	4.2600E+01		2
34	InstExp	n	1.8340E+03	6.2469E+04		24
35	GradRate	n	8.0000E+00	1.1800E+02		69
36	Type	c			3	
37	FullPSal	n	2.7000E+02	1.0090E+03		61
45	NFullProf	n	0.0000E+00	9.9700E+02		
===== Constructed variables =====						
50	PubPr=Publ	f	0.0000E+00	1.0000E+00		

Total #cases	#cases w/ miss.	#cases w/ D miss.	#X-var	#N-var	#F-var	#S-var	#B-var	#C-var
1134	13	621	32	14	0	0	1	1

No weight variable in data file

Number of cases used for training = 1121

Number of cases excluded due to zero weight or missing D-values = 13

Interaction tests on all variables

Pruning by v-fold cross-validation, with v = 10

Selected tree is based on mean of CV estimates

Fraction of cases used for splitting each node = 0.19493

Max number of split levels = 10

Minimum node size = 10

Number of SE's for pruned tree = 5.0000E-01

Size and CV MSE and SE of subtrees:

Tree	#Tnodes	Mean MSE	SE(Mean)	BSE(Mean)	Median MSE	BSE(Median)
1	70	5.138E+06	3.501E+05	4.303E+05	4.426E+06	5.680E+05
2	68	5.138E+06	3.501E+05	4.303E+05	4.426E+06	5.680E+05
3	67	5.138E+06	3.501E+05	4.303E+05	4.426E+06	5.680E+05
4	65	5.138E+06	3.501E+05	4.303E+05	4.426E+06	5.680E+05
5	64	5.138E+06	3.501E+05	4.303E+05	4.426E+06	5.680E+05

6	63	5.138E+06	3.501E+05	4.303E+05	4.426E+06	5.680E+05
7	60	5.138E+06	3.501E+05	4.303E+05	4.426E+06	5.680E+05
8	59	5.138E+06	3.501E+05	4.303E+05	4.426E+06	5.680E+05
9	57	5.138E+06	3.501E+05	4.303E+05	4.426E+06	5.680E+05
10	56	5.138E+06	3.501E+05	4.303E+05	4.426E+06	5.680E+05
11	55	5.138E+06	3.501E+05	4.303E+05	4.426E+06	5.680E+05
12	54	5.138E+06	3.501E+05	4.303E+05	4.426E+06	5.680E+05
13	51	5.038E+06	3.473E+05	4.601E+05	4.559E+06	6.542E+05
14	50	5.038E+06	3.473E+05	4.601E+05	4.559E+06	6.542E+05
15	48	5.038E+06	3.473E+05	4.601E+05	4.559E+06	6.542E+05
16	47	5.044E+06	3.472E+05	4.601E+05	4.588E+06	6.554E+05
17	46	5.045E+06	3.472E+05	4.596E+05	4.588E+06	6.554E+05
18	45	5.045E+06	3.472E+05	4.596E+05	4.588E+06	6.554E+05
19	43	5.025E+06	3.473E+05	4.648E+05	4.577E+06	6.411E+05
20	42	5.025E+06	3.473E+05	4.649E+05	4.574E+06	6.415E+05
21	40	5.053E+06	3.476E+05	4.619E+05	4.675E+06	6.257E+05
22	35	5.050E+06	3.477E+05	4.619E+05	4.675E+06	6.370E+05
23	32	5.047E+06	3.477E+05	4.630E+05	4.671E+06	6.370E+05
24	30	5.042E+06	3.476E+05	4.630E+05	4.644E+06	6.365E+05
25	29	5.019E+06	3.469E+05	4.639E+05	4.533E+06	6.449E+05
26	28	5.015E+06	3.468E+05	4.630E+05	4.533E+06	6.378E+05
27	26	5.014E+06	3.468E+05	4.627E+05	4.533E+06	6.352E+05
28	25	5.016E+06	3.468E+05	4.642E+05	4.533E+06	6.353E+05
29	22	4.929E+06	3.413E+05	4.752E+05	4.254E+06	7.310E+05
30	18	4.988E+06	3.456E+05	4.577E+05	4.254E+06	7.202E+05
31	17	4.851E+06	3.283E+05	4.409E+05	4.319E+06	4.928E+05
32	16	4.774E+06	3.181E+05	3.966E+05	4.365E+06	4.875E+05
33	15	4.696E+06	3.141E+05	3.647E+05	4.309E+06	4.603E+05
34	14	4.591E+06	3.097E+05	3.233E+05	4.349E+06	4.598E+05
35+	11	4.560E+06	3.060E+05	3.444E+05	4.176E+06	6.198E+05
36	9	4.555E+06	2.910E+05	3.103E+05	4.336E+06	5.102E+05
37++	7	4.408E+06	2.519E+05	2.310E+05	4.294E+06	4.420E+05
38**	5	4.531E+06	2.593E+05	2.258E+05	4.495E+06	4.033E+05
39	4	5.001E+06	2.906E+05	2.531E+05	5.190E+06	3.005E+05
40	3	5.254E+06	3.061E+05	2.210E+05	5.484E+06	2.822E+05
41	2	5.828E+06	3.452E+05	4.175E+05	5.724E+06	6.024E+05
42	1	1.109E+07	6.495E+05	4.561E+05	1.062E+07	6.688E+05

0-SE tree based on mean is marked with *

0-SE tree based on median with bootstrap SE is marked with +

Selected-SE tree based on mean using naive SE is marked with **

Selected-SE tree based on mean using bootstrap SE is marked with --

Selected-SE tree based on median and bootstrap SE is marked with ++

++ tree same as -- tree

* tree same as ++ tree

* tree same as -- tree

Following tree is based on mean CV with naive SE estimate.

Structure of final tree. Each terminal node is marked with a T.

D-mean is mean of OutTuition in the node

Cases fit give the number of cases used to fit node

MSE and R² are based on all cases in node

Node label	Total cases	Cases fit	Matrix rank	Node D-mean	Node MSE	Node R ²	Split variable	Interact. variable	Linear variable
1	1134	1121	3	9.447E+03	1.093E+07	0.3852	InstExp		+CombSAT
2T	395	391	3	6.196E+03	2.634E+06	0.4433	InstExp		+RnBcost
3	739	730	3	1.119E+04	1.027E+07	0.3643	RnBcost		+CombSAT
6T	338	335	3	9.240E+03	6.851E+06	0.2677	GradRate		+CombSAT
7	401	395	3	1.284E+04	9.026E+06	0.4376	InstExp		+CombSAT
14T	226	223	3	1.092E+04	2.541E+06	0.6283	GradRate		+InstExp
15	175	172	3	1.532E+04	1.112E+07	0.3543	Top25		+CombSAT
30T	22	22	3	1.256E+04	4.716E+06	0.6560	Top10		+AppsAcc
31T	153	150	3	1.573E+04	9.825E+06	0.4036	GradRate		+CombSAT

Number of terminal nodes of final tree: 5

Total number of nodes of final tree: 9

Regression tree:

```

Node 1: InstExp <= 6.82350E+03
  Node 2: OutTuition-mean = 6.19621E+03
Node 1: InstExp > 6.82350E+03
  Node 3: RnBcost <= 4.34050E+03
    Node 6: OutTuition-mean = 9.24049E+03
  Node 3: RnBcost > 4.34050E+03
    Node 7: InstExp <= 1.09750E+04
      Node 14: OutTuition-mean = 1.09243E+04
    Node 7: InstExp > 1.09750E+04
      Node 15: Top25 <= 52.00000
        Node 30: OutTuition-mean = 1.25643E+04
      Node 15: Top25 > 52.00000
        Node 31: OutTuition-mean = 1.57251E+04

```

 In the following the predictor node mean is mean of complete cases
 Regression coefficients are computed from the complete cases

Node 1: Intermediate node

A case goes into Node 2 if InstExp <= 6.8235000E+03
 InstExp mean = 9.0272E+03

```

-----
Node 2: Terminal node
Coefficients of least squares regression function:
Regressor    Coefficient    t-stat  p-val      Min      Mean      Max
Constant     3.2926E+03      7.84  0.0000
RnBcost      1.2668E+00     11.67  0.0000    1.3060E+03  3.3979E+03  5.7000E+03
PubPr=Publ   -2.0137E+03    -10.82  0.0000    0.0000E+00  6.7263E-01  1.0000E+00
Predicted mean if regression function is inapplicable due to missing values =
  5714.000000000000
-----
Node 3: Intermediate node
A case goes into Node 6 if RnBcost <=  4.3405000E+03
      RnBcost mean =  4.5771E+03
-----
Node 6: Terminal node
Coefficients of least squares regression function:
Regressor    Coefficient    t-stat  p-val      Min      Mean      Max
Constant     -1.9797E+03    -1.26  0.0000
CombSAT      1.3039E+01     8.17  0.0000    7.4100E+02  9.7229E+02  1.2400E+03
PubPr=Publ   -3.2528E+03    -9.58  0.0000    0.0000E+00  3.2537E-01  1.0000E+00
Predicted mean if regression function is inapplicable due to missing values =
  8699.48965517241
-----
Node 7: Intermediate node
A case goes into Node 14 if InstExp <=  1.0975000E+04
      InstExp mean =  1.2518E+04
-----
Node 14: Terminal node
Coefficients of least squares regression function:
Regressor    Coefficient    t-stat  p-val      Min      Mean      Max
Constant     3.5454E+03     4.14  0.0000
InstExp      9.1933E-01     9.67  0.0000    6.8270E+03  8.8612E+03  1.0961E+04
PubPr=Publ   -4.3883E+03    -15.27  0.0000    0.0000E+00  1.7489E-01  1.0000E+00
Predicted mean if regression function is inapplicable due to missing values =
  10924.3408071749
-----
Node 15: Intermediate node
A case goes into Node 30 if Top25 <=  5.2000000E+01
      Top25 mean =  7.5573E+01
-----
Node 30: Terminal node
Coefficients of least squares regression function:
Regressor    Coefficient    t-stat  p-val      Min      Mean      Max
Constant     1.3193E+04    19.76  0.0000
AppsAcc      1.4415E+00     2.99  0.0000    7.0000E+01  1.6116E+03  6.0080E+03
PubPr=Publ   -1.0823E+04    -5.39  0.0000    0.0000E+00  2.7273E-01  1.0000E+00

```

```

Predicted mean if regression function is inapplicable due to missing values =
  12564.2727272727
-----
Node 31: Terminal node
Coefficients of least squares regression function:
Regressor      Coefficient      t-stat  p-val      Min      Mean      Max
Constant       6.0488E+02       0.28  0.0000
CombSAT        1.4041E+01       7.56  0.0000   8.3000E+02  1.1352E+03  1.4100E+03
PubPr=Publ    -6.5012E+03      -9.85  0.0000   0.0000E+00  1.1333E-01  1.0000E+00
Predicted mean if regression function is inapplicable due to missing values =
  15906.8000000000
-----

LaTeX code for tree is in file: ancova.tex

Proportion of variance (R-squared) explained by tree model =  0.7253

Elapsed time in seconds:    16.46575

```

Figure 5 shows the \LaTeX tree. The node sample mean and the selected linear regressor are given below each leaf node. When there is a high proportion of missing values, as in this example, a piecewise-constant model may be just as good in terms of prediction accuracy. Such a model is shown in Figure 6.

7.3 Quantile regression

Instead of estimating the conditional mean, we can estimate conditional quantiles (Koenker and Bassett, 1978; Chaudhuri and Loh, 2002). Following is a session log to do this for the 90th percentile for the tuition data.

```

Choose one of the following options:
1. Read the warranty disclaimer
2. Fit a model
3. Convert data to other formats
4. Create a batch input file
5. Rank and select regressor variables
Input your choice: 2
Input name of file to store results: quant90.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
You can fit a classification tree or a regression tree
Input 1 for classification, 2 for regression ([1:2], <cr>=1):2
Choose type of regression model:

```

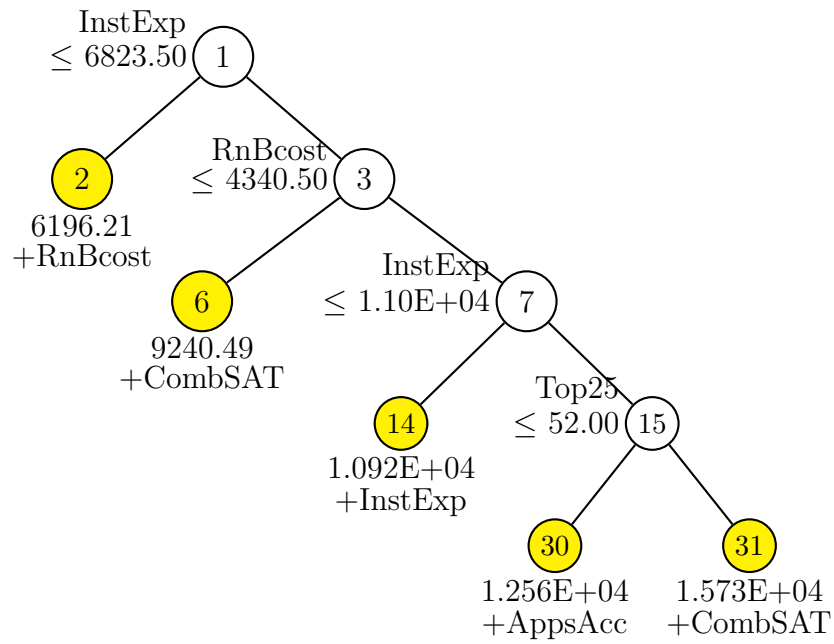


Figure 5: GUIDE piecewise ANCOVA regression tree model. At each intermediate node, a case goes to the left branch if and only if the condition is satisfied. Beneath each leaf node are sample mean of OutTuition and sign and name of the numerical predictor.

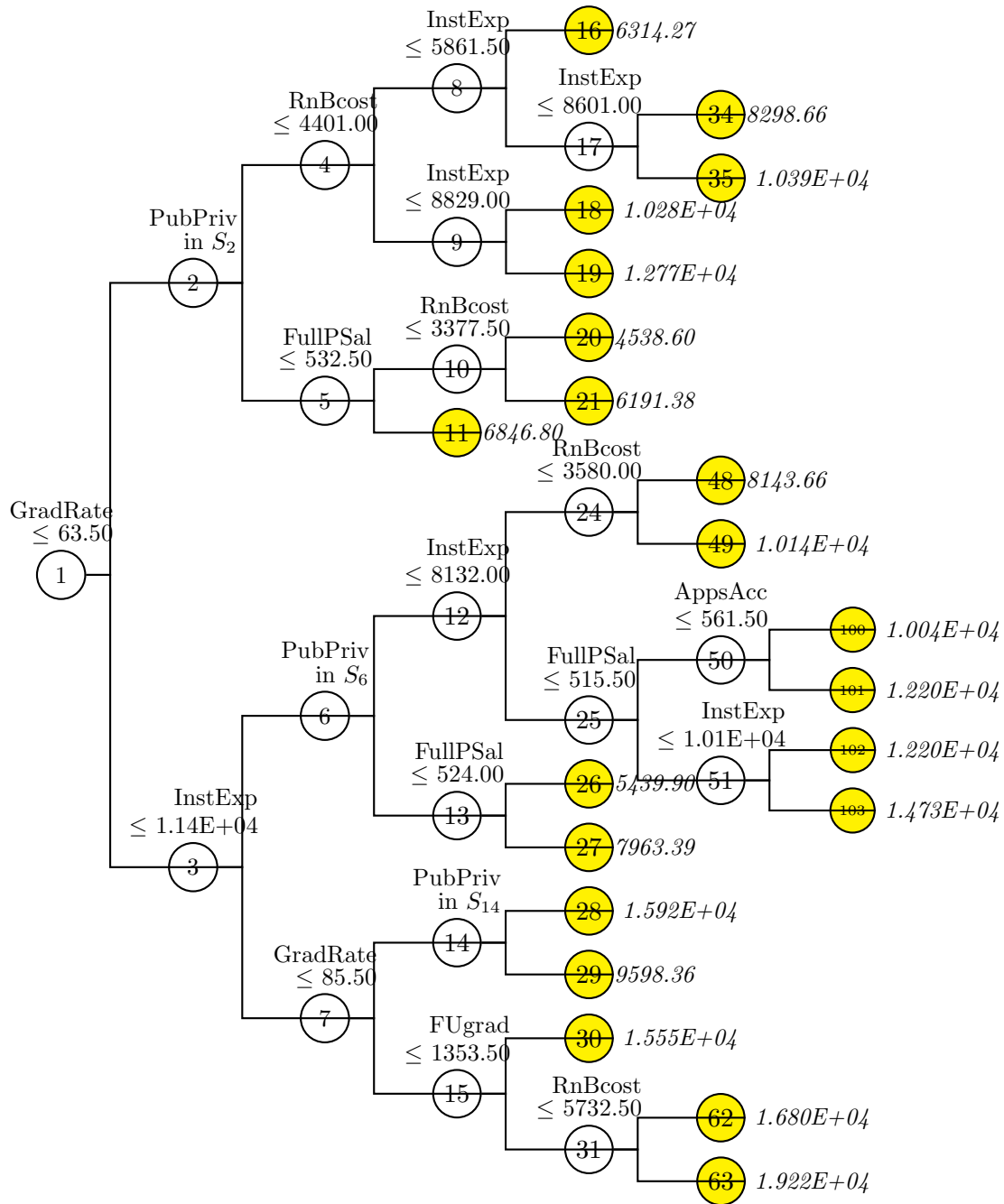


Figure 6: GUIDE piecewise constant least-squares regression tree model. At each intermediate node, a case goes to the upper branch if and only if the condition is satisfied. Number in italics beside leaf node is sample mean of OutTuition.

```

1=linear, 2=quantile, 3=Poisson, 4=proportional hazards, 5=multiresponse,
6=longitudinal data (requires T variables).
Input choice ([1:6], <cr>=1):2
Choose complexity of model to use at each node:
1: multiple linear, 2: best polynomial, 3: constant ([1:3], <cr>=1):3
Input quantile ([0.00:1.00], <cr>=0.50):.9
Input 1 for interaction tests, 2 to skip them ([1:2], <cr>=1):
Input 1 to prune by CV, 2 for no pruning ([1:2], <cr>=1):

Input name of data description file (max 100 chars; enclose within
quotes if it contains spaces): tuitiondsc.txt
Reading data description file ...
Training sample file: tuitiondat.txt
Missing value code: NA
Warning: N variables changed to S
Dependent variable is OutTuition
Warning: B variables changed to C
Length of longest data entry = 20
Total number of cases =      1134
Cat. var. in column  #levels (incl. missing)  #missing values
                   4                2                0
                   36               3                0

Checking data ...
      Total #cases w/ #cases w/
      #cases  miss. D  miss. val  #X-var  #N-var  #F-var  #S-var  #B-var  #C-var
      1134      13      621      32      0      0      14      1      1
Number of cases used for training =      1121
Number of cases excluded due to zero weight or missing D-values =      13
Default number of cross-validations =      10
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Best tree may be chosen based on mean or median CV estimate
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50):
Choose fraction of cases for splitting
Larger values give more splits: 0 = median split and 1 = all possible splits
Default fraction is 0.89206E-01
Choose 1 to accept default split fraction, 2 to change it
Input 1 or 2 ([1:2], <cr>=1):
Default max number of split levels =      10
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Smallest possible node sample size =      5
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Input 2 for LaTeX tree diagrams, 1 to skip them ([1:2], <cr>=1):2
Input file name to store LaTeX code (use .tex as suffix): quant90.tex
Input 1 for a vertical tree, 2 for a sideways tree ([1:2], <cr>=1):
Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1):2

```

Choose a color for the leaf nodes:

- (1) white
- (2) lightgray
- (3) gray
- (4) darkgray
- (5) black
- (6) yellow
- (7) red
- (8) blue
- (9) green
- (10) magenta
- (11) cyan

Input your choice ([1:11], <cr>=6):

You can store the variables and/or values used to split and fit in a file

Choose 1 to skip this step, 2 to store split and fit variables,

3 to store split variables and their values

Input your choice ([1:3], <cr>=1):

Input 2 to save predicted values and node IDs, 1 otherwise ([1:2], <cr>=1):

Input 2 to save terminal node IDs for importance scoring later; 1 otherwise ([1:2], <cr>=1):

Constructing main tree ...

Number of terminal nodes of largest tree = 149

Performing cross-validation:

Finished cross-validation iteration	1
Finished cross-validation iteration	2
Finished cross-validation iteration	3
Finished cross-validation iteration	4
Finished cross-validation iteration	5
Finished cross-validation iteration	6
Finished cross-validation iteration	7
Finished cross-validation iteration	8
Finished cross-validation iteration	9
Finished cross-validation iteration	10

Pruning main tree. Please wait.

Results of subtree sequence

Trees based on mean with naive SE are marked with * and **

Tree based on mean with bootstrap SE is marked with --

Trees based on median with finite bootstrap SE are marked with + and ++

Subtree	#Terminal nodes
1	148
2	147
3	146
4	145
5	144
6	143

7	142
8	141
9	140
10	138
11	137
12	136
13	135
14	134
15	133
16	132
17	131
18	129
19	128
20	127
21	125
22	124
23	123
24	122
25	121
26	117
27	116
28	114
29	111
30	110
31	109
32	108
33	106
34	104
35	103
36	102
37	101
38	100
39	99
40	98
41	97
42	96
43	95
44	93
45	92
46	91
47	90
48	88
49	86
50	85
51	83
52	82

53	81
54	79
55	78
56	76
57	75
58	73
59	72
60	71
61	70
62	68
63	67
64	66
65	65
66	64
67	62
68	61
69	60
70	59
71	58
72	53
73	52
74	51
75	49
76*	47
77	46
78	44
79	43
80	42
81	40
82	39
83	38
84	35
85	34
86	33
87	32
88	29
89	21
90	20
91	17
92	16
93	15
94	13
95++	9
96**	8
97	7
98	6

```

    99                5
   100                4
   101                3
   102                2
   103                1
** tree same as -- tree
+ tree same as ++ tree

```

LaTeX code for tree is in file: quant90.tex
 Results are stored in file: quant90.txt

Contents of quant90.txt

```

Quantile regression tree with quantile value 0.9000
Pruning by cross-validation
Data description file: tuitiondsc.txt
Training sample file: tuitiondat.txt
Missing value code: NA
Warning: N variables changed to S
Dependent variable is OutTuition
Warning: B variables changed to C
Piecewise constant model
Length of longest data entry = 20

```

Summary information (without x variables)

d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical, n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight
 For categorical variables, #categories include one for missing values

Column number	Variable name	Variable type	Minimum value	Maximum value	Number of categories	Number missing
4	PubPriv	c			2	
7	CombSAT	s	6.0000E+02	1.4100E+03		471
15	AppsRec	s	5.7000E+01	4.8094E+04		9
16	AppsAcc	s	4.4000E+01	2.6330E+04		9
17	NewEnrol	s	2.1000E+01	7.4250E+03		5
18	Top10	s	1.0000E+00	9.8000E+01		183
19	Top25	s	1.1000E+01	1.0000E+02		155
20	FUgrad	s	1.1800E+02	3.1643E+04		3
23	OutTuition	d	1.0440E+03	2.5750E+04		13
24	RnBcost	s	1.3060E+03	8.7000E+03		57
30	PFacPhD	s	8.0000E+00	1.0500E+02		29
32	StudFac	s	2.5000E+00	4.2600E+01		2
34	InstExp	s	1.8340E+03	6.2469E+04		24
35	GradRate	s	8.0000E+00	1.1800E+02		69
36	Type	c			3	

```

37 FullPSal          s  2.7000E+02  1.0090E+03          61
45 NFullProf        s  0.0000E+00  9.9700E+02

```

```

Total #cases w/ #cases w/
#cases miss. D miss. val #X-var #N-var #F-var #S-var #B-var #C-var
1134      13      621      32      0      0      14      1      1
Number of cases used for training = 1121
Number of cases excluded due to zero weight or missing D-values = 13

```

```

Interaction tests on all variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Fraction of cases used for splitting each node = 0.89206E-01
Max number of split levels = 10
Minimum node size = 5
Number of SE's for pruned tree = 5.0000E-01

```

Size and CV Loss and SE of subtrees:

Tree	#Tnodes	Mean Loss	SE(Mean)	BSE(Mean)	Median Loss	BSE(Median)
1	148	3.802E+02	1.351E+01	1.401E+01	3.901E+02	1.990E+01
2	147	3.802E+02	1.351E+01	1.401E+01	3.901E+02	1.990E+01
3	146	3.801E+02	1.350E+01	1.402E+01	3.900E+02	2.000E+01
4	145	3.802E+02	1.351E+01	1.403E+01	3.904E+02	2.005E+01
5	144	3.802E+02	1.351E+01	1.403E+01	3.904E+02	2.005E+01
6	143	3.802E+02	1.351E+01	1.402E+01	3.904E+02	2.003E+01
7	142	3.801E+02	1.351E+01	1.401E+01	3.904E+02	2.007E+01
8	141	3.802E+02	1.351E+01	1.399E+01	3.904E+02	1.996E+01
9	140	3.802E+02	1.351E+01	1.399E+01	3.904E+02	1.996E+01
10	138	3.802E+02	1.351E+01	1.398E+01	3.902E+02	1.989E+01
11	137	3.803E+02	1.351E+01	1.404E+01	3.902E+02	2.004E+01
12	136	3.811E+02	1.358E+01	1.426E+01	3.902E+02	2.034E+01
13	135	3.810E+02	1.365E+01	1.405E+01	3.902E+02	2.026E+01
14	134	3.810E+02	1.366E+01	1.407E+01	3.894E+02	2.016E+01
15	133	3.812E+02	1.365E+01	1.423E+01	3.894E+02	2.041E+01
16	132	3.816E+02	1.369E+01	1.412E+01	3.898E+02	2.046E+01
17	131	3.815E+02	1.369E+01	1.409E+01	3.894E+02	2.035E+01
18	129	3.813E+02	1.368E+01	1.408E+01	3.894E+02	2.014E+01
19	128	3.813E+02	1.369E+01	1.406E+01	3.894E+02	2.012E+01
20	127	3.810E+02	1.369E+01	1.400E+01	3.894E+02	2.020E+01
21	125	3.807E+02	1.369E+01	1.398E+01	3.883E+02	2.009E+01
22	124	3.809E+02	1.369E+01	1.403E+01	3.883E+02	2.000E+01
23	123	3.806E+02	1.368E+01	1.400E+01	3.866E+02	1.983E+01
24	122	3.798E+02	1.368E+01	1.413E+01	3.866E+02	1.955E+01
25	121	3.799E+02	1.368E+01	1.410E+01	3.866E+02	1.943E+01
26	117	3.799E+02	1.368E+01	1.410E+01	3.868E+02	1.943E+01
27	116	3.796E+02	1.368E+01	1.407E+01	3.851E+02	1.934E+01

28	114	3.800E+02	1.369E+01	1.416E+01	3.851E+02	1.919E+01
29	111	3.798E+02	1.369E+01	1.412E+01	3.855E+02	1.888E+01
30	110	3.785E+02	1.368E+01	1.425E+01	3.786E+02	1.954E+01
31	109	3.771E+02	1.365E+01	1.443E+01	3.776E+02	2.036E+01
32	108	3.761E+02	1.359E+01	1.466E+01	3.761E+02	2.143E+01
33	106	3.761E+02	1.359E+01	1.466E+01	3.761E+02	2.143E+01
34	104	3.761E+02	1.359E+01	1.466E+01	3.761E+02	2.143E+01
35	103	3.762E+02	1.358E+01	1.470E+01	3.761E+02	2.143E+01
36	102	3.757E+02	1.355E+01	1.474E+01	3.761E+02	2.199E+01
37	101	3.758E+02	1.355E+01	1.470E+01	3.761E+02	2.195E+01
38	100	3.758E+02	1.355E+01	1.470E+01	3.761E+02	2.195E+01
39	99	3.758E+02	1.355E+01	1.470E+01	3.761E+02	2.195E+01
40	98	3.758E+02	1.355E+01	1.470E+01	3.761E+02	2.195E+01
41	97	3.758E+02	1.355E+01	1.466E+01	3.761E+02	2.190E+01
42	96	3.756E+02	1.353E+01	1.464E+01	3.741E+02	2.199E+01
43	95	3.756E+02	1.358E+01	1.464E+01	3.741E+02	2.197E+01
44	93	3.759E+02	1.358E+01	1.469E+01	3.751E+02	2.157E+01
45	92	3.754E+02	1.358E+01	1.486E+01	3.751E+02	2.230E+01
46	91	3.749E+02	1.361E+01	1.444E+01	3.751E+02	2.327E+01
47	90	3.754E+02	1.367E+01	1.425E+01	3.751E+02	2.327E+01
48	88	3.755E+02	1.367E+01	1.429E+01	3.751E+02	2.325E+01
49	86	3.755E+02	1.367E+01	1.429E+01	3.751E+02	2.325E+01
50	85	3.751E+02	1.370E+01	1.455E+01	3.751E+02	2.423E+01
51	83	3.743E+02	1.366E+01	1.435E+01	3.753E+02	2.301E+01
52	82	3.742E+02	1.366E+01	1.430E+01	3.753E+02	2.299E+01
53	81	3.743E+02	1.367E+01	1.433E+01	3.753E+02	2.311E+01
54	79	3.741E+02	1.364E+01	1.442E+01	3.753E+02	2.346E+01
55	78	3.742E+02	1.360E+01	1.446E+01	3.753E+02	2.370E+01
56	76	3.742E+02	1.360E+01	1.446E+01	3.753E+02	2.370E+01
57	75	3.728E+02	1.351E+01	1.415E+01	3.753E+02	2.201E+01
58	73	3.722E+02	1.347E+01	1.395E+01	3.753E+02	2.171E+01
59	72	3.710E+02	1.341E+01	1.386E+01	3.749E+02	2.098E+01
60	71	3.691E+02	1.338E+01	1.422E+01	3.749E+02	2.124E+01
61	70	3.691E+02	1.338E+01	1.422E+01	3.749E+02	2.124E+01
62	68	3.731E+02	1.372E+01	1.601E+01	3.749E+02	2.226E+01
63	67	3.732E+02	1.370E+01	1.591E+01	3.753E+02	2.190E+01
64	66	3.694E+02	1.364E+01	1.575E+01	3.675E+02	2.336E+01
65	65	3.706E+02	1.374E+01	1.573E+01	3.730E+02	2.368E+01
66	64	3.706E+02	1.374E+01	1.594E+01	3.730E+02	2.382E+01
67	62	3.689E+02	1.357E+01	1.607E+01	3.730E+02	2.383E+01
68	61	3.692E+02	1.358E+01	1.632E+01	3.730E+02	2.463E+01
69	60	3.684E+02	1.354E+01	1.540E+01	3.730E+02	2.283E+01
70	59	3.690E+02	1.353E+01	1.536E+01	3.730E+02	2.258E+01
71	58	3.682E+02	1.347E+01	1.537E+01	3.730E+02	2.354E+01
72	53	3.656E+02	1.336E+01	1.518E+01	3.724E+02	2.097E+01
73	52	3.647E+02	1.321E+01	1.591E+01	3.708E+02	2.103E+01

74	51	3.640E+02	1.317E+01	1.685E+01	3.708E+02	2.117E+01
75	49	3.646E+02	1.317E+01	1.688E+01	3.732E+02	2.150E+01
76*	47	3.637E+02	1.316E+01	1.685E+01	3.685E+02	2.158E+01
77	46	3.645E+02	1.318E+01	1.719E+01	3.685E+02	2.179E+01
78	44	3.652E+02	1.320E+01	1.734E+01	3.685E+02	2.255E+01
79	43	3.657E+02	1.355E+01	1.755E+01	3.654E+02	2.540E+01
80	42	3.675E+02	1.355E+01	1.682E+01	3.655E+02	2.480E+01
81	40	3.710E+02	1.410E+01	1.498E+01	3.619E+02	2.336E+01
82	39	3.764E+02	1.482E+01	1.526E+01	3.656E+02	2.885E+01
83	38	3.776E+02	1.486E+01	1.533E+01	3.656E+02	2.880E+01
84	35	3.762E+02	1.479E+01	1.472E+01	3.656E+02	2.884E+01
85	34	3.749E+02	1.479E+01	1.420E+01	3.656E+02	2.855E+01
86	33	3.720E+02	1.468E+01	1.483E+01	3.566E+02	2.923E+01
87	32	3.676E+02	1.438E+01	1.458E+01	3.581E+02	2.801E+01
88	29	3.676E+02	1.438E+01	1.458E+01	3.581E+02	2.801E+01
89	21	3.690E+02	1.443E+01	1.448E+01	3.651E+02	2.796E+01
90	20	3.710E+02	1.443E+01	1.645E+01	3.651E+02	2.943E+01
91	17	3.681E+02	1.392E+01	1.573E+01	3.577E+02	2.279E+01
92	16	3.678E+02	1.403E+01	1.613E+01	3.618E+02	2.658E+01
93	15	3.687E+02	1.404E+01	1.529E+01	3.580E+02	2.360E+01
94	13	3.711E+02	1.404E+01	1.518E+01	3.589E+02	2.280E+01
95++	9	3.647E+02	1.322E+01	1.417E+01	3.558E+02	1.504E+01
96**	8	3.682E+02	1.267E+01	1.573E+01	3.731E+02	1.952E+01
97	7	3.806E+02	1.262E+01	1.675E+01	3.904E+02	2.469E+01
98	6	4.100E+02	1.324E+01	1.441E+01	4.165E+02	2.151E+01
99	5	4.295E+02	1.317E+01	1.261E+01	4.388E+02	1.626E+01
100	4	4.569E+02	1.372E+01	1.417E+01	4.530E+02	2.109E+01
101	3	4.949E+02	1.451E+01	1.464E+01	4.945E+02	1.538E+01
102	2	5.387E+02	1.434E+01	1.459E+01	5.317E+02	1.234E+01
103	1	8.711E+02	2.062E+01	1.414E+01	8.813E+02	2.128E+01

0-SE tree based on mean is marked with *

0-SE tree based on median with bootstrap SE is marked with +

Selected-SE tree based on mean using naive SE is marked with **

Selected-SE tree based on mean using bootstrap SE is marked with --

Selected-SE tree based on median and bootstrap SE is marked with ++

** tree same as -- tree

+ tree same as ++ tree

Following tree is based on mean CV with naive SE estimate.

Structure of final tree. Each terminal node is marked with a T.

D-quant is quantile of OutTuition in the node

Cases fit give the number of cases used to fit node

Node	Total	Cases	Matrix	Node	Split	Interacting
------	-------	-------	--------	------	-------	-------------

label	cases	fit	rank	D-quant	variable	variable
1	1134	1121	1	1.591E+04	InstExp	
2	803	793	1	1.143E+04	InstExp	
4	354	351	1	8.850E+03	PubPriv	
8T	111	111	1	9.985E+03	GradRate	
9T	243	240	1	7.604E+03	NFullProf	
5	449	442	1	1.225E+04	RnBcost	
10T	124	122	1	9.843E+03	PubPriv	
11	325	320	1	1.267E+04	InstExp	
22T	160	157	1	1.160E+04	RnBcost	
23T	165	163	1	1.324E+04	InstExp	
3	331	328	1	1.892E+04	InstExp	
6	192	191	1	1.613E+04	Top25	
12T	39	39	1	1.354E+04	PubPriv	
13T	153	152	1	1.630E+04	Top25	
7T	139	137	1	1.963E+04	RnBcost	

Number of terminal nodes of final tree: 8

Total number of nodes of final tree: 15

Regression tree:

```

Node 1: InstExp <= 9.50150E+03
  Node 2: InstExp <= 6.58100E+03
    Node 4: PubPriv = Private
      Node 8: OutTuition sample quantile = 9.98500E+03
    Node 4: PubPriv = Public
      Node 9: OutTuition sample quantile = 7.60400E+03
  Node 2: InstExp > 6.58100E+03
    Node 5: RnBcost <= 3.55250E+03
      Node 10: OutTuition sample quantile = 9.84300E+03
    Node 5: RnBcost > 3.55250E+03
      Node 11: InstExp <= 7.97900E+03
        Node 22: OutTuition sample quantile = 1.16000E+04
      Node 11: InstExp > 7.97900E+03
        Node 23: OutTuition sample quantile = 1.32400E+04
  Node 1: InstExp > 9.50150E+03
    Node 3: InstExp <= 1.30325E+04
      Node 6: Top25 <= 44.50000
        Node 12: OutTuition sample quantile = 1.35400E+04
      Node 6: Top25 > 44.50000
        Node 13: OutTuition sample quantile = 1.63040E+04
    Node 3: InstExp > 1.30325E+04
      Node 7: OutTuition sample quantile = 1.96290E+04

```

In the following the predictor node mean is mean of complete cases

Node 1: Intermediate node

A case goes into Node 2 if InstExp <= 9.5015000E+03

InstExp mean = 9.0272E+03

Node 2: Intermediate node

A case goes into Node 4 if InstExp <= 6.5810000E+03

InstExp mean = 6.7454E+03

Node 4: Intermediate node

A case goes into Node 8 if PubPriv = Private

PubPriv mode = Public

Node 8: Terminal node

Coefficients of quantile regression function:

Regressor	Coefficient
Constant	9.9850E+03
Predicted quantile = 9985.00000000000	

Constant 9.9850E+03

Predicted quantile = 9985.00000000000

Node 9: Terminal node

Coefficients of quantile regression function:

Regressor	Coefficient
Constant	7.6040E+03
Predicted quantile = 7604.00000000000	

Constant 7.6040E+03

Predicted quantile = 7604.00000000000

Node 5: Intermediate node

A case goes into Node 10 if RnBcost <= 3.5525000E+03

RnBcost mean = 4.1096E+03

Node 10: Terminal node

Coefficients of quantile regression function:

Regressor	Coefficient
Constant	9.8430E+03
Predicted quantile = 9843.00000000000	

Constant 9.8430E+03

Predicted quantile = 9843.00000000000

Node 11: Intermediate node

A case goes into Node 22 if InstExp <= 7.9790000E+03

InstExp mean = 8.0235E+03

Node 22: Terminal node

Coefficients of quantile regression function:

Regressor	Coefficient
Constant	1.1600E+04
Predicted quantile = 11600.00000000000	

Constant 1.1600E+04

Predicted quantile = 11600.00000000000

```

Node 23: Terminal node
Coefficients of quantile regression function:
Regressor   Coefficient
Constant    1.3240E+04
Predicted quantile = 13240.0000000000
-----
Node 3: Intermediate node
A case goes into Node 6 if InstExp <= 1.3032500E+04
          InstExp mean = 1.4384E+04
-----
Node 6: Intermediate node
A case goes into Node 12 if Top25 <= 4.4500000E+01
          Top25 mean = 5.8635E+01
-----
Node 12: Terminal node
Coefficients of quantile regression function:
Regressor   Coefficient
Constant    1.3540E+04
Predicted quantile = 13540.0000000000
-----
Node 13: Terminal node
Coefficients of quantile regression function:
Regressor   Coefficient
Constant    1.6304E+04
Predicted quantile = 16304.0000000000
-----
Node 7: Terminal node
Coefficients of quantile regression function:
Regressor   Coefficient
Constant    1.9629E+04
Predicted quantile = 19629.0000000000
-----

LaTeX code for tree is in file: quant90.tex

```

The \LaTeX tree is shown in Figure 7.

7.4 Longitudinal data

GUIDE can fit piecewise-constant regression tree models to longitudinal response data. These variables must be designated as D variables in the description file, which must also contain the corresponding “time” variables (designated as T). The ordering of the D variables is assumed to match that of the T variables. The data file must be

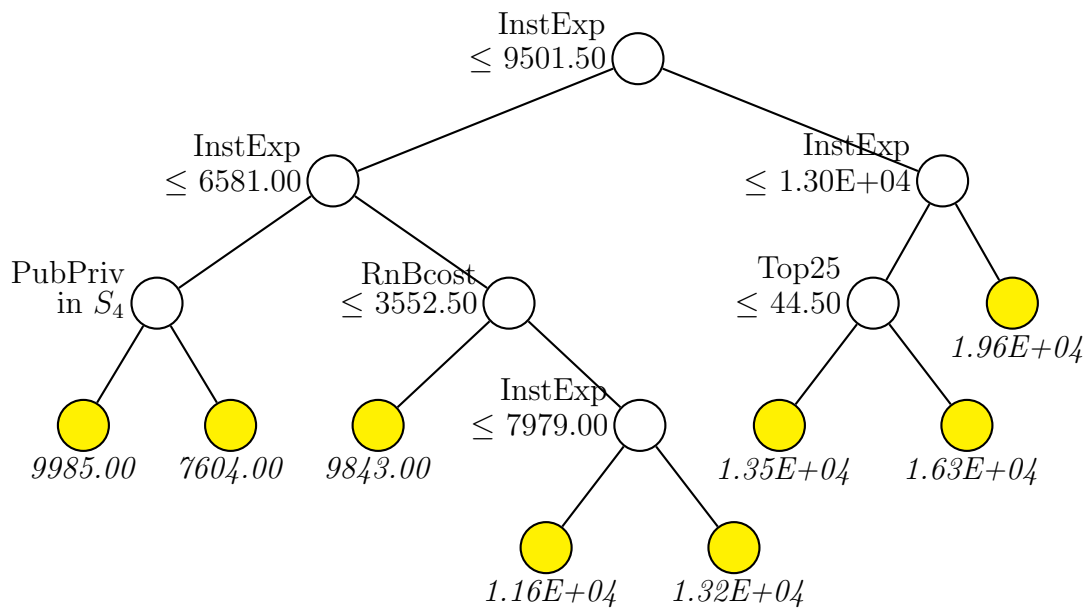


Figure 7: GUIDE piecewise constant quantile regression tree model. At each intermediate node, a case goes to the left branch if and only if the condition is satisfied. Number in italics beneath a leaf is sample quantile of OutTuition.

in “wide” format, with each row representing one individual. We use the AIDS data from [Fitzmaurice et al. \(2004, pp. 224–1230\)](#) for illustration. The data are in the file `cd4-wide.txt`. The description file `cd4dsc.txt` and the session log are given below.

```
cd4-wide.txt
NA
c1 c2 c3
1 ID x
2 Treatment c
3 Age s
4 Gender c
5 Week1 t
6 logCD41 d
7 Week2 t
8 logCD42 d
9 Week3 t
10 logCD43 d
11 Week4 t
12 logCD44 d
13 Week5 t
14 logCD45 d
15 Week6 t
16 logCD46 d
17 Week7 t
18 logCD47 d
19 Week8 t
20 logCD48 d
21 Week9 t
22 logCD49 d
```

7.4.1 Session log

```
Choose one of the following options:
1. Read the warranty disclaimer
2. Fit a model
3. Convert data to other formats
4. Create a batch input file
5. Rank and select regressor variables
Input your choice: 2
Input name of file to store results: cd4out.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
You can fit a classification tree or a regression tree
Input 1 for classification, 2 for regression ([1:2], <cr>=1):2
Choose type of regression model:
```

```

1=linear, 2=quantile, 3=Poisson, 4=proportional hazards, 5=multiresponse,
6=longitudinal data (requires T variables).
Input choice ([1:6], <cr>=1):6
Input 1 for lowess smoothing, 2 for spline smoothing ([1:2], <cr>=1):
Lowess gives smoother fits.
Input 1 for interaction tests, 2 to skip them ([1:2], <cr>=1):
Input 1 to prune by CV, 2 for no pruning ([1:2], <cr>=1):

Input name of data description file (max 100 chars; enclose within
quotes if it contains spaces): cd4dsc.txt
Reading data description file ...
Training sample file: cd4-wide.txt
Missing value code: NA
Number of D variables =          9
D variables are:
logCD41
logCD42
logCD43
logCD44
logCD45
logCD46
logCD47
logCD48
logCD49
T variables are:
Week1
Week2
Week3
Week4
Week5
Week6
Week7
Week8
Week9
The D variables can be grouped into segments to look for patterns
Input 1 for equal-sized groups, 2 for customized groups ([1:2], <cr>=1):
If option 2 is selected, you will be asked to specify the group boundaries.
Input number of equal-sized groups ([2:9], <cr>=3):2
We choose two equal-sized groups here.
Input number of interpolating points for prediction ([10:100], <cr>=31):
Input 1 to normalize D variables, 2 for no normalization ([1:2], <cr>=2):
Normalization is recommended if the D variables are not measured in the same scale.
Length of longest data entry = 12
Total number of cases =      1309
Cat. var. in column   #levels (incl. missing)   #missing values
                   2                       4                       0

```

```

                                4                2                0
Checking data ...
#cases w/ miss. D refers to number of cases with all D values missing
  Total #cases w/ #cases w/
  #cases miss. D miss. val #X-var #N-var #F-var #S-var #B-var #C-var
    1309      0    1308      1      0      0      1      0      2
Number of cases used for training =          1309
Number of cases excluded due to zero weight or missing D-values =          0
Default number of cross-validations =          10
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Best tree may be chosen based on mean or median CV estimate
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50):0
We choose 0-SE for illustration here; the default 0.5-SE tree has no splits.
Choose a split point selection method for numerical variables:
Choose 1 to use faster method based on sample quantiles
Choose 2 to use exhaustive search
Input 1 or 2 ([1:2], <cr>=2):
Default max number of split levels =          10
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Smallest possible node sample size =          26
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Input 2 for LaTeX tree diagrams, 1 to skip them ([1:2], <cr>=1):2
Input file name to store LaTeX code (use .tex as suffix): cd4.tex
Input 1 for a vertical tree, 2 for a sideways tree ([1:2], <cr>=1):
Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1):2
Choose a color for the leaf nodes:
(1) white
(2) lightgray
(3) gray
(4) darkgray
(5) black
(6) yellow
(7) red
(8) blue
(9) green
(10) magenta
(11) cyan
Input your choice ([1:11], <cr>=6):
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save node IDs of cases, 1 otherwise ([1:2], <cr>=1):2
Input name of file to store terminal node IDs: cd4node.txt
Input 1 to save fitted values at each node; 2 otherwise ([1:2], <cr>=2):

```

```

Input 2 to save terminal node IDs for importance scoring later; 1 otherwise ([1:2], <cr>=1):
Constructing main tree ...
Number of terminal nodes of largest tree =          33
Performing cross-validation:
Finished cross-validation iteration          1
Finished cross-validation iteration          2
Finished cross-validation iteration          3
Finished cross-validation iteration          4
Finished cross-validation iteration          5
Finished cross-validation iteration          6
Finished cross-validation iteration          7
Finished cross-validation iteration          8
Finished cross-validation iteration          9
Finished cross-validation iteration         10

```

Pruning main tree. Please wait.

Results of subtree sequence

Trees based on mean with naive SE are marked with * and **

Tree based on mean with bootstrap SE is marked with --

Trees based on median with finite bootstrap SE are marked with + and ++

Subtree	#Terminal nodes
1	31
2	30
3	29
4	28
5	27
6	26
7	24
8	23
9	22
10++	19
11	14
12	9
13	8
14	7
15	6
16	5
17	4
18**	2
19	1

** tree same as -- tree

+ tree same as ++ tree

* tree same as ** tree

* tree same as -- tree

LaTeX code for tree is in file: cd4.tex

Case and node IDs are in file: cd4node.txt
 Results are stored in file: cd4out.txt

7.4.2 Contents of output file cd4out.txt

```

Lowess smoothing
Longitudinal data with T variables
Pruning by cross-validation
Data description file: cd4dsc.txt
Training sample file: cd4-wide.txt
Missing value code: NA
Number of D variables = 9
D variables are:
logCD41
logCD42
logCD43
logCD44
logCD45
logCD46
logCD47
logCD48
logCD49
T variables are:
Week1
Week2
Week3
Week4
Week5
Week6
Week7
Week8
Week9
Time interval divided into 2 equal-sized groups
D variables not normalized for distance calculations
Piecewise constant model
Length of longest data entry = 12

Summary information (without x variables)
d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical,
n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight
For categorical variables, #categories include one for missing values
Column Variable      Variable Minimum      Maximum      Number of      Number
number  name              type    value         value         categories     missing
   2  Treatment          c                4
   3  Age                s  1.4902E+01  7.4193E+01

```

4	Gender	c				2	
5	Week1	t	0.0000E+00	1.7857E+01			
6	logCD41	d	0.0000E+00	5.1985E+00			
7	Week2	t	2.1429E+00	3.8000E+01			122
8	logCD42	d	0.0000E+00	6.2971E+00			122
9	Week3	t	7.2857E+00	4.0000E+01			273
10	logCD43	d	0.0000E+00	5.7714E+00			273
11	Week4	t	1.5857E+01	4.0000E+01			526
12	logCD44	d	0.0000E+00	6.1003E+00			526
13	Week5	t	1.9714E+01	4.0000E+01			779
14	logCD45	d	0.0000E+00	5.6595E+00			779
15	Week6	t	2.3571E+01	4.0000E+01			1126
16	logCD46	d	0.0000E+00	5.3660E+00			1126
17	Week7	t	2.7571E+01	3.9857E+01			1303
18	logCD47	d	0.0000E+00	3.4012E+00			1303
19	Week8	t	3.1714E+01	3.1714E+01			1308
20	logCD48	d	2.4849E+00	2.4849E+00			1308
21	Week9	t	3.9143E+01	3.9143E+01			1308
22	logCD49	d	2.3979E+00	2.3979E+00			1308

#cases w/ miss. D refers to number of cases with all D values missing

Total #cases	#cases w/ miss. D	#cases w/ miss. val	#X-var	#N-var	#F-var	#S-var	#B-var	#C-var
1309	0	1308	1	0	0	1	0	2

Number of cases used for training = 1309

Number of cases excluded due to zero weight or missing D-values = 0

Interaction tests on all variables

Pruning by v-fold cross-validation, with v = 10

Selected tree is based on mean of CV estimates

Split values for N and S variables based on exhaustive search

Max number of split levels = 10

Minimum node size = 26

Number of SE's for pruned tree = 0.0000E+00

Size and CV Loss and SE of subtrees:

Tree	#Tnodes	Mean Loss	SE(Mean)	BSE(Mean)	Median Loss	BSE(Median)
1	31	1.301E+01	1.884E-01	1.551E-01	1.305E+01	2.187E-01
2	30	1.301E+01	1.886E-01	1.543E-01	1.304E+01	2.138E-01
3	29	1.301E+01	1.881E-01	1.491E-01	1.304E+01	2.115E-01
4	28	1.301E+01	1.882E-01	1.488E-01	1.303E+01	2.116E-01
5	27	1.300E+01	1.879E-01	1.463E-01	1.302E+01	2.127E-01
6	26	1.299E+01	1.877E-01	1.485E-01	1.302E+01	2.079E-01
7	24	1.296E+01	1.875E-01	1.452E-01	1.299E+01	2.048E-01
8	23	1.296E+01	1.874E-01	1.452E-01	1.297E+01	2.055E-01
9	22	1.295E+01	1.873E-01	1.469E-01	1.296E+01	2.170E-01

10++	19	1.292E+01	1.853E-01	1.377E-01	1.294E+01	2.221E-01
11	14	1.292E+01	1.861E-01	1.378E-01	1.295E+01	2.258E-01
12	9	1.294E+01	1.857E-01	1.373E-01	1.298E+01	2.118E-01
13	8	1.293E+01	1.857E-01	1.366E-01	1.298E+01	2.095E-01
14	7	1.291E+01	1.847E-01	1.448E-01	1.298E+01	2.154E-01
15	6	1.287E+01	1.840E-01	1.354E-01	1.298E+01	1.971E-01
16	5	1.288E+01	1.841E-01	1.345E-01	1.298E+01	1.909E-01
17	4	1.285E+01	1.827E-01	1.377E-01	1.300E+01	1.974E-01
18**	2	1.284E+01	1.824E-01	1.336E-01	1.299E+01	1.950E-01
19	1	1.286E+01	1.851E-01	1.374E-01	1.301E+01	2.013E-01

0-SE tree based on mean is marked with *

0-SE tree based on median with bootstrap SE is marked with +

Selected-SE tree based on mean using naive SE is marked with **

Selected-SE tree based on mean using bootstrap SE is marked with --

Selected-SE tree based on median and bootstrap SE is marked with ++

** tree same as -- tree

+ tree same as ++ tree

* tree same as ** tree

* tree same as -- tree

Following tree is based on mean CV with naive SE estimate.

Structure of final tree. Each terminal node is marked with a T.

Cases fit give the number of cases used to fit node

MSE is residual sum of squares divided by number of cases in node

Node label	Total cases	Cases fit	Node MSE	Split variable	Interacting variable
1	1309	1309	4.421E+00	Treatment	
2T	330	330	5.066E+00	Gender	
3T	979	979	4.136E+00	Treatment	

Number of terminal nodes of final tree: 2

Total number of nodes of final tree: 3

Regression tree for longitudinal data:

Node 1: Treatment = 4

Node 2: Mean cost = 5.06639E+00

Node 1: Treatment = 1, 2, 3

Node 3: Mean cost = 4.13630E+00

In the following the predictor node mean is mean of complete cases

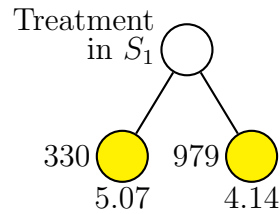


Figure 8: GUIDE regression tree model for longitudinal data. At each intermediate node, a case goes to the left branch if and only if the condition is satisfied. Sample size on left and MSE beneath each leaf node. $S_1 = \{4\}$.

```

Node 1: Intermediate node
A case goes into Node 2 if Treatment = 4
      Treatment mode = 3
-----
  
```

```

Node 2: Terminal node
-----
  
```

```

Node 3: Terminal node
-----
  
```

```

LaTeX code for tree is in file: cd4.tex
Case and node IDs are in file: cd4node.txt
  
```

The single-split tree is shown in Figure 8.

7.5 Multi-response data

GUIDE can fit a piecewise-constant regression model for two or more dependent variables simultaneously if there are no T variables. We use the tuition data set as an example, with in-state and out-of-state tuition being the two variables designated as D variables. The contents of the description file `tuitiondsc2.txt` are:

```

tuitiondat.txt
NA
col_num var_name var_type
1 FICE x
2 CollName x
3 State x
4 PubPriv b
5 MathSAT x
6 VerbSAT x
  
```

7 CombsAT n
8 ACT x
9 Q1MSAT x
10 Q3MSAT x
11 Q1VSAT x
12 Q3VSAT x
13 Q1ACT x
14 Q3ACT x
15 AppsRec n
16 AppsAcc n
17 NewEnrol n
18 Top10 n
19 Top25 n
20 FUgrad n
21 PUgrad x
22 InTuition d
23 OutTuition d
24 RnBcost n
25 RmCost x
26 BrdCost x
27 AddFees x
28 BookCost x
29 PerSpend x
30 PFacPhD n
31 PFacTerm x
32 StudFac n
33 PAIDonate x
34 InstExp n
35 GradRate n
36 Type c
37 FullPSal n
38 AssocPSal x
39 AsstPSal x
40 AveSal x
41 FullPComp x
42 AssocPComp x
43 AsstPComp x
44 AveComp x
45 NFullProf n
46 NAssocProf x
47 NAsstProf x
48 NInstr x
49 NAllFac x

Following is a session log.

```

Choose one of the following options:
1. Read the warranty disclaimer
2. Fit a model
3. Convert data to other formats
4. Create a batch input file
5. Rank and select regressor variables
Input your choice: 2
Input name of file to store results: multi.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
You can fit a classification tree or a regression tree
Input 1 for classification, 2 for regression ([1:2], <cr>=1):2
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards, 5=multiresponse,
6=longitudinal data (requires T variables).
Input choice ([1:6], <cr>=1):5
Input 1 for interaction tests, 2 to skip them ([1:2], <cr>=1):
Input 1 to prune by CV, 2 for no pruning ([1:2], <cr>=1):

Input name of data description file (max 100 chars; enclose within
quotes if it contains spaces): tuitiondsc2.txt
Reading data description file ...
Training sample file: tuitiondat.txt
Missing value code: NA
Warning: N variables changed to S
Number of D variables =          2
D variables are:
InTuition
OutTuition
Input 1 to normalize D variables, 2 for no normalization ([1:2], <cr>=1):
Option 1 will normalize the D variables by dividing the values of each by their
standard deviation. This option is particularly useful if the D variables have
different units.
Warning: B variables changed to C
Length of longest data entry = 20
Total number of cases =          1134
Cat. var. in column   #levels (incl. missing)   #missing values
                   4                          2                0
                   36                         3                0

Checking data ...
#cases w/ miss. D refers to number of cases with all D values missing
  Total #cases w/ #cases w/
#cases  miss. D  miss. val  #X-var  #N-var  #F-var  #S-var  #B-var  #C-var
  1134     12     619     31     0     0     14     1     1
Number of cases used for training =          1122
Number of cases excluded due to zero weight or missing D-values =          12
Default number of cross-validations =          10

```

```

Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Best tree may be chosen based on mean or median CV estimate
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50):
Choose a split point selection method for numerical variables:
Choose 1 to use faster method based on sample quantiles
Choose 2 to use exhaustive search
Input 1 or 2 ([1:2], <cr>=2):
Default max number of split levels =          10
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Smallest possible node sample size =          22
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Input 2 for LaTeX tree diagrams, 1 to skip them ([1:2], <cr>=1):2
Input file name to store LaTeX code (use .tex as suffix): multi.tex
Input 1 for a vertical tree, 2 for a sideways tree ([1:2], <cr>=1):
Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1):2
Choose a color for the leaf nodes:
(1) white
(2) lightgray
(3) gray
(4) darkgray
(5) black
(6) yellow
(7) red
(8) blue
(9) green
(10) magenta
(11) cyan
Input your choice ([1:11], <cr>=6):
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save node IDs of cases, 1 otherwise ([1:2], <cr>=1):2
Input name of file to store terminal node IDs: multinode.txt
This file shows which node each case belongs to.
Input 1 to save fitted values at each node; 2 otherwise ([1:2], <cr>=2):1
Input name of file to store node fitted values: multifit.txt
Input 2 to save terminal node IDs for importance scoring later; 1 otherwise ([1:2], <cr>=1):
Constructing main tree ...
Number of terminal nodes of largest tree =          39
Performing cross-validation:
Finished cross-validation iteration          1
Finished cross-validation iteration          2
Finished cross-validation iteration          3
Finished cross-validation iteration          4

```

```

Finished cross-validation iteration      5
Finished cross-validation iteration      6
Finished cross-validation iteration      7
Finished cross-validation iteration      8
Finished cross-validation iteration      9
Finished cross-validation iteration     10

```

Pruning main tree. Please wait.

Results of subtree sequence

Trees based on mean with naive SE are marked with * and **

Tree based on mean with bootstrap SE is marked with --

Trees based on median with finite bootstrap SE are marked with + and ++

Subtree	#Terminal nodes
1	38
2+	37
3*	36
4	35
5	34
6	32
7	31
8	30
9	29
10	28
11	27
12	26
13	25
14	22
15	21
16**	20
17	19
18++	18
19	17
20	16
21	14
22	13
23	12
24	11
25	10
26	9
27	8
28	7
29	6
30	5
31	4
32	3
33	2

```

34                                1
** tree same as -- tree

LaTeX code for tree is in file: multi.tex
Case and node IDs are in file: multinode.txt
Results are stored in file: multi.txt

```

7.5.1 Contents of multi.txt

```

Multi-response or longitudinal data without T variables
Pruning by cross-validation
Data description file: tuitiondsc2.txt
Training sample file: tuitiondat.txt
Missing value code: NA
Warning: N variables changed to S
Number of D variables = 2
D variables are:
InTuition
OutTuition
Segment boundaries are:
1.5
Mean-squared errors (MSE) are calculated from normalized D variables
Warning: B variables changed to C
Piecewise constant model
Length of longest data entry = 20

Summary information (without x variables)
d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical,
n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight
For categorical variables, #categories include one for missing values

```

Column number	Variable name	Variable type	Minimum value	Maximum value	Number of categories	Number missing
4	PubPriv	c			2	
7	CombSAT	s	6.0000E+02	1.4100E+03		471
15	AppsRec	s	5.7000E+01	4.8094E+04		9
16	AppsAcc	s	4.4000E+01	2.6330E+04		9
17	NewEnrol	s	2.1000E+01	7.4250E+03		5
18	Top10	s	1.0000E+00	9.8000E+01		183
19	Top25	s	1.1000E+01	1.0000E+02		155
20	FUgrad	s	1.1800E+02	3.1643E+04		3
22	InTuition	d	4.8000E+02	2.5750E+04		23
23	OutTuition	d	1.0440E+03	2.5750E+04		13
24	RnBcost	s	1.3060E+03	8.7000E+03		57
30	PFacPhD	s	8.0000E+00	1.0500E+02		29
32	StudFac	s	2.5000E+00	4.2600E+01		2

34	InstExp	s	1.8340E+03	6.2469E+04					24
35	GradRate	s	8.0000E+00	1.1800E+02					69
36	Type	c					3		
37	FullPSal	s	2.7000E+02	1.0090E+03					61
45	NFullProf	s	0.0000E+00	9.9700E+02					

#cases w/ miss. D refers to number of cases with all D values missing

Total #cases	#cases w/ miss. D	#cases w/ miss. val	#X-var	#N-var	#F-var	#S-var	#B-var	#C-var
1134	12	619	31	0	0	14	1	1

Number of cases used for training = 1122

Number of cases excluded due to zero weight or missing D-values = 12

Interaction tests on all variables

Pruning by v-fold cross-validation, with v = 10

Selected tree is based on mean of CV estimates

Split values for N and S variables based on exhaustive search

Max number of split levels = 10

Minimum node size = 22

Number of SE's for pruned tree = 5.0000E-01

Size and CV Loss and SE of subtrees:

Tree	#Tnodes	Mean Loss	SE(Mean)	BSE(Mean)	Median Loss	BSE(Median)
1	38	3.468E-01	2.225E-02	2.114E-02	3.533E-01	2.366E-02
2+	37	3.468E-01	2.225E-02	2.119E-02	3.533E-01	2.384E-02
3*	36	3.465E-01	2.223E-02	2.135E-02	3.539E-01	2.413E-02
4	35	3.466E-01	2.223E-02	2.153E-02	3.539E-01	2.431E-02
5	34	3.476E-01	2.232E-02	2.137E-02	3.580E-01	2.469E-02
6	32	3.477E-01	2.232E-02	2.151E-02	3.580E-01	2.460E-02
7	31	3.479E-01	2.232E-02	2.132E-02	3.581E-01	2.376E-02
8	30	3.483E-01	2.234E-02	2.151E-02	3.602E-01	2.499E-02
9	29	3.499E-01	2.239E-02	2.159E-02	3.655E-01	2.591E-02
10	28	3.503E-01	2.240E-02	2.162E-02	3.676E-01	2.619E-02
11	27	3.512E-01	2.241E-02	2.121E-02	3.676E-01	2.549E-02
12	26	3.514E-01	2.240E-02	2.122E-02	3.681E-01	2.543E-02
13	25	3.515E-01	2.238E-02	2.125E-02	3.681E-01	2.545E-02
14	22	3.517E-01	2.238E-02	2.127E-02	3.681E-01	2.565E-02
15	21	3.524E-01	2.238E-02	2.093E-02	3.681E-01	2.565E-02
16**	20	3.555E-01	2.237E-02	2.036E-02	3.631E-01	2.244E-02
17	19	3.593E-01	2.249E-02	2.045E-02	3.631E-01	2.528E-02
18++	18	3.593E-01	2.259E-02	2.069E-02	3.629E-01	2.533E-02
19	17	3.605E-01	2.261E-02	2.115E-02	3.662E-01	2.592E-02
20	16	3.604E-01	2.260E-02	2.099E-02	3.680E-01	2.449E-02
21	14	3.660E-01	2.135E-02	2.201E-02	3.784E-01	2.834E-02
22	13	3.736E-01	2.141E-02	2.166E-02	3.803E-01	2.732E-02
23	12	3.736E-01	2.141E-02	2.166E-02	3.803E-01	2.732E-02

24	11	3.760E-01	2.144E-02	2.281E-02	3.808E-01	2.743E-02
25	10	3.784E-01	2.153E-02	2.298E-02	3.825E-01	2.995E-02
26	9	3.975E-01	2.227E-02	2.753E-02	3.825E-01	2.767E-02
27	8	4.260E-01	2.367E-02	2.758E-02	4.182E-01	2.498E-02
28	7	4.461E-01	2.408E-02	3.268E-02	4.304E-01	2.939E-02
29	6	4.686E-01	2.508E-02	3.162E-02	4.573E-01	3.745E-02
30	5	4.803E-01	2.568E-02	3.029E-02	4.846E-01	3.525E-02
31	4	4.895E-01	2.598E-02	2.972E-02	4.861E-01	3.625E-02
32	3	6.271E-01	3.314E-02	4.141E-02	6.048E-01	4.748E-02
33	2	9.193E-01	4.731E-02	3.159E-02	8.856E-01	4.690E-02
34	1	1.949E+00	7.140E-02	2.720E-02	1.951E+00	3.440E-02

0-SE tree based on mean is marked with *

0-SE tree based on median with bootstrap SE is marked with +

Selected-SE tree based on mean using naive SE is marked with **

Selected-SE tree based on mean using bootstrap SE is marked with --

Selected-SE tree based on median and bootstrap SE is marked with ++

** tree same as -- tree

Following tree is based on mean CV with naive SE estimate.

Structure of final tree. Each terminal node is marked with a T.

Cases fit give the number of cases used to fit node

MSE is residual sum of squares divided by number of cases in node

Node label	Total cases	Cases fit	Node MSE	Split variable	Interacting variable
1	1134	1122	1.904E+00	PubPriv	
2	692	687	1.320E+00	PFacPhD	
4	463	459	7.387E-01	RnBcost	
8	280	277	5.975E-01	RnBcost	
16	143	140	4.849E-01	AppsRec	InstExp
32	89	89	4.053E-01	InstExp	
64T	32	32	2.477E-01		
65T	57	57	3.445E-01	NewEnrol	
33T	54	51	5.699E-01		
17	137	137	5.136E-01	GradRate	
34	86	86	4.710E-01	InstExp	
68T	63	63	2.510E-01	GradRate	
69T	23	23	9.106E-01		
35T	51	51	3.616E-01	NFullProf	
9	183	182	4.976E-01	InstExp	
18	134	134	2.516E-01	InstExp	
36T	68	68	2.094E-01	RnBcost	
37T	66	66	1.769E-01	GradRate	
19	49	48	5.461E-01	AppsAcc	

38T	26	25	4.671E-01		
39T	23	23	2.460E-01		
5	229	228	1.080E+00	InstExp	
10	106	105	4.998E-01	InstExp	
20	48	47	3.082E-01	RnBcost	FullPSal
40T	22	22	3.881E-01		
41T	26	25	1.085E-01		
21T	58	58	3.667E-01	InstExp	
11	123	123	6.797E-01	RnBcost	
22	49	49	5.738E-01	NewEnrol	
44T	22	22	3.955E-01		
45T	27	27	5.962E-01		
23	74	74	4.347E-01	GradRate	
46T	22	22	2.672E-01		
47T	52	52	4.335E-01	StudFac	
3	442	435	2.554E-01	FullPSal	
6	287	283	1.494E-01	RnBcost	
12T	120	120	1.071E-01	FullPSal	
13T	167	163	1.270E-01	Top25	
7T	155	152	2.682E-01	Top10	

Number of terminal nodes of final tree: 20

Total number of nodes of final tree: 39

Regression tree for multi-response or longitudinal data:

Node 1: PubPriv = Private

Node 2: PFacPhD <= 78.50000

Node 4: RnBcost <= 4.27500E+03

Node 8: RnBcost <= 3.58850E+03

Node 16: AppsRec <= 7.86500E+02

Node 32: InstExp <= 6.35200E+03

Node 64: Mean cost = 2.47697E-01

Node 32: InstExp > 6.35200E+03

Node 65: Mean cost = 3.44460E-01

Node 16: AppsRec > 7.86500E+02

Node 33: Mean cost = 5.69918E-01

Node 8: RnBcost > 3.58850E+03

Node 17: GradRate <= 65.50000

Node 34: InstExp <= 8.33900E+03

Node 68: Mean cost = 2.51000E-01

Node 34: InstExp > 8.33900E+03

Node 69: Mean cost = 9.10579E-01

Node 17: GradRate > 65.50000

Node 35: Mean cost = 3.61553E-01

Node 4: RnBcost > 4.27500E+03

```

Node 9: InstExp <= 9.73050E+03
Node 18: InstExp <= 7.94850E+03
Node 36: Mean cost = 2.09432E-01
Node 18: InstExp > 7.94850E+03
Node 37: Mean cost = 1.76938E-01
Node 9: InstExp > 9.73050E+03
Node 19: AppsAcc <= 7.56500E+02
Node 38: Mean cost = 4.67096E-01
Node 19: AppsAcc > 7.56500E+02
Node 39: Mean cost = 2.45984E-01
Node 2: PFacPhD > 78.50000
Node 5: InstExp <= 1.19335E+04
Node 10: InstExp <= 9.37050E+03
Node 20: RnBcost <= 4.28000E+03
Node 40: Mean cost = 3.88057E-01
Node 20: RnBcost > 4.28000E+03
Node 41: Mean cost = 1.08477E-01
Node 10: InstExp > 9.37050E+03
Node 21: Mean cost = 3.66662E-01
Node 5: InstExp > 1.19335E+04
Node 11: RnBcost <= 5.48500E+03
Node 22: NewEnrol <= 3.99500E+02
Node 44: Mean cost = 3.95475E-01
Node 22: NewEnrol > 3.99500E+02
Node 45: Mean cost = 5.96227E-01
Node 11: RnBcost > 5.48500E+03
Node 23: GradRate <= 80.50000
Node 46: Mean cost = 2.67244E-01
Node 23: GradRate > 80.50000
Node 47: Mean cost = 4.33452E-01
Node 1: PubPriv = Public
Node 3: FullPSal <= 5.76500E+02
Node 6: RnBcost <= 3.07650E+03
Node 12: Mean cost = 1.07147E-01
Node 6: RnBcost > 3.07650E+03
Node 13: Mean cost = 1.26985E-01
Node 3: FullPSal > 5.76500E+02
Node 7: Mean cost = 2.68167E-01

```

```

*****
In the following the predictor node mean is mean of complete cases

```

```

Node 1: Intermediate node
A case goes into Node 2 if PubPriv = Private
      PubPriv mode = Private
Estimated D values are:

```

```

7.9483E+03  9.4466E+03
-----
Node 2: Intermediate node
A case goes into Node 4 if PFacPhD <= 7.850000E+01
      PFacPhD mean = 6.8653E+01
Estimated D values are:
  1.1501E+04  1.1514E+04
-----
Node 4: Intermediate node
A case goes into Node 8 if RnBcost <= 4.275000E+03
      RnBcost mean = 4.2156E+03
Estimated D values are:
  9.8404E+03  9.8591E+03
-----
Node 8: Intermediate node
A case goes into Node 16 if RnBcost <= 3.588500E+03
      RnBcost mean = 3.5265E+03
Estimated D values are:
  8.6496E+03  8.6777E+03
-----
Node 16: Intermediate node
A case goes into Node 32 if AppsRec <= 7.865000E+02
      AppsRec mean = 8.9386E+02
Estimated D values are:
  7.5755E+03  7.6116E+03
-----
Node 32: Intermediate node
A case goes into Node 64 if InstExp <= 6.352000E+03
      InstExp mean = 7.2410E+03
Estimated D values are:
  7.9458E+03  7.9418E+03
-----
Node 64: Terminal node
Estimated D values are:
  6.5372E+03  6.5259E+03
-----
Node 65: Terminal node
Estimated D values are:
  8.7367E+03  8.7367E+03
-----
Node 33: Terminal node
Estimated D values are:
  6.9293E+03  7.0354E+03
-----
Node 17: Intermediate node
A case goes into Node 34 if GradRate <= 6.550000E+01

```

```

          GradRate mean = 5.9606E+01
Estimated D values are:
  9.7472E+03  9.7671E+03
-----
Node 34: Intermediate node
A case goes into Node 68 if InstExp <= 8.3390000E+03
          InstExp mean = 7.7413E+03
Estimated D values are:
  8.9787E+03  9.0105E+03
-----
Node 68: Terminal node
Estimated D values are:
  8.5398E+03  8.5831E+03
-----
Node 69: Terminal node
Estimated D values are:
  1.0181E+04  1.0181E+04
-----
Node 35: Terminal node
Estimated D values are:
  1.1043E+04  1.1043E+04
-----
Node 9: Intermediate node
A case goes into Node 18 if InstExp <= 9.7305000E+03
          InstExp mean = 9.1803E+03
Estimated D values are:
  1.1653E+04  1.1657E+04
-----
Node 18: Intermediate node
A case goes into Node 36 if InstExp <= 7.9485000E+03
          InstExp mean = 7.9069E+03
Estimated D values are:
  1.0812E+04  1.0818E+04
-----
Node 36: Terminal node
Estimated D values are:
  9.9945E+03  1.0006E+04
-----
Node 37: Terminal node
Estimated D values are:
  1.1654E+04  1.1654E+04
-----
Node 19: Intermediate node
A case goes into Node 38 if AppsAcc <= 7.5650000E+02
          AppsAcc mean = 1.2934E+03
Estimated D values are:

```

```

1.4000E+04  1.4000E+04
-----
Node 38: Terminal node
Estimated D values are:
  1.2582E+04  1.2582E+04
-----
Node 39: Terminal node
Estimated D values are:
  1.5542E+04  1.5542E+04
-----
Node 5: Intermediate node
A case goes into Node 10 if InstExp <=  1.1933500E+04
      InstExp mean =  1.4606E+04
Estimated D values are:
  1.4845E+04  1.4844E+04
-----
Node 10: Intermediate node
A case goes into Node 20 if InstExp <=  9.3705000E+03
      InstExp mean =  9.6199E+03
Estimated D values are:
  1.2263E+04  1.2262E+04
-----
Node 20: Intermediate node
A case goes into Node 40 if RnBcost <=  4.2800000E+03
      RnBcost mean =  4.4798E+03
Estimated D values are:
  1.0736E+04  1.0736E+04
-----
Node 40: Terminal node
Estimated D values are:
  9.7492E+03  9.7492E+03
-----
Node 41: Terminal node
Estimated D values are:
  1.1605E+04  1.1605E+04
-----
Node 21: Terminal node
Estimated D values are:
  1.3501E+04  1.3499E+04
-----
Node 11: Intermediate node
A case goes into Node 22 if RnBcost <=  5.4850000E+03
      RnBcost mean =  5.5490E+03
Estimated D values are:
  1.7048E+04  1.7048E+04
-----

```

```
Node 22: Intermediate node
A case goes into Node 44 if NewEnrol <= 3.9950000E+02
      NewEnrol mean = 4.7773E+02
Estimated D values are:
  1.5216E+04  1.5216E+04
-----
Node 44: Terminal node
Estimated D values are:
  1.4230E+04  1.4230E+04
-----
Node 45: Terminal node
Estimated D values are:
  1.6020E+04  1.6020E+04
-----
Node 23: Intermediate node
A case goes into Node 46 if GradRate <= 8.0500000E+01
      GradRate mean = 8.5822E+01
Estimated D values are:
  1.8261E+04  1.8261E+04
-----
Node 46: Terminal node
Estimated D values are:
  1.7077E+04  1.7077E+04
-----
Node 47: Terminal node
Estimated D values are:
  1.8762E+04  1.8762E+04
-----
Node 3: Intermediate node
A case goes into Node 6 if FullPSal <= 5.7650000E+02
      FullPSal mean = 5.4733E+02
Estimated D values are:
  2.1916E+03  6.1746E+03
-----
Node 6: Intermediate node
A case goes into Node 12 if RnBcost <= 3.0765000E+03
      RnBcost mean = 3.2503E+03
Estimated D values are:
  1.9678E+03  5.3794E+03
-----
Node 12: Terminal node
Estimated D values are:
  1.7086E+03  4.5104E+03
-----
Node 13: Terminal node
Estimated D values are:
```

```

2.1555E+03  6.0230E+03
-----
Node 7: Terminal node
Estimated D values are:
2.6314E+03  7.6500E+03
-----

```

LaTeX code for tree is in file: multi.tex

The resulting tree model is shown in Figure 9. The file `multifit.txt` collects together in tabular form the sample mean values of the D variables in the terminal nodes of the tree. The first column in the file gives the terminal node label and subsequent columns give the sample mean values of the D variables.

```

64  0.65372E+04  0.65259E+04
65  0.87367E+04  0.87367E+04
33  0.69293E+04  0.70354E+04
68  0.85398E+04  0.85831E+04
69  0.10181E+05  0.10181E+05
35  0.11043E+05  0.11043E+05
36  0.99945E+04  0.10006E+05
37  0.11654E+05  0.11654E+05
38  0.12582E+05  0.12582E+05
39  0.15542E+05  0.15542E+05
40  0.97492E+04  0.97492E+04
41  0.11605E+05  0.11605E+05
21  0.13501E+05  0.13499E+05
44  0.14230E+05  0.14230E+05
45  0.16020E+05  0.16020E+05
46  0.17077E+05  0.17077E+05
47  0.18762E+05  0.18762E+05
12  0.17086E+04  0.45104E+04
13  0.21555E+04  0.60230E+04
7   0.26314E+04  0.76500E+04

```

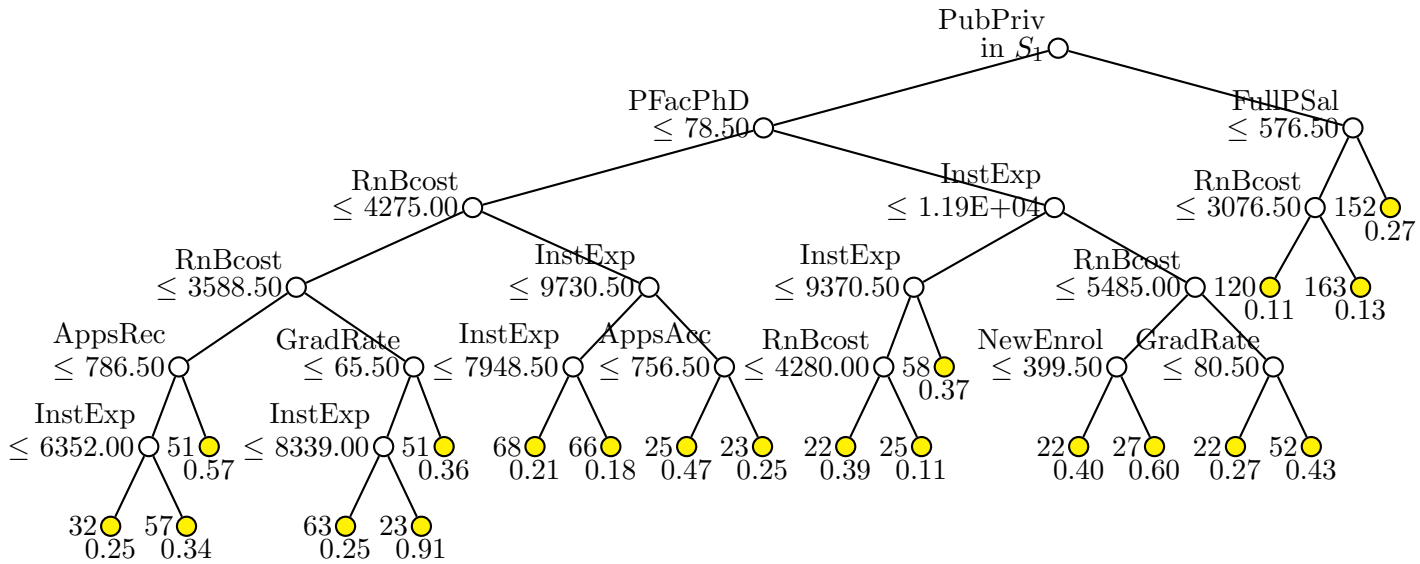


Figure 9: GUIDE regression tree model for multi-response data. At each intermediate node, a case goes to the left branch if and only if the condition is satisfied. Sample size on left and (normalized) MSE beneath each leaf node.

8 Other features

8.1 Pruning with test samples

GUIDE typically has three pruning options for deciding the size of the final tree: (i) cross-validation, (ii) test sample, and (iii) no pruning. Test-sample pruning is available only when there are no derived variables, such as creation of dummy indicator variables when ‘b’ variables are present. If test-sample pruning is chosen, the program will ask for the name of the file containing the test samples. This file must have the same column format as the training sample file.

8.2 Prediction of test samples

Often, we want to use a training sample to construct a classification or a regression tree and then use it to predict the **d**-variable values for a test sample. The latter may contain the true **d**-values or it may not, in which case the values are given the missing value code. GUIDE can do the construction and prediction in one step if the records in the two samples are placed into one data file and a weight variable is added that takes value 1 for the training cases and 0 for the test cases. GUIDE will ignore the cases with 0 weights during tree construction. A prompt towards the end of the program execution gives the option of writing the predicted value and leaf node membership of each record to a separate file.

8.3 Least median of squares, Poisson, and relative risk regression

GUIDE can also construct least median of squares ([Rousseeuw and Leroy, 1987](#)), Poisson, and relative risk regression tree models. These methods are selected at the same place in the interactive dialog as least squares regression. For relative risk regression, survival time is designated as a **t**-variable and the **d**-variable acts as a **death** or censoring indicator, taking value 1 for an uncensored (death) and 0 for a censored time survival time.

8.4 Unattended (batch) operation

GUIDE can be executed in unattended (batch) mode by choosing option 4 at the first prompt. The program then leads the user through a series of questions to gather information for a batch input file. The information includes the name of the batch

input file to hold the desired options and the name of the output file. For example, if the input file name is `input.txt`, the batch job can be started with the command:

```
guide < input.txt > log.txt
```

The file `log.txt` will contain program prompts and any error messages.

This mode can be used to carry out simulations or to run GUIDE repeatedly on bootstrapped samples to produce an ensemble of tree models. The steps for doing the latter are as:

1. Create a file (with file name `data.txt`, say) containing one set of bootstrapped data.
2. Create a data description file (with file name `desc.txt`, say) for GUIDE that refers to the data file name `data.txt`.
3. Use the batch option to create an input file (with file name `input.txt`, say) that points to the description file `desc.txt`.
4. Write a DOS batch program (Windows) or a shell script (Linux or Macintosh) that repeatedly:
 - (a) replaces the file `data.txt` with new bootstrapped samples;
 - (b) calls GUIDE with the command: `guide < input.txt > log.txt`;
 - (c) reads and processes the results from each GUIDE run.

8.5 Forests and tree ensembles

GUIDE has two methods of constructing an ensemble of trees. One is bagging, which fits GUIDE trees to bootstrap samples [Breiman \(1996\)](#). The other is similar to random forests [Breiman \(2001\)](#), which also uses bootstrap samples but randomly selects a small subset of variables for split selection at each node. If there are very many variables only a few of which are relevant, the first option tends to be more accurate. Here we demonstrate the second option.

Choose one of the following options:

1. Read the warranty disclaimer
2. Fit a model
3. Convert data to other formats
4. Create a batch input file
5. Rank and select regressor variables

```

Input your choice: 2
Input name of file to store results: forestout.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):2
Input 1 for bagging, 2 for rforest: ([1:2], <cr>=2):
You can fit a classification tree or a regression tree
Input 1 for classification, 2 for regression ([1:2], <cr>=1):2
Input 1 for interaction tests, 2 to skip them ([1:2], <cr>=2):

Input name of data description file (max 100 chars; enclose within
quotes if it contains spaces): bbdsc.txt
Reading data description file ...
Training sample file: bbdatt.txt
Missing value code: NA
Warning: N variables changed to S
Warning: B variables changed to C
Dependent variable is Logsalary
Length of longest data entry = 17
Total number of cases =      263
Cat. var. in column  #levels (incl. missing)  #missing values
                16                2                0
                17                2                0
                18               24                0
                19               23                0
                24                2                0
                25               24                0
Checking data ...
  Total #cases w/ #cases w/
  #cases  miss. D  miss. val  #X-var  #N-var  #F-var  #S-var  #B-var  #C-var
    263      0      0      3      0      0      16      0      6
No weight variable in data file
Number of cases used for training =      263
Default number of trees =      500
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Default number of variables used for splitting =      5
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Choose a variable selection method:
Choose 1 for unbiased interaction and curvature detection
Choose 2 for greedy but biased (RPART) search
Input 1 or 2 ([1:2], <cr>=1):
Choose a split point selection method for numerical variables:
Choose 1 to use faster method based on sample quantiles
Choose 2 to use exhaustive search
Input 1 or 2 ([1:2], <cr>=1):
Choose fraction of cases for splitting
Larger values give more splits: 0 = median split and 1 = all possible splits
Default fraction is 0.38023

```

```

Choose 1 to accept default split fraction, 2 to change it
Input 1 or 2 ([1:2], <cr>=1):
Default max number of split levels =          20
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Smallest possible node sample size =          5
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Input name of file to store predicted values: forestfit.txt
Constructing the trees ...
Finished iteration number:          50
Finished iteration number:          100
Finished iteration number:          150
Finished iteration number:          200
Finished iteration number:          250
Finished iteration number:          300
Finished iteration number:          350
Finished iteration number:          400
Finished iteration number:          450
Finished iteration number:          500
Results are stored in forestout.txt and forestfit.txt

```

Contents of forestout.txt

```

Random forest of piecewise-constant regression trees
Data description file: bbdsc.txt
Training sample file: bbdat.txt
Missing value code: NA
Warning: N variables changed to S
Warning: B variables changed to C
Dependent variable is Logsalary
Piecewise constant model
Length of longest data entry = 17

```

Summary information (without x variables)

d=dependent, b=split and fit cat variable using 0-1 dummies, c=split-only categorical, n=split and fit numerical, f=fit-only numerical, s=split-only numerical, w=weight

For categorical variables, #categories include one for missing values

Column number	Variable name	Variable type	Minimum value	Maximum value	Number of categories	Number missing
3	Bat86	s	1.2700E+02	6.8700E+02		
4	Hit86	s	3.2000E+01	2.3800E+02		
5	Hr86	s	0.0000E+00	4.0000E+01		
6	Run86	s	1.3000E+01	1.3000E+02		
7	Rb86	s	8.0000E+00	1.2100E+02		
8	Wlk86	s	3.0000E+00	1.0500E+02		
9	Yrs	s	1.0000E+00	2.4000E+01		

10	Batcr	s	1.8100E+02	1.4053E+04					
11	Hitcr	s	4.2000E+01	4.2560E+03					
12	Hrcr	s	0.0000E+00	5.4800E+02					
13	Runcr	s	1.8000E+01	2.1650E+03					
14	Rbcr	s	9.0000E+00	1.6590E+03					
15	Wlcr	s	8.0000E+00	1.5660E+03					
16	Leag86	c							2
17	Div86	c							2
18	Team86	c							24
19	Pos86	c							23
20	Puto86	s	0.0000E+00	1.3770E+03					
21	Asst86	s	0.0000E+00	4.9200E+02					
22	Err86	s	0.0000E+00	3.2000E+01					
24	Leag87	c							2
25	Team87	c							24
26	Logsalary	d	4.2121E+00	7.8079E+00					

Total #cases	#cases w/ miss. D	#cases w/ miss. val	#X-var	#N-var	#F-var	#S-var	#B-var	#C-var
263	0	0	3	0	0	16	0	6

No weight variable in data file

Number of cases used for training = 263

No interaction tests

Number of trees in forest = 500

Number of variables used for splitting = 5

Fraction of cases used for splitting each node = 0.38023

Max number of split levels = 20

Minimum node size = 5

Mean number of terminal nodes = 35.78

Resubstitution estimate of mean squared error = 5.933009650759087E-002

based on number of training cases = 263

Proportion of variance (R-squared) explained by ensemble model = 0.9237

Predicted values are in file forestfit.out

Following are the predicted values of the first seven cases from the forest model. They may be compared with the predicted values from the single tree model in Section 7.1.

```

6.24524171995602
6.24461424557850
6.63520239401012
4.62593215856375
6.61316897289852
4.37622175329919

```

4.61653842039342

8.6 Importance scoring and ranking of variables

GUIDE can rank the variables in order of their importance for predicting the dependent variable. The ranking also identifies any variables found to be unimportant. This is done by choosing option 5 at the first prompt. The following session log shows how to get a ranking of the variables for the baseball data using the description file `bbdsc.txt`.

```

Choose one of the following options:
1. Read the warranty disclaimer
2. Fit a model
3. Convert data to other formats
4. Create a batch input file
5. Rank and select regressor variables
Input your choice: 5
Input name of file to store results: ranks.txt
You can fit a classification tree or a regression tree
Input 1 for classification, 2 for regression ([1:2], <cr>=1):2
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards, 5=multiresponse,
6=longitudinal data (requires T variables).
Input choice ([1:6], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Input 1 for interaction tests, 2 to skip them ([1:2], <cr>=2):

Input name of data description file (max 100 chars; enclose within
quotes if it contains spaces): bbdsc.txt
Reading data description file ...
Training sample file: bbdat.txt
Missing value code: NA
Warning: N variables changed to S
Warning: B variables changed to C
Dependent variable is Logsalary
Length of longest data entry = 17
Total number of cases =          263
Cat. var. in column   #levels (incl. missing)   #missing values
           16                2                0
           17                2                0
           18               24                0
           19               23                0
           24                2                0
           25               24                0

Checking data ...

```

```

    Total #cases w/ #cases w/
    #cases miss. D miss. val #X-var #N-var #F-var #S-var #B-var #C-var
      263      0      0      3      0      0      16      0      6
No weight variable in data file
Number of cases used for training =          263
Default number of variables expected to be erroneously selected is          1
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Choose a split point selection method for numerical variables:
Choose 1 to use faster method based on sample quantiles
Choose 2 to use exhaustive search
Input 1 or 2 ([1:2], <cr>=1):
Choose fraction of cases for splitting
Larger values give more splits: 0 = median split and 1 = all possible splits
Default fraction is 0.38023E-02
Choose 1 to accept default split fraction, 2 to change it
Input 1 or 2 ([1:2], <cr>=1):
Do you have a file containing the max no. split levels, minimum node size,
frac cuts, and terminal node IDS?
Input 1 for no, 2 for yes ([1:2], <cr>=1):
Choose 2 if this is a follow-up to a prior run where this file is produced.
Default max number of split levels =          4
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Smallest possible node sample size =          5
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
You can create a description file with the selected variables included or excluded
Input 2 to create such a file, 1 otherwise ([1:2], <cr>=1):
This prompt gives the option of automatically writing another copy of the data
description file that has the unimportant variables excluded. We skip it here.
You can also output the importance scores and variable names to a file
Input 2 to create such a file, 1 otherwise ([1:2], <cr>=1):
Constructing main tree ...
Number of important splitting variables =          18
Number of unimportant splitting variables =          4
Results are stored in file ranks.txt

```

The following results from the bottom of the file `ranks.txt` give the scaled (so that the largest score is 100) and unscaled importance scores. A variable with an unscaled score greater than 1 is considered important.

```

Predictor variables sorted by importance scores
(F and R variables are excluded)
Importance Scores          Variable
Scaled      Unscaled      name      rank
100.0      1.37946E+01      Hitcr      1
 93.4      1.28831E+01      Batcr      2
 86.8      1.19753E+01      Runcr      3
 81.3      1.12139E+01      Rbcr      4

```

72.1	9.94407E+00	Yrs	5
67.5	9.31302E+00	Wlkr	6
42.7	5.88665E+00	Hit86	7
42.5	5.86403E+00	Hrcr	8
34.4	4.74177E+00	Bat86	9
34.2	4.72180E+00	Run86	10
33.1	4.56482E+00	Rb86	11
28.8	3.96731E+00	Wlk86	12
19.9	2.74980E+00	Hr86	13
12.0	1.65573E+00	Pos86	14
9.5	1.31128E+00	Puto86	15
7.7	1.05774E+00	Err86	16
7.6	1.04555E+00	Asst86	17
7.5	1.03847E+00	Team87	18
----- cut-off -----			
4.6	6.28312E-01	Leag87	19
4.2	5.81948E-01	Team86	20
4.0	5.58337E-01	Leag86	21
3.2	4.47781E-01	Div86	22

Number of important splitting variables = 18
 Number of unimportant splitting variables = 4

8.7 Automatic generation of powers and products

GUIDE allows the creation of certain powers and products of regressor variables on the fly. Specifically, variables of the form $X_1^p X_2^q$, where X_1 and X_2 are numerical predictor variables and p and q are integers, can be created by adding one or more lines of the form

```
0 i p j q a
```

at the end of the data description file. Here i and j are integers giving the column numbers of variables X_1 and X_2 , respectively, in the data file and a is one of the letters n , s , or f (corresponding to a numerical variable used for both splitting and fitting, splitting only, or fitting only).

To illustrate, suppose we wish to fit a piecewise quadratic model in the variable **Yrs** for the baseball data. This is easily done by adding one line to the file `bbdsc.txt`. First we assign the s (for splitting only) designator to every numerical predictor except **Yrs**. This will prevent all variables other than **Yrs** from acting as regressors in the piecewise quadratic models. To create the variable Yrs^2 , add the line

```
0 9 2 9 0 f
```

to the end of `bbdsc.txt`. The 9's in the above line refers to the column number of the variables `Yrs` in the data file, and the `f` tells the program to use the variable `Yrs2` for fitting leaf node models only. Note: The line defines `Yrs2` as `Yrs2 × Yrs0`. Since we can equivalently define the variable by `Yrs2 = Yrs1 × Yrs1`, we could also have used the line: “0 9 1 9 1 f”.

The resulting description file now looks like this:

```
bbdat.txt
NA
column, varname, vartype
1 Id x
2 Name x
3 Bat86 s
4 Hit86 s
5 Hr86 s
6 Run86 s
7 Rb86 s
8 Wlk86 s
9 Yrs n
10 Batcr s
11 Hitcr s
12 Hrcr s
13 Runcr s
14 Rbcr s
15 Wlkcr s
16 Leag86 c
17 Div86 c
18 Team86 c
19 Pos86 c
20 Puto86 s
21 Asst86 s
22 Err86 s
23 Salary x
24 Leag87 c
25 Team87 c
26 Logsalary d
0 9 2 9 0 f
```

When the program is given this description file, the output will show the regression coefficients of `Yrs` and `Yrs2` in each leaf node of the tree.

8.8 Data formatting functions

The program includes a utility function for reformatting data files into forms required by some statistical software packages:

1. R/Splus: Fields are space delimited. Missing values are coded as NA. Each record is written on one line. Variable names are given on the first line.
2. SAS: Fields are space delimited. Missing values are coded with periods. Character strings are truncated to eight characters. Spaces within character strings are replaced with underscores (`_`).
3. TEXT: Fields are comma delimited. Empty fields denote missing values. Character strings longer than eight characters are truncated. Each record is written on one line. Variable names are given on the first line.
4. STATISTICA: Fields are comma delimited. Commas in character strings are stripped. Empty fields denote missing values. Each record occupies one line.
5. SYSTAT: Fields are comma delimited. Strings are truncated to eight characters. Missing character values are replaced with spaces, missing numerical values with periods. Each record occupies one line.
6. BMDP: Fields are space delimited. Categorical values are sorted in alphabetic order and then assigned integer codes. Missing values are indicated by asterisks. Variable names longer than eight characters are truncated.
7. DataDesk: Fields are space delimited. Missing categorical values are coded with question marks. Missing numerical values are coded with asterisks. Each record is written on one line. Spaces within categorical values are replaced with underscores. Variable names are given on the first line of the file.
8. MINITAB: Fields are space delimited. Categorical values are sorted in alphabetic order and then assigned integer codes. Missing values are coded with asterisks. Variable names longer than eight characters are truncated.
9. NUMBERS: Same as **TEXT** option except that categorical values are converted to integer codes.
10. C4.5: This is the format required by the C4.5 (Quinlan, 1993) program.
11. ARFF: This is the format required by the WEKA (Witten and Frank, 2000) programs.

Following is a sample session where the iris data are reformatted for R or Splus.

Choose one of the following options:

1. Read the warranty disclaimer
2. Fit a model
3. Convert data to other formats
4. Create a batch input file
5. Rank and select regressor variables (experimental)

Input your choice: 3

Input name of log file: log.txt

Input 1 if D variable is categorical, 2 if real, 0 if none ([0:2], <cr>=1):

Input name of data description file (max 100 chars; enclose within quotes if it contains spaces): irisdesc.txt

Training sample file is: irisdata.txt

Missing value code is: ?

Reading data description file ...

Length of longest data entry = 11

Number of classes = 3

Choose one of the following data formats:

No.	Name	Field Separ	Miss. char.	val. numer.	codes Remarks
1	Spplus/R	space	NA	NA	1 line/case, var names on 1st line
2	SAS	space	.	.	strings trunc., spaces -> '_'
3	TEXT	comma	empty	empty	1 line/case, var names on 1st line
4	STATISTICA	comma	empty	empty	1 line/case, commas stripped var names on 1st line
5	SYSTAT	comma	space	.	1 line/case, var names on 1st line strings trunc. to 8 chars
6	BMDP	space		*	strings trunc. to 8 chars cat values -> integers (alph. order)
7	DATADESK	space	?	*	1 line/case, var names on 1st line spaces -> '_'
8	MINITAB	space		*	cat values -> integers (alph. order) var names trunc. to 8 chars
9	NUMBERS	comma	NA	NA	1 line/case, var names on 1st line cat values -> integers (alph. order)
10	C4.5	comma	?	?	1 line/case, dependent variable last
11	ARFF	comma	?	?	1 line/case

0 abort this job

Input your choice ([0:11], <cr>=3):1

Input name of new data file: iris.rdata

Follow the commented lines in "iris.rdata" to read the data into R or Spplus

References

- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24:123–140.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.
- Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont.
- Chan, K.-Y. and Loh, W.-Y. (2004). LOTUS: An algorithm for building accurate and comprehensible logistic regression trees. *Journal of Computational and Graphical Statistics*, 13:826–852.
- Chaudhuri, P. and Loh, W.-Y. (2002). Nonparametric estimation of conditional quantiles using quantile regression trees. *Bernoulli*, 8:561–576. <http://www.stat.wisc.edu/~loh/treeprogs/guide/quantile.pdf>.
- Fitzmaurice, G. M., Laird, N. M., and Ware, J. H. (2004). *Applied Longitudinal Analysis*. Wiley.
- Kim, H. and Loh, W.-Y. (2001). Classification trees with unbiased multiway splits. *Journal of the American Statistical Association*, 96:589–604. <http://www.stat.wisc.edu/~loh/treeprogs/cruise/cruise.pdf>.
- Kim, H. and Loh, W.-Y. (2003). Classification trees with bivariate linear discriminant node models. *Journal of Computational and Graphical Statistics*, 12:512–530. <http://www.stat.wisc.edu/~loh/treeprogs/cruise/jcgs.pdf>.
- Kim, H., Loh, W.-Y., Shih, Y.-S., and Chaudhuri, P. (2007). Visualizable and interpretable regression models with good prediction power. *IIE Transactions*, 39:565–579. <http://www.stat.wisc.edu/~loh/treeprogs/guide/iie.pdf>.
- Koenker, R. W. and Bassett, G. (1978). Regression quantiles. *Econometrica*, 46:33–50.
- Loh, W.-Y. (2002). Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, 12:361–386. <http://www3.stat.sinica.edu.tw/statistica/j12n2/j12n21/j12n21.htm>.
- Loh, W.-Y. (2006a). Logistic regression tree analysis. In Pham, H., editor, *Handbook of Engineering Statistics*, pages 537–549. Springer.

- Loh, W.-Y. (2006b). Regression tree models for designed experiments. In Rojo, J., editor, *The Second Erich L. Lehmann Symposium—Optimality*, volume 49, pages 210–228. Institute of Mathematical Statistics Lecture Notes-Monograph Series. <http://arxiv.org/abs/math.ST/0611192>.
- Loh, W.-Y. (2008a). Classification and regression tree methods. In Ruggeri, F., Kenett, R., and Faltin, F. W., editors, *Encyclopedia of Statistics in Quality and Reliability*, pages 315–323. Wiley, Chichester, UK. <http://www.stat.wisc.edu/~loh/treeprogs/guide/eqr.pdf>.
- Loh, W.-Y. (2008b). Regression by parts: Fitting visually interpretable models with GUIDE. In Chen, C., Härdle, W., and Unwin, A., editors, *Handbook of Computational Statistics*, pages 447–469. Springer. <http://www.stat.wisc.edu/~loh/treeprogs/guide/handbk.pdf>.
- Loh, W.-Y. (2009). Improving the precision of classification trees. *Annals of Applied Statistics*, 3:1710–1737. <http://www.stat.wisc.edu/~loh/treeprogs/guide/aoas260.pdf>.
- Loh, W.-Y. (2010). Tree-structured classifier. *Wiley Interdisciplinary Reviews: Computational Statistics*. In press.
- Loh, W.-Y., Chen, C.-W., and Zheng, W. (2007). Extrapolation errors in linear model trees. *ACM Trans. Knowl. Discov. Data*, 1(2):6. <http://www.stat.wisc.edu/~loh/treeprogs/guide/acm.pdf>.
- Loh, W.-Y. and Shih, Y.-S. (1997). Split selection methods for classification trees. *Statistica Sinica*, 7:815–840. <http://www3.stat.sinica.edu.tw/statistica/j7n4/j7n41/j7n41.htm>.
- Quinlan, J. R. (1992). Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, pages 343–348.
- Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
- Rousseeuw, P. J. and Leroy, A. M. (1987). *Robust Regression and Outlier Detection*. Wiley.
- Witten, I. and Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations*. Morgan Kaufmann, San Francisco, CA. <http://www.cs.waikato.ac.nz/ml/weka>.