

GUIDE User Manual*

(updated for version 29.6)

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

October 18, 2018

Contents

1	Warranty disclaimer	4
2	Introduction	5
2.1	Installation	6
2.2	L ^A T _E X	9
3	Program operation	10
3.1	Required files	10
3.2	Input file creation	13
4	Classification	14
4.1	Univariate splits, ordinal predictors: glaucoma data	14
4.1.1	Input file generation	14
4.1.2	Contents of <code>glaucoma.in</code>	16
4.1.3	Executing the program	17
4.1.4	Interpreting the output file	20
4.2	Linear splits: glaucoma data	28

*Based on work partially supported by grants from the U.S. Army Research Office, National Science Foundation, National Institutes of Health, Bureau of Labor Statistics, and Eli Lilly & Co. Work on precursors to GUIDE additionally supported by IBM and Pfizer.

4.3	Univariate splits, categorical predictors: peptide data	32
4.3.1	Input file generation	32
4.3.2	Results	34
4.4	Unbalanced classes and equal priors: hepatitis data	37
4.5	Unequal misclassification costs: hepatitis data	42
4.6	More than 2 classes: dermatology	42
4.6.1	Default option	43
4.6.2	Nearest-neighbor option	51
4.6.3	Kernel density option	62
4.7	More than 2 classes: heart disease	70
4.7.1	Input file creation	71
4.7.2	Results	75
4.7.3	RPART model	92
5	Regression	94
5.1	Least squares constant: birthwt data	94
5.1.1	Input file creation	94
5.1.2	Results	97
5.1.3	Contents of <code>cons.var</code>	111
5.2	Least squares simple linear: birthwt data	112
5.2.1	Input file creation	112
5.2.2	Results	115
5.2.3	Contents of <code>lin.var</code>	119
5.2.4	Contents of <code>lin.reg</code>	121
5.3	Multiple linear: birthwt data	121
5.3.1	Input file creation	121
5.3.2	Results	124
5.3.3	Contents of <code>mul.var</code>	128
5.3.4	Contents of <code>mul.reg</code>	130
5.4	Stepwise linear: birthwt data	130
5.4.1	Input file creation	130
5.4.2	Contents of <code>step.reg</code>	133
5.5	Best ANCOVA: birthwt data	133
5.5.1	Input file creation	134
5.5.2	Contents of output file	137
5.5.3	Contents of <code>ancova.reg</code>	140
5.6	Quantile regression: birthwt data	140
5.6.1	Piecewise constant: 1 quantile	140

5.6.2	Input file creation	141
5.6.3	Results	142
5.6.4	Piecewise constant: 2 quantiles	148
5.6.5	Input file creation	148
5.6.6	Results	151
5.6.7	Piecewise simple linear	160
5.6.8	Input file creation	160
5.6.9	Results	161
5.6.10	Piecewise multiple linear	166
5.6.11	Input file creation	166
5.6.12	Results	167
5.7	Least median of squares: birthwt data	171
5.7.1	Results	173
5.8	Poisson regression with offset: lung cancer data	179
5.8.1	Input file creation	179
5.8.2	Results	181
5.9	Censored response: heart attack data	185
5.9.1	Results	187
5.10	Multi-response: public health data	195
5.10.1	Input file creation	195
5.10.2	Results	198
5.11	Longitudinal response with varying time: wage data	202
5.11.1	Input file creation	205
5.11.2	Results	208
5.12	Multiple longitudinal series: mother and child health	215
5.12.1	Input file creation	215
5.12.2	Results	218
5.13	Subgroup identification: breast cancer	225
5.13.1	Without linear prognostic control	225
5.13.2	Simple linear prognostic control	234
5.13.3	Multiple linear prognostic control	239
6	Multiple missing value codes: CE survey	244
6.1	Classification	245
6.1.1	Input file creation	245
6.1.2	Results	250
6.2	Regression	278

7	Periodic variables: NHTSA crash tests	280
8	Importance scoring	283
8.1	Classification: glaucoma data	285
8.1.1	Input file creation	285
8.1.2	Contents of <code>imp.out</code>	286
8.2	Regression with censoring: heart attack data	289
9	Differential item functioning: GDS data	292
10	Tree ensembles	296
10.1	GUIDE forest: hepatitis data	296
10.2	Input file creation	296
10.3	Results	298
10.4	Bagged GUIDE	300
11	Other features	304
11.1	Pruning with test samples	304
11.2	Prediction of test samples	304
11.3	GUIDE in R and in simulations	305
11.4	Generation of powers and products	306
11.5	Data formatting functions	307

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 an algorithm for construction of classification and regression trees and forests. It is a descendent of the FACT (Loh and Vanichsetakul, 1988), SUPPORT (Chaudhuri et al., 1994, 1995), QUEST (Loh and Shih, 1997), CRUISE (Kim and Loh, 2001, 2003), and LOTUS (Chan and Loh, 2004; Loh, 2006a) algorithms. GUIDE is the only classification and regression tree algorithm with all these features:

1. Unbiased variable selection with and without missing data.
2. Automatic handling of missing values without requiring prior imputation.
3. One or more missing value codes.
4. Kernel and nearest-neighbor node models for classification trees.
5. Weighted least squares, least median of squares, quantile, Poisson, and relative risk (proportional hazards) regression models.
6. Univariate, multivariate, censored, and longitudinal response variables.
7. Pairwise interaction detection at each node.
8. Linear splits on two variables at a time for classification trees.
9. Categorical variables for splitting only, fitting only (via 0-1 dummy variables), or both in regression tree models.
10. Periodic variables, such as angles, hour of day, day of week, month of year, seasons.
11. Importance scoring and thresholding of predictor variables.

12. Tree ensembles (bagging and forests).
13. Subgroup identification for differential treatment effects.

Tables 1 and 2 compare the features of GUIDE with QUEST, CRUISE, C4.5 (Quinlan, 1993), RPART (Therneau et al., 2017)¹, and M5' (Quinlan, 1992; Witten and Frank, 2000).

The GUIDE algorithm is documented in Loh (2002) for regression trees and Loh (2009) for classification trees. Reviews of the subject may be found in Loh (2008a), Loh (2011) and Loh (2014). Some advanced features of the algorithm are reported in Chaudhuri and Loh (2002), Loh (2006b), Kim et al. (2007), Loh et al. (2007), and Loh (2008b). A list of third-party applications of GUIDE, CRUISE, QUEST, and LOTUS is maintained in <http://www.stat.wisc.edu/~loh/apps.html>. This manual illustrates the use of the GUIDE software and the interpretation of the output.

2.1 Installation

GUIDE is available free from www.stat.wisc.edu/~loh/guide.html in the form of compiled 32- and 64-bit executables for Linux, Mac OS X, and Windows on Intel and compatible processors. Data and description files used in this manual are in the zip file www.stat.wisc.edu/~loh/treeprogs/guide/datafiles.zip.

Linux: There are three 64-bit executables to choose from: **Intel**, **NAG**, and **Gfortran**. The Intel version is best for Intel processors and the NAG version for AMD processors. The Gfortran version is compiled under Ubuntu 16.0. If necessary, make the unzipped file executable by issuing the command “`chmod a+x guide`” in a **Terminal** window.

Mac OS X: There are two executables to choose from. Make the unzipped file executable by issuing this command in a **Terminal** application in the folder where the file is located: `chmod a+x guide`

NAG. This version may be the fastest. It requires no additional software besides file the `guide.gz`.

Gfortran. This version requires **Xcode** and **gfortran** to be installed. To ensure that the gfortran libraries are placed in the right place, follow these steps:

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
Periodic (cyclic) variables	Yes	No	No	No	No
Node models	S, K, N	S	S, L	S	S
Missing values	Novel	Imputation	Surrogate	Surrogate	Weights
Missing-value flag variables	Yes	No	No	No	No
Tree diagrams	Text and L ^A T _E X			Proprietary	Text
Bagging	Yes	No	No	No	No
Forests	Yes	No	No	No	No

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 stepwise
Linear models	Multiple or stepwise (forward and forward-backward)	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
Periodic (cyclic) variables	Yes	No	No
Tree diagrams	Text and L ^A T _E X	Proprietary	PostScript
Operation modes	Batch	Interactive and batch	Interactive
Case weights	Yes	Yes	No
Transformations	Powers and products	No	No
Missing values in split variables	Missing values treated as special categories	Surrogate splits	Imputation
Missing values in linear predictors	Choice of separate constant models or mean imputation	N/A	Imputation
Missing-value flag variables	Yes	No	No
Bagging & forests	Yes & yes	No & no	No & no
Subgroup identification	Yes	No	No
Data conversions	ARFF, C4.5, Minitab, R, SAS, Statistica, Systat, CSV	No	No

1. Install **Xcode** from <https://developer.apple.com/xcode/downloads/>.
2. Go to <http://hpc.sourceforge.net> and download file `gcc-8.1-bin.tar.gz` to your Downloads folder. The direct link to the file is <http://prdownloads.sourceforge.net/hpc/gcc-8.1-bin.tar.gz?download>
3. Open a **Terminal** window and type (or copy and paste):
 - (a) `cd ~/Downloads`
 - (b) `gunzip gcc-8.1-bin.tar.gz`
 - (c) `sudo tar -xvf gcc-8.1-bin.tar -C /`

Windows: There are three executables to choose from: **Intel** (64 or 32 bit) and **Gfortran** (64 bit). The 32-bit executable may run a bit faster but the 64-bit versions can handle larger arrays. Download the 32 or 64-bit executable `guide.zip` and unzip it (right-click on file icon and select “Extract all”). The resulting file `guide.exe` may be placed in one of three places:

1. top level of your **C:** drive (where it can be invoked by typing `C:\guide` in a terminal window—see Section 3.1),
2. a folder that contains your data files, or
3. a folder on your search path.

2.2 L^AT_EX

GUIDE uses the public-domain software L^AT_EX (<http://www.ctan.org>) to produce tree diagrams. The L^AT_EX software may be obtained from:

Linux: TeX Live <http://www.tug.org/texlive/>

Mac: MacTeX <http://tug.org/mactex/> or

MikTeX <https://miktex.org/howto/install-miktex-mac>

Windows: MikTeX <https://miktex.org/howto/install-miktex> or
proTeXt <http://www.tug.org/protext/>

After L^AT_EX is installed, a pdf file of a L^AT_EX file, called `diagram.tex` say, produced by GUIDE can be obtained by typing the following three commands in a **Terminal** (Linux or Mac) or **Command** (Win) window. (**Important:** Do not use the menu commands of the L^AT_EX GUI to compile the L^AT_EX files, because they tend to invoke the **pdflatex** compiler by default, instead of the **latex** compiler.)

1. `latex diagram`
2. `dvips diagram`
3. `ps2pdf diagram.ps`

The first command produces a file called `diagram.dvi` which the second command uses to create a postscript file called `diagram.ps`. The latter can be viewed and printed if a postscript viewer (such as *Preview* for the Mac) is installed. If no postscript viewer is available, the last command can be used to convert the postscript file into a pdf file, which can be viewed and printed with *Adobe Reader*. The file `diagram.tex` can be edited to change colors, node sizes, etc. See the **pstricks** manual <http://tug.org/PSTricks/main.cgi/>.

Windows users: Convert the postscript figure to *Enhanced-format Meta File* (emf) format for use in Windows applications such as Word or PowerPoint. There are many conversion programs available on the web, such as *Graphic Converter* (<http://www.graphic-converter.net/>) and *pstoedit* (<http://www.pstoedit.net/>).

3 Program operation

3.1 Required 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 time variables. Entries in each record are comma, space, or tab delimited (multiple spaces are treated as one space, but not for commas). A record can occupy more than one line in the file, but each record must begin on a new line.

Values of categorical variables can contain any ascii character except single and double quotation marks, which are used to enclose values that contain spaces and commas. Values can be up to 60 characters long. Class labels are truncated to 10 characters in tabular displays.

A common problem among first-time users is getting the data file in proper shape. If the data are in a spreadsheet and there are **no empty cells**, export them to a **MS-DOS Comma Separated** (csv) file (the MS-DOS CSV format takes care of carriage return and line feed characters properly). If there are empty cells, a good solution is to read the spreadsheet into R (using `read.csv`

with proper specification of the `na.strings` argument), verify that the data are correctly read, and then export them to a text file using either `write.table` or `write.csv`.

Description file: This provides information about the name and location of the data file, names and column positions of the variables, and their roles in the analysis. Different models may be fitted by changing the roles of the variables. We demonstrate with the text files `glaucomadata.txt` and `glaucoma.dsc` — from www.stat.wisc.edu/~loh/treeprogs/guide/datafiles.zip or from the R package `ipred` (Peters and Hothorn, 2015)). The data give the values of 66 variables obtained from a laser scan image of the optic nerve for 85 normal people and 85 people with glaucoma. The response variable is `Class` (“normal” or “glaucoma”). The top and bottom lines of the file `glaucoma.dsc` are:

```
glaucomadata.txt
NA
2
1 ag n
2 at n
3 as n
4 an n
5 ai n
:
63 tension n
64 clv n
65 cs n
66 lora n
67 Class d
```

The 1st line gives the name of the data file. If the latter is not in the current folder, gives its full path (e.g., “c:\data\glaucomadata.txt”) surrounded by quotes (because it contains backslashes). The 2nd line gives the missing value code, which can be up to 80 characters long. If it contains non-alphanumeric characters, it too 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 which case any character string not present among the data values can be used). The 3rd line gives the line number of the first data record in the data file. Because `glaucomadata.txt` has the variable names in the first row, a “2” is placed on the third line of `glaucoma.dsc`. Blank lines in

the data and description files are ignored. The position, name and role of each variable comes next (in that order), with one line for each variable.

Variable names must begin with an alphabet and be not more than 60 characters long. If a name contains non-alphanumeric characters, it must be enclosed in matching single or double quotes. Spaces and the four characters #, %, {, and } are replaced by dots (periods) if they appear in a name. Variable names are truncated to 10 characters in tabular output. Leading and trailing spaces are dropped.

The following roles for the variables are permitted. Lower and upper case letters are accepted.

- b** Categorical variable 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** type for classification.
- c** Categorical variable used for splitting only.
- d** Dependent variable or death indicator variable. Except for longitudinal and multiple response data (Sec. 5.10), there can only be one **d** variable. For proportional hazards models, it is the event (death) indicator. For all other models, it is the response variable. It 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.
- i** Categorical variable to be converted to 0-1 indicator variables for fitting node models.
- m** Missing value flag variable. Each such variable should follow immediately after an **n**, **p** or **s** variable in the description file.
- n** Numerical variable used both for splitting the nodes and for fitting the node models. It is converted to type **s** in classification.
- p** Periodic (cyclic) variable, such as an angular measurement, hour of day, day of week, or month of year.
- r** Categorical treatment (Rx) variable used only for fitting the linear models in the nodes of the tree. It is not used for splitting the nodes. If this variable is present, all **n** variables are automatically changed to **s**.
- 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.

Table 3: Predictor variable role descriptors

Variable	Split nodes	Fit node models	Both
Categorical	c	i	b
Numerical	s	f	n

- t** Time variable, either time to event for proportional hazards models or observation time for longitudinal models.
- w** Weight variable for weighted least squares regression or for excluding observations in the training sample from tree construction. See section 11.2 for the latter. Except for longitudinal models, a record with a missing value in a **d**, **t**, or **z**-variable is automatically assigned zero weight.
- x** Excluded variable. This allows models to be fitted to different subsets of the variables without reformatting the data file.
- z** Offset variable used only in Poisson regression.

Table 3 summarizes the possible roles for predictor variables.

GUIDE runs within a **terminal window** of the computer operating system.

Do not double-click its icon on the desktop!

Linux. Any terminal program will do.

Mac OS X. The program is called **Terminal**; it is in the **Applications Folder**.

Windows. The terminal program is started from the **Start button** by choosing **All Programs → Accessories → Command Prompt**

After the terminal window is opened, change to the folder where the data and program files are stored. For Windows users who do not know how to do this, read <http://www.digitalcitizen.life/command-prompt-how-use-basic-commands>.

3.2 Input file creation

GUIDE is started by typing its (lowercase) name in a terminal. The preferred way is to create an input file (option 1 below) for subsequent execution. The input file may be edited if you wish to change some input parameters later. In the following, the sign (**>**) is the terminal prompt (not to be typed!).

```
> guide
GUIDE Classification and Regression Trees and Forests
Version 29.6 (Build date: October 18, 2018)
Compiled with GFortran 6.2.0 on Mac OS X High Sierra 10.13.5
Copyright (c) 1997-2018 Wei-Yin Loh. All rights reserved.
This software is based upon work supported by the U.S. Army Research Office,
the National Science Foundation and the National Institutes of Health.

Choose one of the following options:
0. Read the warranty disclaimer
1. Create an input file for model fitting or importance scoring (recommended)
2. Convert data to other formats without creating input file
Input your choice:
```

The meanings of these options are:

0. Print the warranty disclaimer.
1. Create an input file for model fitting or importance scoring (recommended).
2. Convert the data file into a format suitable for importation into database, spreadsheet, or statistics software. See Table 2 for the statistical packages supported. Section 11.5 has an example.

4 Classification

4.1 Univariate splits, ordinal predictors: glaucoma data

We first show how to generate an input file to produce a classification tree from the data in the file `glaucomadata.txt`, using the default options. 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. Annotations are printed in *blue italics* in this manual.

4.1.1 Input file generation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: glaucoma.in
This file will store your answers to the prompts.
```

Input 1 for model fitting, 2 for importance or DIF scoring,
3 for data conversion ([1:3], <cr>=1):
Press the ENTER or RETURN key to accept the default selection.
Name of batch output file: glaucoma.out
This file will contain the results when you apply the input file to GUIDE later.
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Option 2 is for bagging and random forest-type methods.
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
The default option will produce a traditional classification tree.
Choose option 2 for more advanced features.
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: glaucoma.dsc
Reading data description file ...
Training sample file: glaucomadata.txt
The name of the data set is read from the description file.
Some information about the data are printed in the next few lines.
Missing value code: NA
Records in data file start on line 2
Warning: N variables changed to S
This warning is due to N variables being always used as S in classification.
Dependent variable is Class
Reading data file ...
Number of records in data file: 170
Length of longest entry in data file: 8
Checking for missing values ...
Total number of cases: 170
Number of classes: 2
Re-checking data ...
Allocating missing value information
Assigning codes to categorical and missing values
Data checks complete
Creating missing value indicators
Rereading data

Class	#Cases	Proportion
glaucoma	85	0.50000000
normal	85	0.50000000

Total	#cases w/	#missing					
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var	
170	0	17	0	0	0		66
#M-var	#B-var	#C-var					
0	0	0					

No. cases used for training: 170
No. cases excluded due to 0 weight or missing D: 0

Finished reading data file
Choose 1 for estimated priors, 2 for equal priors, 3 for priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
See other parts of manual for examples of equal and specified priors.
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Choose option 2 if you do not want LaTeX code.
Input file name to store LaTeX code (use .tex as suffix): glaucoma.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: glaucoma.fit
This file will contain the node number and predicted class for each observation.
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: glaucomapred.r
This file will contain an R function for prediction.
Input file is created!
Run GUIDE with the command: guide < glaucoma.in
Press ENTER or RETURN to quit

4.1.2 Contents of glaucoma.in

Here are the contents of the input file:

```
GUIDE      (do not edit this file unless you know what you are doing)
 29.6      (version of GUIDE that generated this file)
 1         (1=model fitting, 2=importance or DIF scoring, 3=data conversion)
"glaucoma.out" (name of output file)
 1         (1=one tree, 2=ensemble)
 1         (1=classification, 2=regression, 3=propensity score grouping)
 1         (1=simple model, 2=nearest-neighbor, 3=kernel)
 1         (0=linear 1st, 1=univariate 1st, 2=skip linear, 3=skip linear and interaction)
 1         (1=prune by CV, 2=by test sample, 3=no pruning)
"glaucoma.dsc" (name of data description file)
    10     (number of cross-validations)
 1         (1=mean-based CV tree, 2=median-based CV tree)
    0.500  (SE number for pruning)
 1         (1=estimated priors, 2=equal priors, 3=other priors)
 1         (1=unit misclassification costs, 2=other)
 2         (1=split point from quantiles, 2=use exhaustive search)
 1         (1=default max. number of split levels, 2=specify no. in next line)
 1         (1=default min. node size, 2=specify min. value in next line)
 1         (1=write latex, 2=skip latex)
"glaucoma.tex" (latex file name)
 1         (1=include node numbers, 2=exclude)
 1         (1=number all nodes, 2=only terminal nodes)
```



```
1          (1=color terminal nodes, 2=no colors)
1          (0=#errors, 1=class sizes in nodes, 2=nothing)
1          (1=no storage, 2=store fit and split variables, 3=store split variables and values)
2          (1=do not save fitted values and node IDs, 2=save in a file)
"glaucoma.fit" (file name for fitted values and node IDs)
2          (1=do not write R function, 2=write R function)
"glaucomapred.r" (R code file)
```

GUIDE reads only the first item in each line; the rest of the line is a comment for human consumption. It is generally not advisable for the user to edit this file because each question depends on the answers given to previous questions.

4.1.3 Executing the program

After the input file is generated, GUIDE can be executed by typing the command “guide < glaucoma.in” at the screen prompt:

```
guide < glaucoma.in
```

This produces the following output to the screen. The alternative command “guide < glaucoma.in > log.txt” sends the screen output to the file log.txt.

```
GUIDE Classification and Regression Trees and Forests
Version 29.6 (Build date: October 18, 2018)
Compiled with GFortran 6.2.0 on Mac OS X High Sierra 10.13.5
Copyright (c) 1997-2018 Wei-Yin Loh. All rights reserved.
This software is based upon work supported by the U.S. Army Research Office,
the National Science Foundation and the National Institutes of Health.

Choose one of the following options:
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: Batch run with input file
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion: 1
Output file is glaucoma.out

Input 1 for single tree, 2 for ensemble of trees: 1
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice: 1
Input 1 for simple, 2 for nearest-neighbor, 3 for kernel method: 1
Input 0 for linear, interaction and univariate splits (in this order),
    1 for univariate, linear and interaction splits (in this order),
```

```

    2 to skip linear splits,
    3 to skip linear and interaction splits: 1
Input 1 to prune by CV, 2 by test sample, 3 for no pruning: 1

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: glaucoma.dsc
Reading data description file ...
Training sample file: glaucomadata.txt
Missing value code: NA
Records in data file start on line 2
Warning: N variables changed to S
Dependent variable is Class
Reading data file ...
Number of records in data file: 170
Length of longest entry in data file: 8
Checking for missing values ...
Total number of cases: 170
Number of classes: 2
Re-checking data ...
Allocating missing value information
Assigning codes to categorical and missing values
Data checks complete
Creating missing value indicators
Rereading data
  Class  #Cases    Proportion
glaucoma     85    0.50000000
normal       85    0.50000000
  Total  #cases w/  #missing
  #cases  miss. D  ord. vals  #X-var  #N-var  #F-var  #S-var
    170         0      17      0      0      0      66
  #M-var  #B-var  #C-var
    0      0      0
No. cases used for training: 170
No. cases excluded due to 0 weight or missing D: 0
Finished reading data file
Univariate split highest priority
Interaction and linear splits 2nd and 3rd priorities
Input number of cross-validations: 10
Selected tree is based on mean of CV estimates
Input number of SEs for pruning: 0.500000000000000000
Choose 1 for estimated priors, 2 for equal priors, 3 for priors from a file
Input 1, 2, or 3: 1
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2: 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: 2
Max. number of split levels: 10
Input 1 for default min. node size,
2 to specify min. value: 1
Min. node size: 2
Input 1 for LaTeX tree code, 2 to skip it: 1
Input file name to store LaTeX code: glaucoma.tex
Warning: LaTeX file is overwritten
Input 1 to include node numbers, 2 to omit them: 1
Input 1 to number all nodes, 2 to number leaves only: 1
Input 0 for #errors, 1 for class proportions, 2 for nothing: 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
Input 2 to save fitted values and node IDs; 1 otherwise: 2
File name is glaucoma.fit
Warning: file is overwritten
Input 2 to write R function for predicting new cases, 1 otherwise: 2
File name for R code: glaucomapred.r
Warning: R file is overwritten
Constructing main tree ...
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              10
    1               8
   2**              5
    3               3
    4               2
```

```
5
1
0-SE tree based on mean is marked with * and has 5 terminal nodes
* tree, ** tree, + tree, and ++ tree all the same

Writing predicted values...
...completed
Results are stored in glaucoma.out
Observed and fitted values are stored in glaucoma.fit
LaTeX code for tree is in glaucoma.tex
R code is stored in glaucomapred.r
```

The final pruned tree is marked with two asterisks (**); it has 5 terminal nodes.

4.1.4 Interpreting the output file

Following is an annotated copy of the contents of the output file.

```
Classification tree
Pruning by cross-validation
Data description file: glaucoma.dsc
Training sample file: glaucomadata.txt
Missing value code: NA
Records in data file start on line 2
This says that the first record begins on line 2 of the data file.
Warning: N variables changed to S
This warning is triggered if classification is chosen and there are predictor
variables designated as 'N'.
Dependent variable is Class
Number of records in data file: 170
Length of longest entry in data file: 8
Number of classes: 2
Training sample class proportions of D variable Class:
  Class #Cases Proportion
glaucoma      85  0.50000000
normal        85  0.50000000
This gives the number of observations in each class.
Summary information (without x variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight

      Codes/
      Levels/
Column Name      Minimum Maximum Periods #Missing
  1 ag          s    1.122    4.611
```

2	at	s	0.1760	0.9240
3	as	s	0.3080	1.173
4	an	s	0.3450	1.564
5	ai	s	0.2940	1.125
6	eag	s	0.4150	3.948
7	eat	s	0.1370	0.8480
8	eas	s	0.4300E-01	1.061
9	ean	s	0.8000E-02	1.266
10	eai	s	0.9800E-01	0.9610
11	abrg	s	0.3000E-02	3.894
12	abrt	s	0.3000E-02	0.8270
13	abrs	s	0.000	0.9010
14	abrn	s	0.000	1.268
15	abri	s	0.000	0.9150
16	hic	s	-0.1890	0.8870
17	mhcg	s	-0.1470	0.3220
18	mhct	s	-0.4700E-01	0.4770
19	mhcs	s	-0.1720	0.2930
20	mhc n	s	-0.2120	0.3850
21	mhci	s	-0.1610	0.4540
22	phcg	s	-0.2860	0.1450
23	phct	s	-0.1210	0.4020
24	phcs	s	-0.2470	0.1600
25	phcn	s	-0.2850	0.2170
26	phci	s	-0.2860	0.3710
27	hvc	s	0.1100	0.7150
28	vbsg	s	0.2000E-01	2.077
29	vbst	s	0.7000E-02	0.4460
30	vbss	s	0.2000E-02	0.5540
31	vbsn	s	0.000	0.6960
32	vbsi	s	0.6000E-02	0.4900
33	vasg	s	0.5000E-02	0.7510
34	vast	s	0.000	0.1500E-01
35	vass	s	0.1000E-02	0.2390
36	vasn	s	0.1000E-02	0.3970
37	vasi	s	0.1000E-02	0.1050
38	vbrg	s	0.000	1.989
39	vbrt	s	0.000	0.3990
40	vbrs	s	0.000	0.5440
41	vbrn	s	0.000	0.6790
42	vbri	s	0.000	0.4280
43	varg	s	0.6000E-02	1.325
44	vart	s	0.1000E-02	0.6500E-01
45	vars	s	0.3000E-02	0.3970
46	varn	s	0.1000E-02	0.5970
47	vari	s	0.000	0.2660

48	mdg	s	0.1210	1.298		
49	mdt	s	0.1170	1.215		
50	mds	s	0.1370	1.351		
51	mdn	s	0.2300E-01	1.260		
52	mdi	s	0.1160	1.247		
53	tmg	s	-0.3530	0.1920		
54	tmt	s	-0.2590	0.3660		
55	tms	s	-0.4300	0.3580		
56	tmn	s	-0.5100	0.2450		
57	tmi	s	-0.4050	0.2860		
58	mr	s	0.5990	1.219		
59	rnf	s	-0.1900E-01	0.4510		
60	mdic	s	0.1200E-01	0.6630		
61	emd	s	0.4700E-01	0.7430		
62	mv	s	0.000	0.1830		
63	tension	s	10.00	25.00		4
64	clv	s	0.000	146.0		12
65	cs	s	0.3300	1.910		1
66	lora	s	0.000	92.58		
67	Class	d			2	

This shows the type, minimum, maximum and number of missing values of each variable.

Total	#cases w/	#missing					
#cases	miss.	D	ord.	vals	#X-var	#N-var	#F-var
170	0		17		0	0	0
#P-var	#M-var	#B-var	#C-var	#I-var			
0	0	0	0	0			

This shows the number of each type of variable.

No. cases used for training: 170

No. cases excluded due to 0 weight or missing D: 0

Univariate split highest priority

Interaction and linear splits 2nd and 3rd priorities

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

Min. node sample size: 2

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	10	1.059E-01	2.360E-02	2.601E-02	8.824E-02	3.796E-02
1	8	1.000E-01	2.301E-02	2.777E-02	8.824E-02	4.343E-02

2**	5	9.412E-02	2.239E-02	3.015E-02	5.882E-02	4.716E-02
3	3	1.353E-01	2.623E-02	3.009E-02	8.824E-02	4.745E-02
4	2	1.471E-01	2.716E-02	2.897E-02	1.176E-01	4.883E-02
5	1	5.000E-01	3.835E-02	9.213E-03	5.000E-01	2.508E-02

0-SE tree based on mean is marked with * and has 5 terminal nodes

0-SE tree based on median is marked with + and has 5 terminal nodes

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

The tree with the smallest mean CV cost is marked with an asterisk.

The selected tree is marked with two asterisks; it is the smallest one having mean CV cost within the specified standard error (SE) bounds.

The mean CV costs and SEs are given in the 3rd and 4th columns.

The other columns are bootstrap estimates used for experimental purposes.

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 variables	Interacting variable
1	170	170	glaucoma	5.000E-01	lora	
2	73	73	normal	9.589E-02	clv	
4T	62	62	normal	0.000E+00	-	
5	11	11	glaucoma	3.636E-01	lora	
10T	4	4	normal	0.000E+00	-	
11T	7	7	glaucoma	0.000E+00	-	
3	97	97	glaucoma	1.959E-01	clv	
6T	15	15	normal	6.667E-02	-	
7T	82	82	glaucoma	6.098E-02	tmi :clv	

This shows the tree structure in tabular form. A node with label k has its left and right child nodes are labeled 2k and 2k+1, respectively. Terminal nodes are indicated with the symbol T. The notation ‘:tmi’ at node 7 indicates that the variable clv has an interaction with the split variable vass.

Number of terminal nodes of final tree: 5

Total number of nodes of final tree: 9

Second best split variable (based on curvature test) at root node is clv

This says that lora is the second best variable to split the root node.

Classification tree:

The tree structure is shown next in indented text form.

Node 1: lora <= 56.400730

Node 2: clv <= 8.4000000 or NA

```

Node 4: normal
Node 2: clv > 8.4000000
Node 5: lora <= 49.233715
Node 10: normal
Node 5: lora > 49.233715 or NA
Node 11: glaucoma
Node 1: lora > 56.400730 or NA
Node 3: clv <= 2.0000000
Node 6: normal
Node 3: clv > 2.0000000 or NA
Node 7: glaucoma

```

Node compositions and other details are given next.

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

```

Node 1: Intermediate node
A case goes into Node 2 if lora <= 56.400730
lora mean = 57.554944
  Class      Number  Posterior
glaucoma      85    0.50000
normal        85    0.50000
Number of training cases misclassified = 85
Predicted class is glaucoma
-----
Node 2: Intermediate node
A case goes into Node 4 if clv <= 8.4000000 or NA
clv mean = 5.4861111
  Class      Number  Posterior
glaucoma       7    0.09589
normal        66    0.90411
Number of training cases misclassified = 7
Predicted class is normal
-----
Node 4: Terminal node
  Class      Number  Posterior
glaucoma       0    0.00000
normal        62    1.00000
Number of training cases misclassified = 0
Predicted class is normal
-----
Node 5: Intermediate node
A case goes into Node 10 if lora <= 49.233715
lora mean = 49.099645
  Class      Number  Posterior
glaucoma       7    0.63636

```



```

normal          4      0.36364
Number of training cases misclassified =  4
Predicted class is glaucoma
-----

Node 10: Terminal node
  Class      Number  Posterior
glaucoma      0      0.00000
normal        4      1.00000
Number of training cases misclassified =  0
Predicted class is normal
-----

Node 11: Terminal node
  Class      Number  Posterior
glaucoma      7      1.00000
normal        0      0.00000
Number of training cases misclassified =  0
Predicted class is glaucoma
-----

Node 3: Intermediate node
A case goes into Node 6 if clv <= 2.0000000
clv mean = 35.820930
  Class      Number  Posterior
glaucoma     78      0.80412
normal       19      0.19588
Number of training cases misclassified =  19
Predicted class is glaucoma
-----

Node 6: Terminal node
  Class      Number  Posterior
glaucoma      1      0.06667
normal       14      0.93333
Number of training cases misclassified =  1
Predicted class is normal
-----

Node 7: Terminal node
  Class      Number  Posterior
glaucoma     77      0.93902
normal        5      0.06098
Number of training cases misclassified =  5
Predicted class is glaucoma
-----

Classification matrix for training sample:
Predicted      True class
class          glaucoma   normal

```

glaucoma	84	5
normal	1	80
Total	85	85

Number of cases used for tree construction: 170

Number misclassified: 6

Resubstitution est. of mean misclassification cost: 0.3529E-01

Observed and fitted values are stored in `glaucoma.fit`

LaTeX code for tree is in `glaucoma.tex`

R code is stored in `glaucomapred.r`

Figure 1 shows the classification tree drawn by LaTeX using the file `glaucoma.tex`. The last sentence in its caption gives the second best variable for splitting the root node. The top lines of the file `glaucoma.fit` are shown below. Their order corresponds to the order of the observations in the training sample file. The 1st column (labeled `train`) indicates whether the observation is used (“y”) or not used (“n”) to fit the model. Since we used the entire data set to fit the model here, all the entries in the first column are y. The 2nd column gives the terminal node number that the observation belongs to and the 3rd and 4th columns give its observed and predicted classes. The last two columns give the number of glaucoma and normal observations in the node where the observation belongs. They may be used to estimate the class probabilities in the node.

train	node	observed	predicted	"glaucoma"	"normal"
y	4	"normal"	"normal"	0	62
y	4	"normal"	"normal"	0	62
y	4	"normal"	"normal"	0	62
y	4	"normal"	"normal"	0	62
y	6	"normal"	"normal"	1	14

The file `glaucomapred.r` contains this R function:

```
predicted <- function(){
  if(!is.na(lora) & lora <= 56.4007300000 ){
    if(is.na(clv) | clv <= 8.40000000000 ){
      nodeid <- 4
      predict <- "normal"
    } else {
      if(!is.na(lora) & lora <= 49.2337150000 ){
        nodeid <- 10
        predict <- "normal"
      } else {
        nodeid <- 11
        predict <- "glaucoma"
      }
    }
  }
}
```

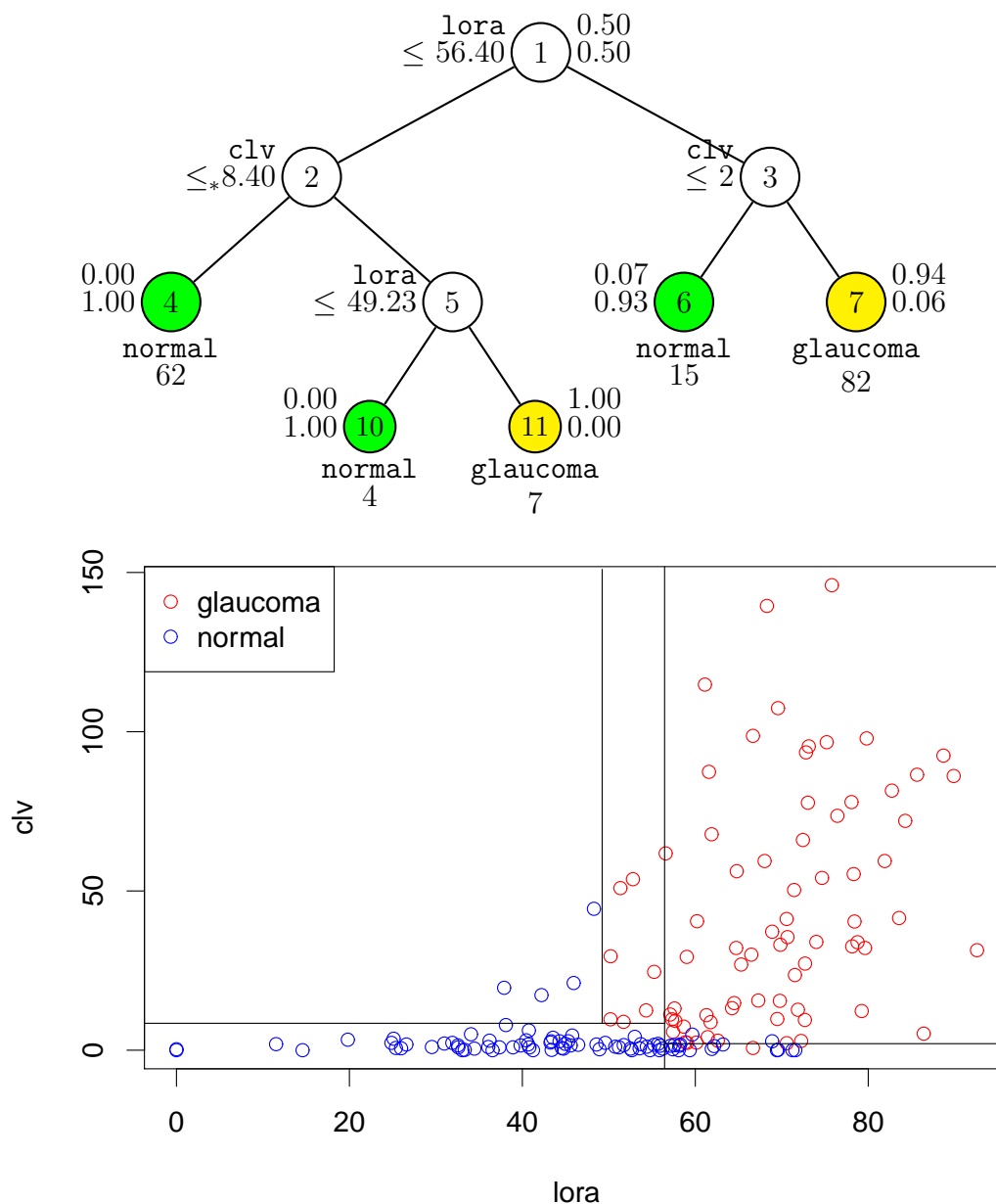


Figure 1: GUIDE v.29.6 0.50-SE pruned classification tree for predicting **Class** using estimated priors and unit misclassification costs. Maximum number of split levels is 10 and minimum node sample size is 2. At each split, an observation goes to the left branch if and only if the condition is satisfied. The symbol ' \leq_* ' stands for ' \leq or missing'. Predicted classes and sample sizes printed below terminal nodes; class proportions for **Class** = *glaucoma* and *normal* beside nodes. Second best split variable at root node is *clv*.

```

    }
  }
} else {
  if(!is.na(clv) & clv <= 2.000000000000 ){
    nodeid <- 6
    predict <- "normal"
  } else {
    nodeid <- 7
    predict <- "glaucoma"
  }
}
return(c(nodeid,predict))
}

```

4.2 Linear splits: glaucoma data

This section shows how to make GUIDE use linear splits on two variables at a time.

```

0. Read the warranty disclaimer
1. Create an input file for model fitting or importance scoring (recommended)
2. Convert data to other formats without creating input file
Input your choice: 1
Name of batch input file: lin.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: lin.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):2
    Choosing 2 enables more options.
Input 1 for simple, 2 for nearest-neighbor, 3 for kernel method
([1:3], <cr>=1):
    Options 2 and 3 yield nearest-neighbor and kernel discriminant node models.
Input 0 for linear, interaction and univariate splits (in this order),
    1 for univariate, linear and interaction splits (in this order),
    2 to skip linear splits,
    3 to skip linear and interaction splits:
Input your choice ([0:3], <cr>=1):0
    Option 1 is the default.
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 characters);
enclose with matching quotes if it has spaces: glaucoma.dsc

```

```

Reading data description file ...
Training sample file: glaucomadata.txt
Missing value code: NA
Records in data file start on line 2
Warning: N variables changed to S
Dependent variable is Class
Reading data file ...
Number of records in data file: 170
Length of longest data entry: 8
Checking for missing values ...
Total number of cases: 170
Number of classes: 2
Re-checking data ...
Allocating missing value information
Assigning codes to categorical and missing values
Finished checking data
Creating missing value indicators
Rereading data

```

Class	#Cases	Proportion
glaucoma	85	0.50000000
normal	85	0.50000000

Total	#cases w/	#missing							
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var	#B-var	#C-var	
170	0	17	0	0	0	66	0	0	

```

No. cases used for training: 170
No. cases excluded due to 0 weight or missing D: 0
Finished reading data file
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 for 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):
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):
Default minimum node sample size is 10
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): lin.tex

```

Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1):
Choosing 2 will give a tree with no node labels.
 Input 1 to number all nodes, 2 to number leaves only ([1:2], <cr>=1):
 Input 1 to color terminal nodes, 2 otherwise ([1:2], <cr>=1):
 Choose amount of detail in nodes of LaTeX tree diagram
 Input 0 for #errors, 1 for class sizes, 2 for nothing ([0:2], <cr>=1):
Choose 2 if a large tree is expected.
 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 variables and their values
 Input your choice ([1:2], <cr>=1): 2
Choose 2 to output the info to another file for further processing.
 Input file name: linvar.txt
 Input 2 to save individual fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
 Input name of file to store node ID and fitted value of each case: lin.fit
 Input 2 to save terminal node IDs for importance scoring; 1 otherwise ([1:2], <cr>=1):
 Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):2
 Input file name: linpred.r
 Input file is created!
 Run GUIDE with the command: guide < lin.in

Running GUIDE with the input file yields the following results. The \LaTeX tree diagram and partitions are shown in Figure 2.

```
Node 1: 0.41110165 * clv + lora <= 59.402920
Node 2: normal
Node 1: 0.41110165 * clv + lora > 59.402920 or NA
Node 3: glaucoma
```

Contents of linvar.txt: This file gives information about the splits:

```
1 1 lora clv      2  0.4111016476E+00  0.5940292030E+02
2 t mdn clv "normal"
3 t cs ean "glaucoma"
```

Each row refers to a node. The 1st column gives the node number. The 2nd column contains the letter 1, n, s, c, or t, indicating a split on two variables, a n variable, a s variable, a c variable, or a terminal node, respectively. The 3rd and 4th columns give the names of the 2 variables in a bivariate split or the names of the split variable and the interacting variable in a univariate split. If a node cannot be split, the words NONE are printed. If a node is terminal, the predicted class is printed in the 5th column. Otherwise, if it is a non-terminal node, the 5th column gives the number of values to follow. In the above example, the 2 in the 5th column of each non-terminal node indicates that it is followed by two parameter values defining the linear split. If the split is on a categorical variable, the 5th column gives the number of categorical values defining the split and the 6th and subsequent columns give their values.

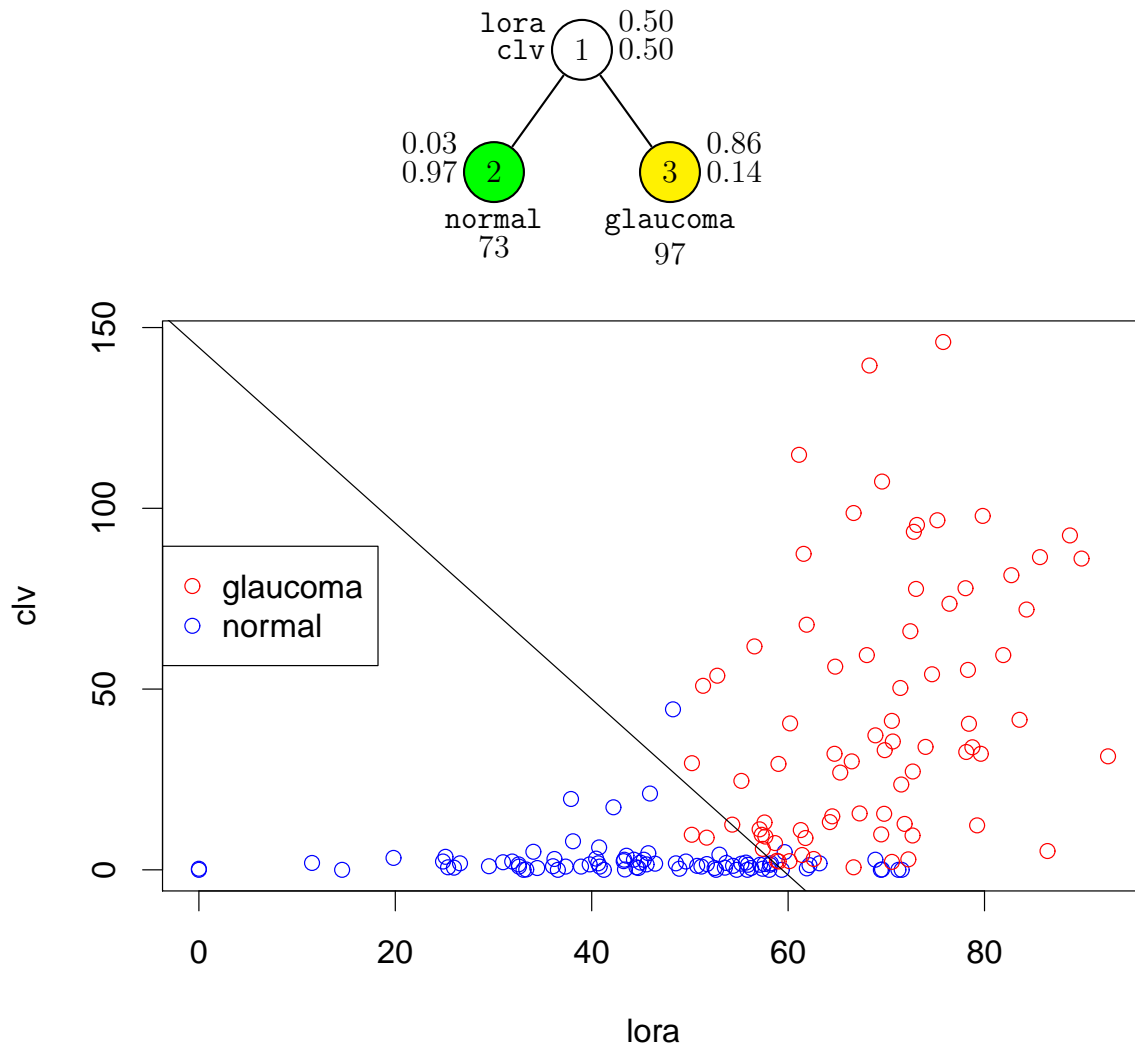


Figure 2: GUIDE v.29.6 0.50-SE pruned classification tree for predicting **Class** using linear split priority, estimated priors and unit misclassification costs. Maximum number of split levels is 10 and minimum node sample size is 10. At each split, an observation goes to the left branch if and only if the condition is satisfied. Predicted classes and sample sizes printed below terminal nodes; class proportions for **Class** = `glaucoma` and `normal` beside nodes.

Contents of linpred.r: This file contains the following R function for predicting future observations:

```
predicted <- function(){  
  if(!is.na(lora) & !is.na(clv) & 0.411101647572 * clv + lora <= 59.4029202973 ){  
    nodeid <- 2  
    predict <- "normal"  
  } else {  
    nodeid <- 3  
    predict <- "glaucoma"  
  }  
  return(c(nodeid,predict))  
}
```

4.3 Univariate splits, categorical predictors: peptide data

GUIDE can be used with categorical (i.e., nominal) predictor variables as well. We show this with a data set on peptide binding analyzed by [Segal \(1988\)](#) who used CART. The data consist of observations on 310 peptides, 181 of which bind to a Class I MHC molecule and 129 do not. The data are in the file `peptidedata.txt`. Column 1 contains the peptide ID and column 2 its binding status (`bind`). The remaining 112 columns are predictor variables, all continuous except for the last 8 which are categorical (named `pos1-pos8`), each taking 18–20 nominal values. Our goal here is to build a model to predict `bind` from these 8 categorical variables.

The GUIDE description is `peptide.dsc`. Note that the 3rd line of the file is “2”, indicating that the data begin on line 2 of `peptidedata.txt` (the first line of the latter contain the names of the variables). Note also that the continuous variables are excluded from the model by designating each of them with an “x”.

4.3.1 Input file generation

We use all the default options to produce a GUIDE input file.

```
0. Read the warranty disclaimer  
1. Create an input file for model fitting or importance scoring (recommended)  
2. Convert data to other formats without creating input file  
Input your choice: 1  
Name of batch input file: peptide.in  
Input 1 for model fitting, 2 for importance or DIF scoring,  
3 for data conversion ([1:3], <cr>=1):  
Name of batch output file: peptide.out  
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):  
Input 1 for classification, 2 for regression, 3 for propensity score grouping
```



```

(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: peptide.dsc
Reading data description file ...
Training sample file: peptidedata.txt
Missing value code: NA
Records in data file start on line 2
Dependent variable is bind
Reading data file ...
Number of records in data file: 310
Length of longest data entry: 6
Checking for missing values ...
Total number of cases: 310
Number of classes =                2
Col. no. Categorical variable      #levels      #missing values
    107 pos1                        18              0
    108 pos2                        20              0
    109 pos3                        20              0
    110 pos4                        20              0
    111 pos5                        20              0
    112 pos6                        20              0
    113 pos7                        19              0
    114 pos8                        20              0
Re-checking data ...
Assigning codes to categorical and missing values
Finished checking data
Rereading data
Class      #Cases      Proportion
0           129      0.41612903
1           181      0.58387097
    Total #cases w/  #missing
    #cases miss. D ord. vals  #X-var  #N-var  #F-var  #S-var  #B-var  #C-var
    310      0      0      105      0      0      0      0      8
No. cases used for training: 310
Finished reading data file
Choose 1 for estimated priors, 2 for equal priors, 3 for 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):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): peptide.tex
Input 2 to save individual fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: peptide.fit

```

Input file is created!
Run GUIDE with the command: `guide < peptide.in`

4.3.2 Results

Results from the output file `peptide.out` follow.

```

Classification tree
Pruning by cross-validation
Data description file: peptide.dsc
Training sample file: peptidedata.txt
Missing value code: NA
Records in data file start on line 2
Dependent variable is bind
Number of records in data file: 310
Length of longest entry in data file: 6
Number of classes: 2
Training sample class proportions of D variable bind:
Class  #Cases      Proportion
0         129      0.41612903
1         181      0.58387097

```

Summary information (without x variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
2	bind	d			2	
107	pos1	c			18	
108	pos2	c			20	
109	pos3	c			20	
110	pos4	c			20	
111	pos5	c			20	
112	pos6	c			20	
113	pos7	c			19	
114	pos8	c			20	

Total	#cases w/ #cases	miss. D	#missing ord. vals	#X-var	#N-var	#F-var	#S-var
310	0	0	105	0	0	0	

#P-var	#M-var	#B-var	#C-var	#I-var
0	0	0	8	0

4.3 Univariate splits, categorical predictors: peptide data 4 CLASSIFICATION

No. cases used for training: 310

Univariate split highest priority

Interaction and linear splits 2nd and 3rd priorities

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

Min. node sample size: 2

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	10	1.097E-01	1.775E-02	2.409E-02	9.677E-02	2.879E-02
2	8	1.097E-01	1.775E-02	2.409E-02	9.677E-02	2.879E-02
3	6	1.097E-01	1.775E-02	2.242E-02	9.677E-02	2.416E-02
4	5	1.097E-01	1.775E-02	2.206E-02	8.065E-02	2.231E-02
5	3	1.194E-01	1.841E-02	2.150E-02	1.129E-01	2.576E-02
6**	2	1.097E-01	1.775E-02	2.286E-02	8.065E-02	2.670E-02
7	1	4.161E-01	2.800E-02	3.207E-03	4.194E-01	1.019E-03

0-SE tree based on mean is marked with * and has 2 terminal nodes

0-SE tree based on median is marked with + and has 2 terminal nodes

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.

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

Node label	Total cases	Train cases	Predicted class	Node cost	Split variables	Interacting variable
1	310	310	1	4.161E-01	pos5	
2T	169	169	1	5.917E-02	pos1	
3T	141	141	0	1.560E-01	pos8	

Number of terminal nodes of final tree: 2

Total number of nodes of final tree: 3

Second best split variable (based on curvature test) at root node is pos1

4.3 Univariate splits, categorical predictors: peptide data 4 CLASSIFICATION

Classification tree:

At splits on categorical variables, values not in training data go to the right

Node 1: pos5 = "F", "M", "Y"
Node 2: 1
Node 1: pos5 /= "F", "M", "Y"
Node 3: 0

Node 1: Intermediate node
A case goes into Node 2 if pos5 =
"F", "M", "Y"
pos5 mode = "Y"

Class	Number	Posterior
0	129	0.41613
1	181	0.58387

Number of training cases misclassified = 129
Predicted class is 1

Node 2: Terminal node

Class	Number	Posterior
0	10	0.05917
1	159	0.94083

Number of training cases misclassified = 10
Predicted class is 1

Node 3: Terminal node

Class	Number	Posterior
0	119	0.84397
1	22	0.15603

Number of training cases misclassified = 22
Predicted class is 0

Classification matrix for training sample:

Predicted	True class	
class	0	1
0	119	22
1	10	159
Total	129	181

Number of cases used for tree construction: 310
Number misclassified: 32

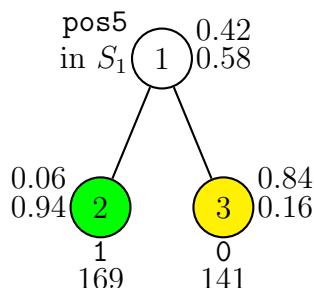


Figure 3: GUIDE v.29.6 0.50-SE pruned classification tree for predicting `bind` using estimated priors and unit misclassification costs. Maximum number of split levels is 10 and minimum node sample size is 2. At each split, an observation goes to the left branch if and only if the condition is satisfied. For splits on categorical variables, values not present in the training sample go to the right. Set $S_1 = \{F, M, Y\}$. Predicted classes and sample sizes printed below terminal nodes; class proportions for `bind` = 0 and 1 beside nodes. Second best split variable at root node is `pos1`.

Resubstitution est. of mean misclassification cost: 0.1032

Observed and fitted values are stored in `peptide.fit`
 LaTeX code for tree is in `peptide.tex`

The results indicate that the largest tree before pruning has 10 terminal nodes. The pruned tree (marked by “**”) has 2 terminal nodes. Its cross-validation estimate of misclassification cost (or error rate here) is 0.1097. Figure 3 shows the pruned tree. It splits on `pos5`, sending values F, M and Y to the left node. The second best variable to split the root node is `pos1`.

4.4 Unbalanced classes and equal priors: hepatitis data

If a data set has one dominant class, a classification tree may be null after pruning, as it may be hard to beat the classifier that predicts every observation to belong to the dominant class. Nonetheless, it may be of interest to find out which variables are more predictive and how they affect the dependent variable. One solution is to use the equal priors option. The resulting model should not be used for prediction. Instead, by comparing the class proportions in each terminal node against those at the root node, it can be used to identify the nodes where the dominant class proportion is much higher or much lower than average (i.e., at the root node).

We use a hepatitis data set to show this. The files are `hepdsc.txt` and `hepdat.txt`; see <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”. That is, 79% of the individuals are in the “live” class. The contents of `hepdsc.txt` are:

```
hepdsc.txt
"?"
1
1 CLASS d
2 AGE n
3 SEX c
4 STEROID c
5 ANTIVIRALS c
6 FATIGUE c
7 MALAISE c
8 ANOREXIA c
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
```

Using the default estimated priors yields a tree with one split, as shown on the left side of Figure 4. To obtain more splits, we can use equal priors.

```
0. Read the warranty disclaimer
1. Create an input file for model fitting or importance scoring (recommended)
2. Convert data to other formats without creating input file
Input your choice: 1
Name of batch input file: hepeq.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: hepeq.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
    Option 2 is needed for equal or specified priors.
Input 1 for simple, 2 for nearest-neighbor, 3 for kernel method ([1:3], <cr>=1):
Input 0 for linear, interaction and univariate splits (in this order),
```

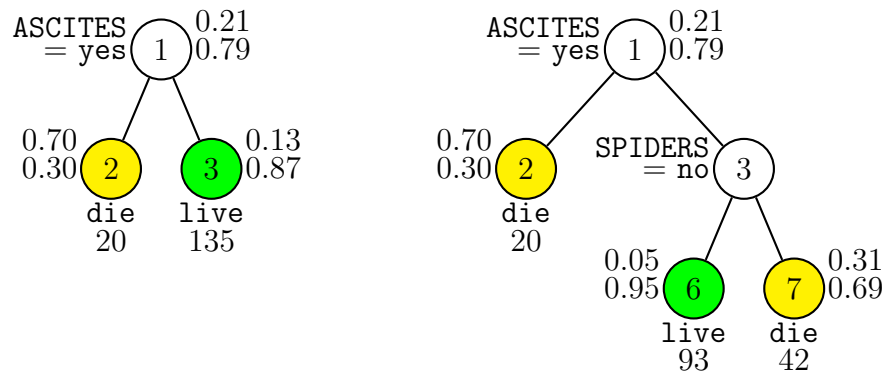


Figure 4: GUIDE v.29.6 0.50-SE classification trees for predicting **CLASS** using estimated (left) and equal (right) priors and unit misclassification costs. At each split, an observation goes to the left branch if and only if the condition is satisfied. Predicted classes and sample sizes printed below terminal nodes; class proportions for **CLASS** = **die** and **live** beside nodes. Second best split variable at both root nodes is **ALBUMIN**.

```

1 for univariate, linear and interaction splits (in this order),
2 to skip linear splits,
3 to skip linear and interaction splits:
Input your choice ([0:3], <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 characters);
enclose with matching quotes if it has spaces: hepdsc.txt
Reading data description file ...
Training sample file: hepdsc.txt
Missing value code: ?
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is CLASS
Reading data file ...
Number of records in data file: 155
Length of longest data entry: 6
Checking for missing values ...
Total number of cases: 155
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Number of classes: 2
Col. no. Categorical variable    #levels    #missing values
      3 SEX                      2           0
      4 STEROID                  2           1

```

5 ANTIVIRALS	2	0
6 FATIGUE	2	1
7 MALAISE	2	1
8 ANOREXIA	2	1
9 BIGLIVER	2	10
10 FIRMLIVER	2	11
11 SPLEEN	2	5
12 SPIDERS	2	5
13 ASCITES	2	5
14 VARICES	2	5
20 HISTOLOGY	2	0

Re-checking data ...
Allocating missing value information
Assigning codes to categorical and missing values
Data checks complete
Creating missing value indicators
Rereading data

Class	#Cases	Proportion
die	32	0.20645161
live	123	0.79354839

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

No. cases used for training: 155
No. cases excluded due to 0 weight or missing D: 0
Finished reading data file
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 for priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):2
Option 2 is for equal priors.
Choose 1 for unit misclassification costs, 2 to input costs from a file
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>=2):
Default max. number of split levels: 10
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 2
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): hepeq.tex


```
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 terminal nodes, 2 otherwise ([1:2], <cr>=1):
Choose amount of detail in nodes of LaTeX tree diagram
Input 0 for #errors, 1 for class sizes, 2 for nothing ([0:2], <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):3
Input file name: hepvar.txt
Contents of this file are shown below.
Input 2 to save individual fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: hepeq.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < hepeq.in
```

The resulting tree in text form is:

```
Node 1: ASCITES = "yes"
  Node 2: die
Node 1: ASCITES /= "yes"
  Node 3: SPIDERS = "no"
    Node 6: live
  Node 3: SPIDERS /= "no"
    Node 7: die
```

Figure 4 shows the L^AT_EX trees using estimated priors (left) and equal priors (right). Nodes that predict the same class have the same color. The tree using equal priors has one more split (on SPIDERS). But both trees misclassify the same number of samples. Therefore the left tree, being shorter, is preferred if priors are estimated. On the other hand, since the ratio of “die” to “live” classes is 32:123, equal priors makes each “die” observation equivalent to $r = 123/32 = 3.84375$ “live” observations. Consequently, a terminal node is classified as “die” if its ratio of “live” to “die” observations is less than r . Note that although only 21% of the data are in the “die” class, most of these individuals are in nodes 2 and 7 (70% and 31%, respectively).

Contents of hepvar.txt: This file summarizes the information by node:

```
1 c ASCITES ASCITES      1  "yes"
2 t BILIRUBIN BILIRUBIN  "die"
1 c ASCITES ASCITES      1  "yes"
3 c SPIDERS SPIDERS       1  "no"
```

```

6 t MALAISE MALAISE "live"
3 c SPIDERS SPIDERS      1  "no"
7 t SEX SEX "die"

```

4.5 Unequal misclassification costs: hepatitis data

So far, we have assumed that the cost of misclassifying a “die” observation as “live” is the same as the opposite. Another way to obtain a nonnull tree for the hepatitis data is to use unequal misclassification costs. For example, if we think that the cost of misclassifying a “die” observation as “live” is four times that of the opposite, we will use the misclassification cost matrix

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

where $C(i, j)$ denotes the cost of classifying an observation as class i given that it belongs to class j . Note that GUIDE sorts the class values in alphabetical order, so that “die” is treated as class 1 and “live” as class 2 here. This matrix is saved in the text file `cost.txt` which has these two lines:

```

0 1
4 0

```

The following lines in the input file generation step shows where this file is used:

```

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

```

The resulting tree is the same as that for equal priors in Figure 4.

4.6 More than 2 classes: dermatology with ordinal predictors

The data, taken from UCI (Ilter and Guvenir, 1998), give the diagnosis (6 classes) and clinical and laboratory measurements of 34 ordinal predictor variables for 358 patients. The description and data files are `derm.dsc` and `derm.dat`, respectively.

4.6.1 Default option

The default option gives the following results.

```

Classification tree
Pruning by cross-validation
Data description file: derm.dsc
Training sample file: derm.dat
Missing value code: ?
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is class
Number of records in data file: 358
Length of longest entry in data file: 2
Number of classes: 6
Training sample class proportions of D variable class:

```

Class	#Cases	Proportion
1	111	0.31005587
2	60	0.16759777
3	71	0.19832402
4	48	0.13407821
5	48	0.13407821
6	20	0.05586592

Summary information (without x variables)

d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	erythema	s	0.000	3.000		
2	scaling	s	0.000	3.000		
3	borders	s	0.000	3.000		
4	itching	s	0.000	3.000		
5	koebner	s	0.000	3.000		
6	polypap	s	0.000	3.000		
7	follipap	s	0.000	3.000		
8	oralmuc	s	0.000	3.000		
9	knee	s	0.000	3.000		
10	scalp	s	0.000	3.000		
11	history	s	0.000	1.000		
12	melanin	s	0.000	3.000		
13	eosin	s	0.000	2.000		
14	PNL	s	0.000	3.000		
15	fibrosis	s	0.000	3.000		

16	exocyto	s	0.000	3.000
17	acantho	s	0.000	3.000
18	hyperker	s	0.000	3.000
19	paraker	s	0.000	3.000
20	clubbing	s	0.000	3.000
21	elongation	s	0.000	3.000
22	thinning	s	0.000	3.000
23	spongiform	s	0.000	3.000
24	munro	s	0.000	3.000
25	hypergran	s	0.000	3.000
26	disappea	s	0.000	3.000
27	basal	s	0.000	3.000
28	spongiosis	s	0.000	3.000
29	sawtooth	s	0.000	3.000
30	hornplug	s	0.000	3.000
31	perifoll	s	0.000	3.000
32	inflamm	s	0.000	3.000
33	bandlike	s	0.000	3.000
34	age	s	0.000	75.00
35	class	d		

6

Total	#cases w/	#missing					
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var	
358	0	0	0	0	0	34	
#P-var	#M-var	#B-var	#C-var	#I-var			
0	0	0	0	0			

No. cases used for training: 358

Univariate split highest priority

Interaction and linear splits 2nd and 3rd priorities

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

Min. node sample size: 2

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	10	4.190E-02	1.059E-02	9.024E-03	2.817E-02	1.040E-02
2**	9	4.190E-02	1.059E-02	9.024E-03	2.817E-02	1.040E-02
3++	8	5.587E-02	1.214E-02	1.649E-02	2.817E-02	1.562E-02
4	7	6.145E-02	1.269E-02	1.648E-02	4.206E-02	2.008E-02

5	5	1.816E-01	2.037E-02	2.921E-02	1.806E-01	2.878E-02
6	2	4.804E-01	2.641E-02	2.058E-02	4.722E-01	2.831E-02
7	1	6.899E-01	2.444E-02	2.184E-02	6.806E-01	2.412E-02

0-SE tree based on mean is marked with * and has 9 terminal nodes

0-SE tree based on median is marked with + and has 8 terminal nodes

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 variables	Interacting variable
1	358	358	1	6.899E-01	polypap	
2	290	290	1	6.172E-01	oralmuc	
4	287	287	1	6.132E-01	fibrosis	
8	239	239	1	5.356E-01	spongiosis	
16	121	121	1	8.264E-02	elongation	
32T	9	9	6	4.444E-01	folliap	
33T	112	112	1	8.929E-03	-	
17	118	118	2	5.169E-01	perifoll	
34	103	103	2	4.563E-01	koebner	
68	64	64	2	1.406E-01	disappea	
136T	58	58	2	5.172E-02	spongiosis	
137T	6	6	4	0.000E+00	-	
69T	39	39	4	2.564E-02	-	
35T	15	15	6	6.667E-02	-	
9T	48	48	5	0.000E+00	-	
5T	3	3	3	0.000E+00	-	
3T	68	68	3	0.000E+00	-	

Number of terminal nodes of final tree: 9

Total number of nodes of final tree: 17

Second best split variable (based on curvature test) at root node is bandlike

Classification tree:

Node 1: polypap <= 0.50000000

```

Node 2: oralmuc <= 0.50000000
Node 4: fibrosis <= 0.50000000
Node 8: spongiosis <= 0.50000000
Node 16: elongation <= 0.50000000
Node 32: 6
Node 16: elongation > 0.50000000 or ?
Node 33: 1
Node 8: spongiosis > 0.50000000 or ?
Node 17: perifoll <= 0.50000000
Node 34: koebner <= 0.50000000
Node 68: disappea <= 0.50000000
Node 136: 2
Node 68: disappea > 0.50000000 or ?
Node 137: 4
Node 34: koebner > 0.50000000 or ?
Node 69: 4
Node 17: perifoll > 0.50000000 or ?
Node 35: 6
Node 4: fibrosis > 0.50000000 or ?
Node 9: 5
Node 2: oralmuc > 0.50000000 or ?
Node 5: 3
Node 1: polypap > 0.50000000 or ?
Node 3: 3

```

```

Node 1: Intermediate node
A case goes into Node 2 if polypap <= 0.50000000
polypap mean = 4.4972E-01
Class      Number  Posterior
1           111    0.31006
2           60     0.16760
3           71     0.19832
4           48     0.13408
5           48     0.13408
6           20     0.05587
Number of training cases misclassified = 247
Predicted class is 1
-----

```

```

Node 2: Intermediate node
A case goes into Node 4 if oralmuc <= 0.50000000
oralmuc mean = 2.0690E-02
Class      Number  Posterior
1           111    0.38276

```

2	60	0.20690
3	3	0.01034
4	48	0.16552
5	48	0.16552
6	20	0.06897

Number of training cases misclassified = 179

Predicted class is 1

Node 4: Intermediate node

A case goes into Node 8 if fibrosis \leq 0.50000000

fibrosis mean = 3.7979E-01

Class	Number	Posterior
1	111	0.38676
2	60	0.20906
3	0	0.00000
4	48	0.16725
5	48	0.16725
6	20	0.06969

Number of training cases misclassified = 176

Predicted class is 1

Node 8: Intermediate node

A case goes into Node 16 if spongiosis \leq 0.50000000

spongiosis mean = 1.0544E+00

Class	Number	Posterior
1	111	0.46444
2	60	0.25105
3	0	0.00000
4	48	0.20084
5	0	0.00000
6	20	0.08368

Number of training cases misclassified = 128

Predicted class is 1

Node 16: Intermediate node

A case goes into Node 32 if elongation \leq 0.50000000

elongation mean = 2.0909E+00

Class	Number	Posterior
1	111	0.91736
2	3	0.02479
3	0	0.00000
4	1	0.00826
5	0	0.00000
6	6	0.04959

Number of training cases misclassified = 10

Predicted class is 1

```

-----
Node 32: Terminal node
Class      Number  Posterior
1           0      0.00000
2           3      0.33333
3           0      0.00000
4           1      0.11111
5           0      0.00000
6           5      0.55556
Number of training cases misclassified =  4
Predicted class is 6
-----
Node 33: Terminal node
Class      Number  Posterior
1          111      0.99107
2           0      0.00000
3           0      0.00000
4           0      0.00000
5           0      0.00000
6           1      0.00893
Number of training cases misclassified =  1
Predicted class is 1
-----
Node 17: Intermediate node
A case goes into Node 34 if perifoll <= 0.50000000
perifoll mean =  2.6271E-01
Class      Number  Posterior
1           0      0.00000
2          57      0.48305
3           0      0.00000
4          47      0.39831
5           0      0.00000
6          14      0.11864
Number of training cases misclassified =  61
Predicted class is 2
-----
Node 34: Intermediate node
A case goes into Node 68 if koebner <= 0.50000000
koebner mean =  5.4369E-01
Class      Number  Posterior
1           0      0.00000
2          56      0.54369
3           0      0.00000
4          47      0.45631
5           0      0.00000
6           0      0.00000

```


Number of training cases misclassified = 47
 Predicted class is 2

 Node 68: Intermediate node

A case goes into Node 136 if disappea \leq 0.50000000
 disappea mean = 9.3750E-02

Class	Number	Posterior
1	0	0.00000
2	55	0.85938
3	0	0.00000
4	9	0.14062
5	0	0.00000
6	0	0.00000

Number of training cases misclassified = 9
 Predicted class is 2

 Node 136: Terminal node

Class	Number	Posterior
1	0	0.00000
2	55	0.94828
3	0	0.00000
4	3	0.05172
5	0	0.00000
6	0	0.00000

Number of training cases misclassified = 3
 Predicted class is 2

 Node 137: Terminal node

Class	Number	Posterior
1	0	0.00000
2	0	0.00000
3	0	0.00000
4	6	1.00000
5	0	0.00000
6	0	0.00000

Number of training cases misclassified = 0
 Predicted class is 4

 Node 69: Terminal node

Class	Number	Posterior
1	0	0.00000
2	1	0.02564
3	0	0.00000
4	38	0.97436
5	0	0.00000
6	0	0.00000

Number of training cases misclassified = 1
 Predicted class is 4

 Node 35: Terminal node

Class	Number	Posterior
1	0	0.00000
2	1	0.06667
3	0	0.00000
4	0	0.00000
5	0	0.00000
6	14	0.93333

Number of training cases misclassified = 1
 Predicted class is 6

 Node 9: Terminal node

Class	Number	Posterior
1	0	0.00000
2	0	0.00000
3	0	0.00000
4	0	0.00000
5	48	1.00000
6	0	0.00000

Number of training cases misclassified = 0
 Predicted class is 5

 Node 5: Terminal node

Class	Number	Posterior
1	0	0.00000
2	0	0.00000
3	3	1.00000
4	0	0.00000
5	0	0.00000
6	0	0.00000

Number of training cases misclassified = 0
 Predicted class is 3

 Node 3: Terminal node

Class	Number	Posterior
1	0	0.00000
2	0	0.00000
3	68	1.00000
4	0	0.00000
5	0	0.00000
6	0	0.00000

Number of training cases misclassified = 0
 Predicted class is 3

Classification matrix for training sample:

Predicted class	True class					
	1	2	3	4	5	6
1	111	0	0	0	0	1
2	0	55	0	3	0	0
3	0	0	71	0	0	0
4	0	1	0	44	0	0
5	0	0	0	0	48	0
6	0	4	0	1	0	19
Total	111	60	71	48	48	20

Number of cases used for tree construction: 358

Number misclassified: 10

Resubstitution est. of mean misclassification cost: 0.2793E-01

Observed and fitted values are stored in uni.fit

LaTeX code for tree is in uni.tex

The tree is shown in Figure 5; it misclassifies 10 observations.

4.6.2 Nearest-neighbor option

One way to obtain a smaller tree is to fit a *classification model* to the data in each node and use it to classify the individual observations there. GUIDE has two means to achieve this: nearest-neighbor and kernel discrimination. For nearest-neighbor, an observation in a node is classified to the plurality class among observations within its neighborhood. The neighborhood is defined to be the whole node if the split variable is categorical. The input file for this option is obtained as follows.

Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: nn.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: nn.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
```

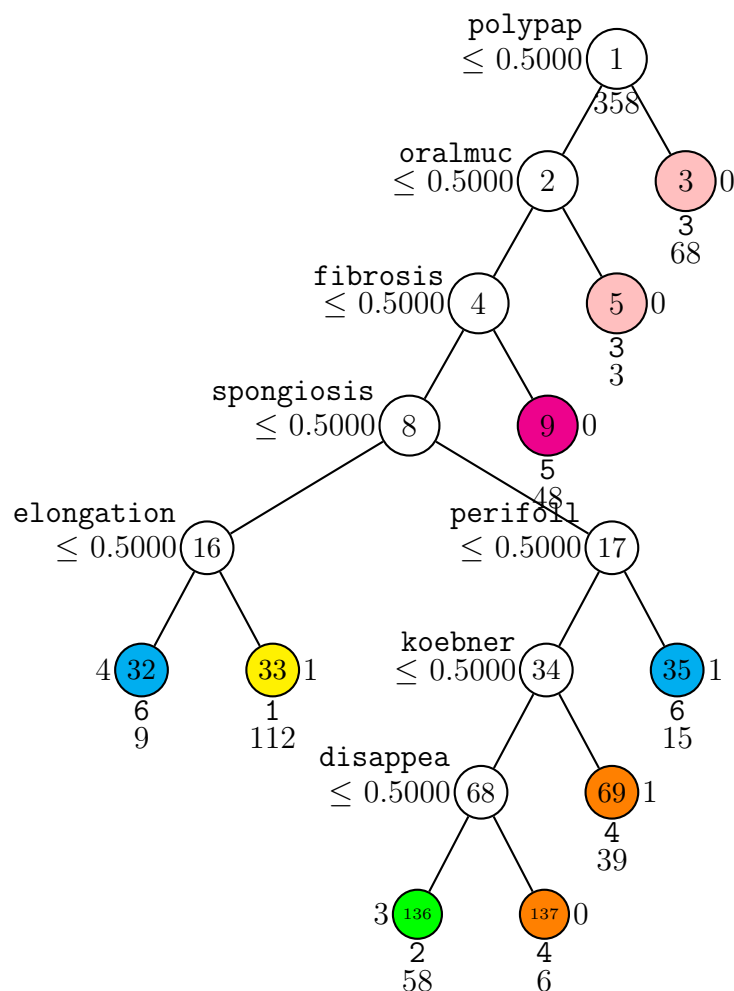


Figure 5: GUIDE v.29.6 0.50-SE pruned classification tree for predicting **class** using estimated priors and unit misclassification costs. Maximum number of split levels is 10 and minimum node sample size is 2. At each split, an observation goes to the left branch if and only if the condition is satisfied. Predicted classes and sample sizes printed below terminal nodes; #misclassified beside nodes. Second best split variable at root node is **bandlike**.

```

Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
Input 1 for simple, 2 for nearest-neighbor, 3 for kernel method ([1:3], <cr>=1): 2
  Choose nearest-neighbor option here.
Input 1 for univariate, 2 for bivariate preference ([1:2], <cr>=1):
  Default is univariate kernels.
Input 1 for interaction tests, 2 to skip them ([1:2], <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 characters);
enclose with matching quotes if it has spaces: derm.dsc
Reading data description file ...
Training sample file: derm.dat
Missing value code: ?
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is class
Reading data file ...
Number of records in data file: 358
Length of longest entry in data file: 2
Checking for missing values ...
Total number of cases: 358
Number of classes: 6
Re-checking data ...
Assigning codes to categorical and missing values
Data checks complete
Rereading data
Class  #Cases    Proportion
1         111    0.31005587
2          60    0.16759777
3          71    0.19832402
4          48    0.13407821
5          48    0.13407821
6          20    0.05586592
    Total  #cases w/  #missing
    #cases  miss. D  ord. vals  #X-var  #N-var  #F-var  #S-var
      358         0         0         0         0         0         34
    #M-var  #B-var  #C-var
         0         0         0
No. cases used for training: 358
Finished reading data file
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 for 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):
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):
Default minimum node sample size is 10
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): nn.tex
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 terminal nodes, 2 otherwise ([1:2], <cr>=1):
Choose amount of detail in nodes of LaTeX tree diagram
Input 0 for #errors, 1 for class proportions, 2 for nothing ([0:2], <cr>=0):
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 fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: nn.fit
Input file is created!
Run GUIDE with the command: guide < nn.in

```

Results

```

Classification tree
Pruning by cross-validation
Data description file: derm.dsc
Training sample file: derm.dat
Missing value code: ?
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is class
Number of records in data file: 358
Length of longest entry in data file: 2
Number of classes: 6
Training sample class proportions of D variable class:
Class  #Cases      Proportion

```

1	111	0.31005587
2	60	0.16759777
3	71	0.19832402
4	48	0.13407821
5	48	0.13407821
6	20	0.05586592

Summary information (without x variables)

d=dependent, b=split and fit cat variable using indicator variables,
 c=split-only categorical, i=fit-only categorical (via indicators),
 s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
 m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	erythema	s	0.000	3.000		
2	scaling	s	0.000	3.000		
3	borders	s	0.000	3.000		
4	itching	s	0.000	3.000		
5	koebner	s	0.000	3.000		
6	polypap	s	0.000	3.000		
7	follipap	s	0.000	3.000		
8	oralmuc	s	0.000	3.000		
9	knee	s	0.000	3.000		
10	scalp	s	0.000	3.000		
11	history	s	0.000	1.000		
12	melanin	s	0.000	3.000		
13	eosin	s	0.000	2.000		
14	PNL	s	0.000	3.000		
15	fibrosis	s	0.000	3.000		
16	exocyto	s	0.000	3.000		
17	acantho	s	0.000	3.000		
18	hyperker	s	0.000	3.000		
19	paraker	s	0.000	3.000		
20	clubbing	s	0.000	3.000		
21	elongation	s	0.000	3.000		
22	thinning	s	0.000	3.000		
23	spongiform	s	0.000	3.000		
24	munro	s	0.000	3.000		
25	hypergran	s	0.000	3.000		
26	disappea	s	0.000	3.000		
27	basal	s	0.000	3.000		
28	spongiosis	s	0.000	3.000		
29	sawtooth	s	0.000	3.000		
30	hornplug	s	0.000	3.000		
31	perifoll	s	0.000	3.000		

```

32 inflamm      s    0.000      3.000
33 bandlike     s    0.000      3.000
34 age          s    0.000     75.00
35 class        d

```

6

```

Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var
  358      0      0      0      0      0      0      34
#P-var #M-var #B-var #C-var #I-var
    0      0      0      0      0

```

No. cases used for training: 358

Univariate split highest priority
 Interaction splits 2nd priority; no linear splits
 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Nearest-neighbor node models
 Univariate preference
 Estimated priors
 Unit misclassification costs
 Split values for N and S variables based on exhaustive search
 Max. number of split levels: 10
 Min. node sample size: 10
 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	9	8.101E-02	1.442E-02	2.077E-02	5.556E-02	2.051E-02
2	8	8.101E-02	1.442E-02	2.077E-02	5.556E-02	2.051E-02
3	7	8.101E-02	1.442E-02	2.077E-02	5.556E-02	2.051E-02
4	6	8.101E-02	1.442E-02	2.077E-02	5.556E-02	2.051E-02
5**	5	8.101E-02	1.442E-02	2.077E-02	5.556E-02	2.051E-02
6	3	1.927E-01	2.085E-02	2.111E-02	1.829E-01	2.672E-02
7	1	5.000E-01	2.643E-02	2.346E-02	5.000E-01	3.233E-02

0-SE tree based on mean is marked with * and has 5 terminal nodes
 0-SE tree based on median is marked with + and has 5 terminal nodes
 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.

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

Node label	Total cases	Train cases	Predicted class	Node cost	Split variable followed by (+)fit variable(s)
1	358	358	1	5.028E-01	polypap +polypap
2	290	290	1	4.517E-01	fibrosis +fibrosis
4	242	242	1	2.851E-01	spongiosis +spongiosis
8T	123	123	1	5.691E-02	elongation +elongation
9	119	119	2	4.202E-01	follipap +follipap
18T	104	104	2	1.058E-01	koebner +koebner
19T	15	15	6	6.667E-02	-
5T	48	48	5	0.000E+00	-
3T	68	68	3	0.000E+00	-

Number of terminal nodes of final tree: 5

Total number of nodes of final tree: 9

Second best split variable (based on curvature test) at root node is bandlike

Classification tree:

```

Node 1: polypap <= 0.50000000
  Node 2: fibrosis <= 0.50000000
    Node 4: spongiosis <= 0.50000000
      Node 8: Mean cost = 0.56910569E-001
      Node 4: spongiosis > 0.50000000 or ?
        Node 9: follipap <= 0.50000000
          Node 18: Mean cost = 0.10576923
          Node 9: follipap > 0.50000000 or ?
            Node 19: Mean cost = 0.66666667E-001
        Node 2: fibrosis > 0.50000000 or ?
          Node 5: Mean cost = 0.00000000
    Node 1: polypap > 0.50000000 or ?
      Node 3: Mean cost = 0.00000000

```

Node 1: Intermediate node

A case goes into Node 2 if polypap <= 0.50000000

Number of nearest neighbors = 6

polypap mean = 4.4972E-01

Class	Number	Posterior	Fit variable polypap
1	111	0.31006	
2	60	0.16760	
3	71	0.19832	
4	48	0.13408	

5 48 0.13408

6 20 0.05587

Number of training cases misclassified = 180

If node model is inapplicable due to missing values, predicted class =

1

Node 2: Intermediate node

A case goes into Node 4 if fibrosis <= 0.50000000

Number of nearest neighbors = 6

fibrosis mean = 3.7586E-01

Class	Number	Posterior	Fit variable fibrosis
-------	--------	-----------	--------------------------

1	111	0.38276	
---	-----	---------	--

2	60	0.20690	
---	----	---------	--

3	3	0.01034	
---	---	---------	--

4	48	0.16552	
---	----	---------	--

5	48	0.16552	
---	----	---------	--

6	20	0.06897	
---	----	---------	--

Number of training cases misclassified = 131

If node model is inapplicable due to missing values, predicted class =

1

Node 4: Intermediate node

A case goes into Node 8 if spongiosis <= 0.50000000

Number of nearest neighbors = 6

spongiosis mean = 1.0537E+00

Class	Number	Posterior	Fit variable spongiosis
-------	--------	-----------	----------------------------

1	111	0.45868	
---	-----	---------	--

2	60	0.24793	
---	----	---------	--

3	3	0.01240	
---	---	---------	--

4	48	0.19835	
---	----	---------	--

5	0	0.00000	
---	---	---------	--

6	20	0.08264	
---	----	---------	--

Number of training cases misclassified = 69

If node model is inapplicable due to missing values, predicted class =

1

Node 8: Terminal node

Number of nearest neighbors = 5

elongation mean = 2.0569E+00

Class	Number	Posterior	Fit variable elongation
-------	--------	-----------	----------------------------

1	111	0.90244	
---	-----	---------	--

2	3	0.02439	
---	---	---------	--

3	2	0.01626	
---	---	---------	--

4	1	0.00813
5	0	0.00000
6	6	0.04878

Node 9: Intermediate node

A case goes into Node 18 if follipap <= 0.50000000

Number of nearest neighbors = 5

follipap mean = 2.5210E-01

Class	Number	Posterior	Fit variable follipap
1	0	0.00000	
2	57	0.47899	
3	1	0.00840	
4	47	0.39496	
5	0	0.00000	
6	14	0.11765	

Number of training cases misclassified = 50

If node model is inapplicable due to missing values, predicted class =
2

Node 18: Terminal node

Number of nearest neighbors = 5

koebner mean = 5.3846E-01

Class	Number	Posterior	Fit variable koebner
1	0	0.00000	
2	56	0.53846	
3	1	0.00962	
4	47	0.45192	
5	0	0.00000	
6	0	0.00000	

Node 19: Terminal node

Number of nearest neighbors = 3

Class	Number	Posterior
1	0	0.00000
2	1	0.06667
3	0	0.00000
4	0	0.00000
5	0	0.00000
6	14	0.93333

Node 5: Terminal node

Number of nearest neighbors = 4

Class	Number	Posterior
1	0	0.00000

```

2          0      0.00000
3          0      0.00000
4          0      0.00000
5         48      1.00000
6          0      0.00000
-----

```

Node 3: Terminal node

Number of nearest neighbors = 5

```

Class      Number  Posterior
1           0      0.00000
2           0      0.00000
3          68      1.00000
4           0      0.00000
5           0      0.00000
6           0      0.00000
-----

```

Classification matrix for training sample:

Predicted	True class					
class	1	2	3	4	5	6
1	111	0	0	0	0	1
2	0	55	1	9	0	0
3	0	0	68	0	0	0
4	0	1	0	38	0	0
5	0	0	0	0	48	0
6	0	4	2	1	0	19
Total	111	60	71	48	48	20

Number of cases used for tree construction: 358

Number misclassified: 19

Resubstitution est. of mean misclassification cost: 0.5307E-01

Observed and fitted values are stored in nn.fit

LaTeX code for tree is in nn.tex

The tree is shown in Figure 6. Although it is shorter, it misclassifies 6 more observations than if the default option is used. Note that the observations in each terminal node are not necessarily predicted to belong to the same class.

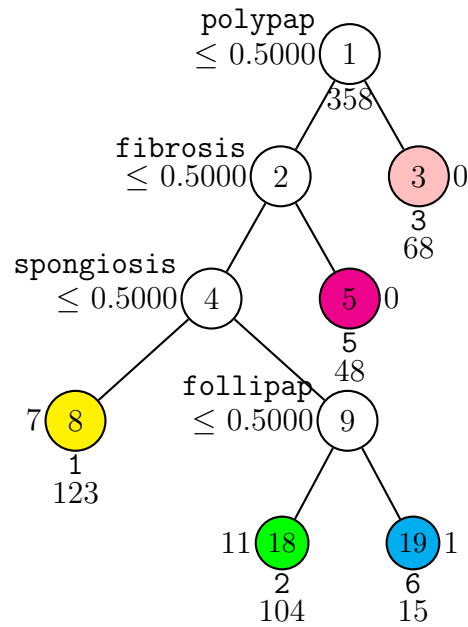


Figure 6: GUIDE v.29.6 0.50-SE pruned classification tree for predicting **class** using univariate nearest-neighbor node models, estimated priors and unit misclassification costs. Maximum number of split levels is 10 and minimum node sample size is 10. At each split, an observation goes to the left branch if and only if the condition is satisfied. Predicted classes and sample sizes printed below terminal nodes; #misclassified beside nodes. Second best split variable at root node is **bandlike**.

4.6.3 Kernel density option

Another alternative is kernel discrimination models, where classification is based on maximum likelihood with class densities estimated by the kernel method. Unlike nearest-neighbor, however, this option also yields an estimated class probability vector for each observation. Therefore it can serve as a nonparametric alternative to multinomial logistic regression. Empirical evidence indicates that the nearest-neighbor and kernel methods possess similar prediction accuracy. See [Loh \(2009\)](#) for more details. Following is a log of the input file generation step for the kernel method.

Input file creation

```

0. Read the warranty disclaimer
1. Create an input file for model fitting or importance scoring (recommended)
2. Convert data to other formats without creating input file
Input your choice: 1
Name of batch input file: ker.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: ker.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
Input 1 for simple, 2 for nearest-neighbor, 3 for kernel method ([1:3], <cr>=1): 3
    This is where kernel density estimation is chosen.
Input 1 for univariate, 2 for bivariate preference ([1:2], <cr>=1):
Input 1 for interaction tests, 2 to skip them ([1:2], <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 characters);
enclose with matching quotes if it has spaces: derm.dsc
Reading data description file ...
Training sample file: derm.dat
Missing value code: ?
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is class
Reading data file ...
Number of records in data file: 358
Length of longest data entry: 2
Checking for missing values ...
Total number of cases: 358

```

```

Number of classes = 6
Re-checking data ...
Assigning codes to categorical and missing values
Finished checking data
Rereading data
Class      #Cases      Proportion
1           111      0.31005587
2           60      0.16759777
3           71      0.19832402
4           48      0.13407821
5           48      0.13407821
6           20      0.05586592
      Total #cases w/ #missing
      #cases miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
      358      0      0      0      0      0      34      0      0
No. cases used for training: 358
Finished reading data file
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 for 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):
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):
Default minimum node sample size is 10
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): ker.tex
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 terminal nodes, 2 otherwise ([1:2], <cr>=1):
Choose amount of detail in nodes of LaTeX tree diagram
Input 0 for #errors, 1 for class sizes, 2 for nothing ([0:2], <cr>=0):
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 individual fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):

```

```

Input name of file to store node ID and fitted value of each case: ker.fit
Input 2 to save terminal node IDs for importance scoring; 1 otherwise ([1:2], <cr>=1):
Input name of file to store predicted class and probability: ker.pro
This file contains the estimated class probabilities for each observation.
Input file is created!
Run GUIDE with the command: guide < ker.in

```

Results

```

Classification tree
Pruning by cross-validation
Data description file: derm.dsc
Training sample file: derm.dat
Missing value code: ?
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is class
Number of records in data file: 358
Length of longest entry in data file: 2
Number of classes: 6
Training sample class proportions of D variable class:

```

Class	#Cases	Proportion
1	111	0.31005587
2	60	0.16759777
3	71	0.19832402
4	48	0.13407821
5	48	0.13407821
6	20	0.05586592

Summary information (without x variables)

d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	erythema	s	0.000	3.000		
2	scaling	s	0.000	3.000		
3	borders	s	0.000	3.000		
4	itching	s	0.000	3.000		
5	koebner	s	0.000	3.000		
6	polypap	s	0.000	3.000		
7	follipap	s	0.000	3.000		
8	oralmuc	s	0.000	3.000		

9	knee	s	0.000	3.000
10	scalp	s	0.000	3.000
11	history	s	0.000	1.000
12	melanin	s	0.000	3.000
13	eosin	s	0.000	2.000
14	PNL	s	0.000	3.000
15	fibrosis	s	0.000	3.000
16	exocyto	s	0.000	3.000
17	acantho	s	0.000	3.000
18	hyperker	s	0.000	3.000
19	paraker	s	0.000	3.000
20	clubbing	s	0.000	3.000
21	elongation	s	0.000	3.000
22	thinning	s	0.000	3.000
23	spongiform	s	0.000	3.000
24	munro	s	0.000	3.000
25	hypergran	s	0.000	3.000
26	disappea	s	0.000	3.000
27	basal	s	0.000	3.000
28	spongiosis	s	0.000	3.000
29	sawtooth	s	0.000	3.000
30	hornplug	s	0.000	3.000
31	perifoll	s	0.000	3.000
32	inflamm	s	0.000	3.000
33	bandlike	s	0.000	3.000
34	age	s	0.000	75.00
35	class	d		

6

Total	#cases w/	#missing					
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var	
358	0	0	0	0	0	34	
#P-var	#M-var	#B-var	#C-var	#I-var			
0	0	0	0	0			

No. cases used for training: 358

Univariate split highest priority

Interaction splits 2nd priority; no linear splits

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

Selected tree is based on mean of CV estimates

Kernel density node models

Univariate preference

Estimated priors

Unit misclassification costs

Split values for N and S variables based on exhaustive search

Max. number of split levels: 10

Min. node sample size: 10

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	9	8.380E-02	1.464E-02	2.073E-02	6.944E-02	2.149E-02
2	8	8.380E-02	1.464E-02	2.073E-02	6.944E-02	2.149E-02
3	7	8.380E-02	1.464E-02	2.073E-02	6.944E-02	2.149E-02
4	6	8.380E-02	1.464E-02	2.073E-02	6.944E-02	2.149E-02
5**	5	7.263E-02	1.372E-02	1.824E-02	5.556E-02	1.440E-02
6	3	2.039E-01	2.129E-02	2.928E-02	1.829E-01	2.672E-02
7	1	5.056E-01	2.642E-02	2.881E-02	5.000E-01	3.310E-02

0-SE tree based on mean is marked with * and has 5 terminal nodes

0-SE tree based on median is marked with + and has 5 terminal nodes

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.

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

Node label	Total cases	Train cases	Predicted class	Node cost	Split variable followed by (+)fit variable(s)
1	358	358	1	5.000E-01	polypap +polypap
2	290	290	1	4.517E-01	fibrosis +fibrosis
4	242	242	1	2.851E-01	spongiosis +spongiosis
8T	123	123	1	7.317E-02	elongation +elongation
9	119	119	2	4.874E-01	follipap +follipap
18T	104	104	2	1.058E-01	koebner +koebner
19T	15	15	6	6.667E-02	-
5T	48	48	5	0.000E+00	-
3T	68	68	3	0.000E+00	-

“Split variable” refers to the variable selected to split the node and “fit variable(s)” refers to the one(s) used to estimate the class kernel densities. Fit variables are indicated with a preceding + sign.

If a categorical variable is selected for fitting, discrete kernel density estimates are used. A dash (-) indicates that a node is not split, usually due to sample size being too small, in which case all the observations in the node are predicted as belonging to the class that minimizes the misclassification cost.

Number of terminal nodes of final tree: 5

Total number of nodes of final tree: 9

Second best split variable (based on curvature test) at root node is bandlike

Classification tree:

```

Node 1: polypap <= 0.50000000
  Node 2: fibrosis <= 0.50000000
    Node 4: spongiosis <= 0.50000000
      Node 8: Mean cost = 0.73170732E-001
      Node 4: spongiosis > 0.50000000 or ?
        Node 9: follipap <= 0.50000000
          Node 18: Mean cost = 0.10576923
          Node 9: follipap > 0.50000000 or ?
            Node 19: Mean cost = 0.66666667E-001
        Node 2: fibrosis > 0.50000000 or ?
          Node 5: Mean cost = 0.00000000
    Node 1: polypap > 0.50000000 or ?
      Node 3: Mean cost = 0.00000000

```

Node 1: Intermediate node

A case goes into Node 2 if polypap <= 0.50000000
 polypap mean = 0.44972067

Class	Number	Posterior	Bandwidth polypap
1	111	0.31006	3.6127E-02
2	60	0.16760	4.0857E-02
3	71	0.19832	3.9504E-01
4	48	0.13408	4.2722E-02
5	48	0.13408	4.2722E-02
6	20	0.05587	5.0897E-02

Number of training cases misclassified = 179

If node model is inapplicable due to missing values, predicted class is "1"
The numbers in the last column give the kernel density bandwidth for each class.

 Node 2: Intermediate node

A case goes into Node 4 if fibrosis <= 0.50000000
 fibrosis mean = 0.37586207

Class	Number	Posterior	Bandwidth fibrosis
1	111	0.38276	3.6127E-02
2	60	0.20690	4.0857E-02
3	3	0.01034	7.4383E-02
4	48	0.16552	4.2722E-02
5	48	0.16552	4.2722E-01

```

6           20      0.06897  5.0897E-02
Number of training cases misclassified = 131
If node model is inapplicable due to missing values, predicted class is "1"
-----

```

Node 4: Intermediate node

```

A case goes into Node 8 if spongiosis <= 0.50000000
spongiosis mean = 1.0537190

```

Class	Number	Posterior	Bandwidth spongiosis
1	111	0.45868	7.6519E-02
2	60	0.24793	4.0857E-01
3	3	0.01240	2.2315E+00
4	48	0.19835	7.5190E-01
5	0	0.00000	0.0000E+00
6	20	0.08264	1.3804E+00

```

Number of training cases misclassified = 69
If node model is inapplicable due to missing values, predicted class is "1"
-----

```

Node 8: Terminal node

```

elongation mean = 2.0569106

```

Class	Number	Posterior	Bandwidth elongation
1	111	0.90244	3.6127E-01
2	3	0.02439	7.8156E-02
3	2	0.01626	8.4758E-02
4	1	0.00813	9.7362E-02
5	0	0.00000	0.0000E+00
6	6	0.04878	7.1324E-01

Node 9: Intermediate node

```

A case goes into Node 18 if follipap <= 0.50000000
follipap mean = 0.25210084

```

Class	Number	Posterior	Bandwidth follipap
1	0	0.00000	0.0000E+00
2	57	0.47899	1.4751E-01
3	1	0.00840	9.3523E-02
4	47	0.39496	4.3301E-02
5	0	0.00000	0.0000E+00
6	14	0.11765	9.0804E-01

```

Number of training cases misclassified = 58
If node model is inapplicable due to missing values, predicted class is "2"
-----

```

Node 18: Terminal node

```

koebner mean = 0.53846154

```

Bandwidth

Class	Number	Posterior	koebner
1	0	0.00000	0.0000E+00
2	56	0.53846	2.9870E-01
3	1	0.00962	1.2607E-01
4	47	0.45192	8.5804E-01
5	0	0.00000	0.0000E+00
6	0	0.00000	0.0000E+00

Node 19: Terminal node

Class	Number	Posterior
1	0	0.00000
2	1	0.06667
3	0	0.00000
4	0	0.00000
5	0	0.00000
6	14	0.93333

Node 5: Terminal node

Class	Number	Posterior
1	0	0.00000
2	0	0.00000
3	0	0.00000
4	0	0.00000
5	48	1.00000
6	0	0.00000

Node 3: Terminal node

Class	Number	Posterior
1	0	0.00000
2	0	0.00000
3	68	1.00000
4	0	0.00000
5	0	0.00000
6	0	0.00000

Classification matrix for training sample:

Predicted	True class					
class	1	2	3	4	5	6
1	111	0	0	0	0	1
2	0	58	3	10	0	5
3	0	0	68	0	0	0
4	0	1	0	38	0	0
5	0	0	0	0	48	0
6	0	1	0	0	0	14

Total	111	60	71	48	48	20
-------	-----	----	----	----	----	----

Number of cases used for tree construction: 358

Number misclassified: 21

Resubstitution est. of mean misclassification cost: 0.58659218E-001

Predicted class probability estimates are stored in `ker.pro`

Observed and fitted values are stored in `ker.fit`

LaTeX code for tree is in `ker.tex`

The tree is the same as the one in Figure 6. Unlike the nearest-neighbor option, the kernel option can provide an estimated class probability vector for each observation. These are contained in the file `ker.pro`, the top few lines of which are given below. For example, the probabilities that the 1st observation belongs to classes 1–6 are (0, 0.876, 0, 0.239, 0, 0). The last two columns give the predicted and observed class of the observation.

"1"	"2"	"3"	"4"	"5"	"6"	predicted	observed
0.00000	0.84423	0.03637	0.11940	0.00000	0.00000	"2"	"2"
0.99616	0.00000	0.00000	0.00000	0.00000	0.00384	"1"	"1"
0.00000	0.00000	1.00000	0.00000	0.00000	0.00000	"3"	"3"
0.99616	0.00000	0.00000	0.00000	0.00000	0.00384	"1"	"1"
0.00000	0.00000	1.00000	0.00000	0.00000	0.00000	"3"	"3"

4.7 More than 2 classes: heart disease, categorical predictors and nodes without labels

CART and algorithms derived from it tend to be overly aggressive in their search for splits. As a consequence, they have two significant weaknesses: (i) bias towards selecting variables that allow more splits and (ii) long computational times when there are categorical predictor variables with many categorical levels. These problems are demonstrated by the heart disease data in the file `heartdata.txt`. The GUIDE description file is `heartdsc.txt` and the class variable is `num`, an integer-valued code (0–4) denoting a diagnosis of heart disease. There are 52 predictor variables, of which 29 are ordinal and 23 are categorical. Among the latter are the `ekgmo` and `ekgday`, the month and day of the EKG, with 12 and 31 categorical levels, respectively. The number of records is 617. They are obtained by combining the Hungarian, Long-beach and Switzerland datasets from the UCI (Ilter and Guvenir, 1998) database of the same name.

4.7.1 Input file creation

If a tree is quite large, as will be seen below, it is often preferable not to number the nodes of the tree. The following dialog shows how to do this.

```

0. Read the warranty disclaimer
1. Create an input file for model fitting, importance scoring or data formatting
2. Convert data to other formats without creating input file
Input your choice: 1
Name of batch input file: heartin.txt
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: heartout.txt
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
    Option 2 allows node labels to be omitted.
Input 1 for simple, 2 for nearest-neighbor, 3 for kernel method ([1:3], <cr>=1):
Input 0 for linear, interaction and univariate splits (in this order),
    1 for univariate, linear and interaction splits (in this order),
    2 to skip linear splits,
    3 to skip linear and interaction splits:
Input your choice ([0:3], <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 characters);
enclose with matching quotes if it has spaces: heartdsc.txt
Reading data description file ...
Training sample file: heartdata.txt
Missing value code: NA
Records in data file start on line 2
Warning: N variables changed to S
Dependent variable is num
Reading data file ...
Number of records in data file: 617
Length of longest data entry: 9
Checking for missing values ...
Total number of cases: 617
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Number of classes: 5
Col. no. Categorical variable    #levels    #missing values
      2 sex                      2              0
      3 painloc                  2              0

```

4	painexer	2	0
5	relrest	2	4
6	cp	4	0
9	smoke	2	387
12	fbs	2	90
13	dm	2	545
14	famhist	2	422
15	restecg	3	2
16	ekgmo	12	53
17	ekgday	31	54
19	dig	2	66
20	prop	3	64
21	nitr	2	63
22	pro	2	61
23	diuretic	2	80
24	proto	14	112
33	exang	2	55
34	xhypo	2	58
36	slope	4	308
40	thal	7	475
53	database	3	0

Re-checking data ...

Allocating missing value information

Assigning codes to categorical and missing values

Data checks complete

Creating missing value indicators

Rereading data

Class	#Cases	Proportion
0	247	0.40032415
1	141	0.22852512
2	99	0.16045381
3	100	0.16207455
4	30	0.04862237

Total	#cases w/	#missing							
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var	#B-var	#C-var	
617	0	615	0	0	0	29	0	23	

No. cases used for training: 617

No. cases excluded due to 0 weight or missing D: 0

Finished reading data file

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

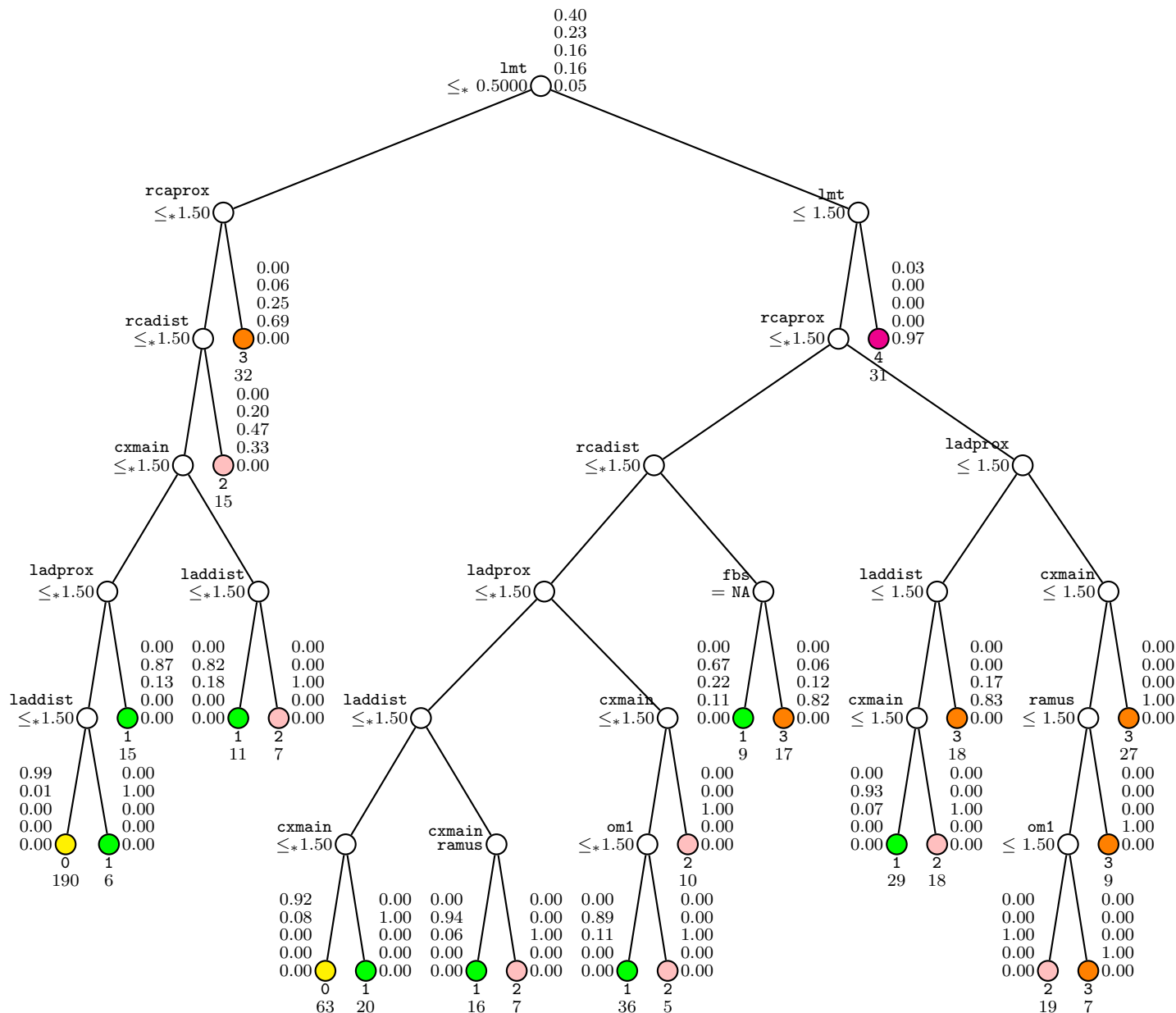
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):

Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50):

Choose 1 for estimated priors, 2 for equal priors, 3 for 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):
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):
Default minimum node sample size is 3
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): heart.tex
Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1): 2
This is where node labels are omitted.
Input 1 to color terminal nodes, 2 otherwise ([1:2], <cr>=1):
Choose amount of detail in nodes of LaTeX tree diagram
Input 0 for #errors, 1 for class proportions, 2 for nothing ([0:2], <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 individual fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: heart.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < heartin.txt



4.7.2 Results

The GUIDE tree is shown in Figure 7 and the text output follows. Despite the tree being quite large, no categorical variable is selected to split the nodes.

```

Classification tree
Pruning by cross-validation
Data description file: heartdsc.txt
Training sample file: heartdata.txt
Missing value code: NA
Records in data file start on line 2
Warning: N variables changed to S
Dependent variable is num
Number of records in data file: 617
Length of longest entry in data file: 9
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Number of classes: 5
Training sample class proportions of D variable num:

```

Class	#Cases	Proportion
0	247	0.40032415
1	141	0.22852512
2	99	0.16045381
3	100	0.16207455
4	30	0.04862237

Summary information (without x variables)

d=dependent, b=split and fit cat variable using indicator variables,
 c=split-only categorical, i=fit-only categorical (via indicators),
 s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
 m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	age	s	28.00	77.00		
2	sex	c			2	
3	painloc	c			2	
4	painexer	c			2	
5	relrest	c			2	4
6	cp	c			4	
7	trestbps	s	0.000	200.0		59
8	chol	s	0.000	603.0		30
9	smoke	c			2	387
10	cigs	s	0.000	80.00		415
11	years	s	0.000	60.00		427
12	fbs	c			2	90

13	dm	c			2	545
14	famhist	c			2	422
15	restecg	c			3	2
16	ekgmo	c			12	53
17	ekgday	c			31	54
18	ekgyr	s	81.00	87.00		53
19	dig	c			2	66
20	prop	c			3	64
21	nitr	c			2	63
22	pro	c			2	61
23	diuretic	c			2	80
24	proto	c			14	112
25	thaldur	s	1.000	24.00		56
26	thaltime	s	0.000	20.00		384
27	met	s	2.000	200.0		105
28	thalach	s	60.00	190.0		55
29	thalrest	s	37.00	139.0		56
30	tpeakbps	s	100.0	240.0		63
31	tpeakbpd	s	11.00	134.0		63
32	trestbpd	s	0.000	120.0		59
33	exang	c			2	55
34	xhypo	c			2	58
35	oldpeak	s	-2.600	5.000		62
36	slope	c			4	308
37	rldv5	s	2.000	36.00		143
38	rldv5e	s	2.000	36.00		142
39	ca	s	0.000	9.000		606
40	thal	c			7	475
41	cyr	s	1.000	87.00		9
42	num	d			5	
43	lmt	s	0.000	162.0		275
44	ladprox	s	1.000	2.000		236
45	laddist	s	1.000	2.000		246
46	diag	s	1.000	2.000		276
47	cxmain	s	1.000	2.000		235
48	ramus	s	1.000	2.000		285
49	om1	s	1.000	2.000		271
50	om2	s	1.000	2.000		290
51	rcaprox	s	1.000	2.000		245
52	rcadist	s	1.000	2.000		270
53	database	c			3	

Total	#cases w/	#missing				
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var
617	0	615	0	0	0	29
#P-var	#M-var	#B-var	#C-var	#I-var		

```

      0      0      0      23      0
No. cases used for training: 617
No. cases excluded due to 0 weight or missing D: 0

```

```

Univariate split highest priority
Interaction and linear splits 2nd and 3rd priorities
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
Min. node sample size: 3
Number of SE's for pruned tree: 0.5000

```

Size and CV mean cost and SE of subtrees:

Tree	#Tnodes	Mean Cost	SE(Mean)	BSE(Mean)	Median Cost	BSE(Median)
1	36	1.086E-01	1.253E-02	1.633E-02	1.056E-01	1.770E-02
2	35	1.086E-01	1.253E-02	1.633E-02	1.056E-01	1.770E-02
3	34	1.086E-01	1.253E-02	1.633E-02	1.056E-01	1.770E-02
4	32	1.053E-01	1.236E-02	1.568E-02	9.744E-02	1.604E-02
5	31	1.037E-01	1.228E-02	1.552E-02	9.744E-02	1.521E-02
6	29	1.037E-01	1.228E-02	1.552E-02	9.744E-02	1.521E-02
7	28	1.037E-01	1.228E-02	1.552E-02	9.744E-02	1.521E-02
8	27	1.021E-01	1.219E-02	1.582E-02	9.677E-02	1.700E-02
9	26	1.005E-01	1.210E-02	1.651E-02	9.677E-02	1.702E-02
10*	25	1.005E-01	1.210E-02	1.651E-02	9.677E-02	1.702E-02
11**	24	1.053E-01	1.236E-02	1.046E-02	1.048E-01	1.137E-02
12--	23	1.086E-01	1.253E-02	9.609E-03	1.048E-01	8.889E-03
13++	22	1.118E-01	1.269E-02	1.013E-02	1.048E-01	1.146E-02
14	21	1.345E-01	1.374E-02	1.118E-02	1.371E-01	2.010E-02
15	18	1.491E-01	1.434E-02	8.664E-03	1.452E-01	1.016E-02
16	17	1.572E-01	1.465E-02	7.460E-03	1.464E-01	1.102E-02
17	16	1.605E-01	1.478E-02	9.529E-03	1.464E-01	1.117E-02
18	13	1.750E-01	1.530E-02	7.772E-03	1.774E-01	1.012E-02
19	12	1.912E-01	1.583E-02	7.351E-03	1.855E-01	1.150E-02
20	9	2.982E-01	1.842E-02	2.724E-02	3.065E-01	4.174E-02
21	5	3.922E-01	1.966E-02	7.611E-03	3.983E-01	7.443E-03
22	4	3.922E-01	1.966E-02	7.611E-03	3.983E-01	7.443E-03
23	2	5.251E-01	2.010E-02	6.049E-03	5.246E-01	8.589E-03
24	1	5.997E-01	1.973E-02	3.358E-03	5.968E-01	5.312E-03

0-SE tree based on mean is marked with * and has 25 terminal nodes

0-SE tree based on median is marked with + and has 25 terminal nodes

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.

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

Node label	Total cases	Train cases	Predicted class	Node cost	Split variables	Interacting variable
1	617	617	0	5.997E-01	lmt	
2	276	276	0	3.188E-01	rcaprox	
4	244	244	0	2.295E-01	rcadist	
8	229	229	0	1.790E-01	cxmain	
16	211	211	0	1.090E-01	ladprox	
32	196	196	0	4.082E-02	laddist	
64T	190	190	0	1.053E-02	om1	
65T	6	6	1	0.000E+00	-	
33T	15	15	1	1.333E-01	trestbps +tpeakbpd	
17	18	18	2	5.000E-01	laddist	
34T	11	11	1	1.818E-01	trestbps	
35T	7	7	2	0.000E+00	-	
9T	15	15	2	5.333E-01	-	
5T	32	32	3	3.125E-01	tpeakbpd :fbs	
3	341	341	1	6.891E-01	lmt	
6	310	310	1	6.581E-01	rcaprox	
12	183	183	1	5.683E-01	rcadist	
24	157	157	1	5.414E-01	ladprox	
48	106	106	0	4.528E-01	laddist	
96	83	83	0	3.012E-01	cxmain	
192T	63	63	0	7.937E-02	om1	
193T	20	20	1	0.000E+00	-	
97	23	23	1	3.478E-01	cxmain +ramus	
194T	16	16	1	6.250E-02	-	
195T	7	7	2	0.000E+00	-	
49	51	51	1	3.725E-01	cxmain	
98	41	41	1	2.195E-01	om1	
196T	36	36	1	1.111E-01	om2	
197T	5	5	2	0.000E+00	-	
99T	10	10	2	0.000E+00	-	
25	26	26	3	4.231E-01	fbs	
50T	9	9	1	3.333E-01	thalach	
51T	17	17	3	1.765E-01	exang	
13	127	127	3	5.433E-01	ladprox	
26	65	65	1	5.846E-01	laddist	

52	47	47	1	4.255E-01	cxmain
104T	29	29	1	6.897E-02	thalach
105T	18	18	2	0.000E+00	-
53T	18	18	3	1.667E-01	cxmain
27	62	62	3	3.065E-01	cxmain
54	35	35	2	4.571E-01	ramus
108	26	26	2	2.692E-01	om1
216T	19	19	2	0.000E+00	-
217T	7	7	3	0.000E+00	-
109T	9	9	3	0.000E+00	-
55T	27	27	3	0.000E+00	-
7T	31	31	4	3.226E-02	-

Number of terminal nodes of final tree: 24

Total number of nodes of final tree: 47

Second best split variable (based on curvature test) at root node is rcaprox

Classification tree:

At splits on categorical variables, values not in training data go to the right

Node 1: lmt <= 0.50000000 or NA

Node 2: rcaprox <= 1.5000000 or NA

Node 4: rcadist <= 1.5000000 or NA

Node 8: cxmain <= 1.5000000 or NA

Node 16: ladprox <= 1.5000000 or NA

Node 32: laddist <= 1.5000000 or NA

Node 64: 0

Node 32: laddist > 1.5000000

Node 65: 1

Node 16: ladprox > 1.5000000

Node 33: 1

Node 8: cxmain > 1.5000000

Node 17: laddist <= 1.5000000 or NA

Node 34: 1

Node 17: laddist > 1.5000000

Node 35: 2

Node 4: rcadist > 1.5000000

Node 9: 2

Node 2: rcaprox > 1.5000000

Node 5: 3

Node 1: lmt > 0.50000000

Node 3: lmt <= 1.5000000

Node 6: rcaprox <= 1.5000000 or NA

Node 12: rcadist <= 1.5000000 or NA

Node 24: ladprox <= 1.5000000 or NA

Node 48: laddist <= 1.5000000 or NA

```

Node 96: cxmain <= 1.5000000 or NA
Node 192: 0
Node 96: cxmain > 1.5000000
Node 193: 1
Node 48: laddist > 1.5000000
Node 97: 1.0000000 * ramus + cxmain <= 2.5000000
Node 194: 1
Node 97: 1.0000000 * ramus + cxmain > 2.5000000 or NA
Node 195: 2
Node 24: ladprox > 1.5000000
Node 49: cxmain <= 1.5000000 or NA
Node 98: om1 <= 1.5000000 or NA
Node 196: 1
Node 98: om1 > 1.5000000
Node 197: 2
Node 49: cxmain > 1.5000000
Node 99: 2
Node 12: rcadist > 1.5000000
Node 25: fbs = "NA"
Node 50: 1
Node 25: fbs != "NA"
Node 51: 3
Node 6: rcaprox > 1.5000000
Node 13: ladprox <= 1.5000000
Node 26: laddist <= 1.5000000
Node 52: cxmain <= 1.5000000
Node 104: 1
Node 52: cxmain > 1.5000000 or NA
Node 105: 2
Node 26: laddist > 1.5000000 or NA
Node 53: 3
Node 13: ladprox > 1.5000000 or NA
Node 27: cxmain <= 1.5000000
Node 54: ramus <= 1.5000000
Node 108: om1 <= 1.5000000
Node 216: 2
Node 108: om1 > 1.5000000 or NA
Node 217: 3
Node 54: ramus > 1.5000000 or NA
Node 109: 3
Node 27: cxmain > 1.5000000 or NA
Node 55: 3
Node 3: lmt > 1.5000000 or NA
Node 7: 4

```

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

Node 1: Intermediate node

A case goes into Node 2 if $\text{limt} \leq 0.50000000$ or NA

limt mean = 1.5555556

Class	Number	Posterior
0	247	0.40032
1	141	0.22853
2	99	0.16045
3	100	0.16207
4	30	0.04862

Number of training cases misclassified = 370

Predicted class is 0

Node 2: Intermediate node

A case goes into Node 4 if $\text{rcaprox} \leq 1.50000000$ or NA

rcaprox mean = 1.7619048

Class	Number	Posterior
0	188	0.68116
1	35	0.12681
2	26	0.09420
3	27	0.09783
4	0	0.00000

Number of training cases misclassified = 88

Predicted class is 0

Node 4: Intermediate node

A case goes into Node 8 if $\text{rcadist} \leq 1.50000000$ or NA

rcadist mean = 1.7894737

Class	Number	Posterior
0	188	0.77049
1	33	0.13525
2	18	0.07377
3	5	0.02049
4	0	0.00000

Number of training cases misclassified = 56

Predicted class is 0

Node 8: Intermediate node

A case goes into Node 16 if $\text{cxmain} \leq 1.50000000$ or NA

cxmain mean = 1.6923077

Class	Number	Posterior
0	188	0.82096
1	30	0.13100
2	11	0.04803

```

3           0      0.00000
4           0      0.00000
Number of training cases misclassified = 41
Predicted class is 0
-----

```

Node 16: Intermediate node

A case goes into Node 32 if ladprox \leq 1.5000000 or NA
 ladprox mean = 1.6818182

Class	Number	Posterior
0	188	0.89100
1	21	0.09953
2	2	0.00948
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 23
 Predicted class is 0

Node 32: Intermediate node

A case goes into Node 64 if laddist \leq 1.5000000 or NA
 laddist mean = 1.2857143

Class	Number	Posterior
0	188	0.95918
1	8	0.04082
2	0	0.00000
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 8
 Predicted class is 0

Node 64: Terminal node

Class	Number	Posterior
0	188	0.98947
1	2	0.01053
2	0	0.00000
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 2
 Predicted class is 0

Node 65: Terminal node

Class	Number	Posterior
0	0	0.00000
1	6	1.00000
2	0	0.00000
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 0
 Predicted class is 1

Node 33: Terminal node

Class	Number	Posterior
0	0	0.00000
1	13	0.86667
2	2	0.13333
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 2
 Predicted class is 1

Node 17: Intermediate node

A case goes into Node 34 if laddist <= 1.5000000 or NA
 laddist mean = 1.8750000

Class	Number	Posterior
0	0	0.00000
1	9	0.50000
2	9	0.50000
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 9
 Predicted class is 2

Node 34: Terminal node

Class	Number	Posterior
0	0	0.00000
1	9	0.81818
2	2	0.18182
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 2
 Predicted class is 1

Node 35: Terminal node

Class	Number	Posterior
0	0	0.00000
1	0	0.00000
2	7	1.00000
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 0
 Predicted class is 2

Node 9: Terminal node

Class	Number	Posterior
0	0	0.00000
1	3	0.20000
2	7	0.46667
3	5	0.33333
4	0	0.00000

Number of training cases misclassified = 8
 Predicted class is 2

Node 5: Terminal node

Class	Number	Posterior
0	0	0.00000
1	2	0.06250
2	8	0.25000
3	22	0.68750
4	0	0.00000

Number of training cases misclassified = 10
 Predicted class is 3

Node 3: Intermediate node

A case goes into Node 6 if $\text{lmt} \leq 1.5000000$
 $\text{lmt mean} = 1.5601173$

Class	Number	Posterior
0	59	0.17302
1	106	0.31085
2	73	0.21408
3	73	0.21408
4	30	0.08798

Number of training cases misclassified = 235
 Predicted class is 1

Node 6: Intermediate node

A case goes into Node 12 if $\text{rcaprox} \leq 1.5000000$ or NA
 $\text{rcaprox mean} = 1.4136808$

Class	Number	Posterior
0	58	0.18710
1	106	0.34194
2	73	0.23548
3	73	0.23548
4	0	0.00000

Number of training cases misclassified = 204
 Predicted class is 1

Node 12: Intermediate node

A case goes into Node 24 if $\text{rcadist} \leq 1.5000000$ or NA
 $\text{rcadist mean} = 1.1444444$

Class	Number	Posterior
0	58	0.31694
1	79	0.43169
2	31	0.16940
3	15	0.08197
4	0	0.00000

Number of training cases misclassified = 104
 Predicted class is 1

Node 24: Intermediate node

A case goes into Node 48 if ladprox \leq 1.5000000 or NA
 ladprox mean = 1.3290323

Class	Number	Posterior
0	58	0.36943
1	72	0.45860
2	27	0.17197
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 85
 Predicted class is 1

Node 48: Intermediate node

A case goes into Node 96 if laddist \leq 1.5000000 or NA
 laddist mean = 1.2211538

Class	Number	Posterior
0	58	0.54717
1	40	0.37736
2	8	0.07547
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 48
 Predicted class is 0

Node 96: Intermediate node

A case goes into Node 192 if cxmain \leq 1.5000000 or NA
 cxmain mean = 1.2439024

Class	Number	Posterior
0	58	0.69880
1	25	0.30120
2	0	0.00000
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 25
 Predicted class is 0

Node 192: Terminal node

Class	Number	Posterior
0	58	0.92063
1	5	0.07937
2	0	0.00000
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 5
 Predicted class is 0

Node 193: Terminal node

Class	Number	Posterior
0	0	0.00000
1	20	1.00000
2	0	0.00000
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 0
 Predicted class is 1

Node 97: Intermediate node

A case goes into Node 194 if
 $1.0000000 * \text{ramus} + \text{cxmain} \leq 2.5000000$
 Linear combination mean = 2.3043478

Class	Number	Posterior
0	0	0.00000
1	15	0.65217
2	8	0.34783
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 8
 Predicted class is 1

Node 194: Terminal node

Class	Number	Posterior
0	0	0.00000
1	15	0.93750
2	1	0.06250
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 1
 Predicted class is 1

Node 195: Terminal node

Class	Number	Posterior
0	0	0.00000
1	0	0.00000

```

2          7      1.00000
3          0      0.00000
4          0      0.00000

```

Number of training cases misclassified = 0

Predicted class is 2

Node 49: Intermediate node

A case goes into Node 98 if cxmain <= 1.5000000 or NA

cxmain mean = 1.2000000

```

Class      Number  Posterior
0           0      0.00000
1          32      0.62745
2          19      0.37255
3           0      0.00000
4           0      0.00000

```

Number of training cases misclassified = 19

Predicted class is 1

Node 98: Intermediate node

A case goes into Node 196 if om1 <= 1.5000000 or NA

om1 mean = 1.1250000

```

Class      Number  Posterior
0           0      0.00000
1          32      0.78049
2           9      0.21951
3           0      0.00000
4           0      0.00000

```

Number of training cases misclassified = 9

Predicted class is 1

Node 196: Terminal node

```

Class      Number  Posterior
0           0      0.00000
1          32      0.88889
2           4      0.11111
3           0      0.00000
4           0      0.00000

```

Number of training cases misclassified = 4

Predicted class is 1

Node 197: Terminal node

```

Class      Number  Posterior
0           0      0.00000
1           0      0.00000
2           5      1.00000
3           0      0.00000

```

```

4           0      0.00000
Number of training cases misclassified = 0
Predicted class is 2
-----
Node 99: Terminal node
Class      Number  Posterior
0           0      0.00000
1           0      0.00000
2          10      1.00000
3           0      0.00000
4           0      0.00000
Number of training cases misclassified = 0
Predicted class is 2
-----
Node 25: Intermediate node
A case goes into Node 50 if fbs = "NA"
fbs mode = "0"
Class      Number  Posterior
0           0      0.00000
1           7      0.26923
2           4      0.15385
3          15      0.57692
4           0      0.00000
Number of training cases misclassified = 11
Predicted class is 3
-----
Node 50: Terminal node
Class      Number  Posterior
0           0      0.00000
1           6      0.66667
2           2      0.22222
3           1      0.11111
4           0      0.00000
Number of training cases misclassified = 3
Predicted class is 1
-----
Node 51: Terminal node
Class      Number  Posterior
0           0      0.00000
1           1      0.05882
2           2      0.11765
3          14      0.82353
4           0      0.00000
Number of training cases misclassified = 3
Predicted class is 3
-----

```


Node 13: Intermediate node

A case goes into Node 26 if ladprox \leq 1.5000000

ladprox mean = 1.4881890

Class	Number	Posterior
0	0	0.00000
1	27	0.21260
2	42	0.33071
3	58	0.45669
4	0	0.00000

Number of training cases misclassified = 69

Predicted class is 3

Node 26: Intermediate node

A case goes into Node 52 if laddist \leq 1.5000000

laddist mean = 1.2769231

Class	Number	Posterior
0	0	0.00000
1	27	0.41538
2	23	0.35385
3	15	0.23077
4	0	0.00000

Number of training cases misclassified = 38

Predicted class is 1

Node 52: Intermediate node

A case goes into Node 104 if cxmain \leq 1.5000000

cxmain mean = 1.3829787

Class	Number	Posterior
0	0	0.00000
1	27	0.57447
2	20	0.42553
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 20

Predicted class is 1

Node 104: Terminal node

Class	Number	Posterior
0	0	0.00000
1	27	0.93103
2	2	0.06897
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 2

Predicted class is 1

Node 105: Terminal node

Class	Number	Posterior
0	0	0.00000
1	0	0.00000
2	18	1.00000
3	0	0.00000
4	0	0.00000

Number of training cases misclassified = 0

Predicted class is 2

Node 53: Terminal node

Class	Number	Posterior
0	0	0.00000
1	0	0.00000
2	3	0.16667
3	15	0.83333
4	0	0.00000

Number of training cases misclassified = 3

Predicted class is 3

Node 27: Intermediate node

A case goes into Node 54 if $cxmain \leq 1.5000000$

$cxmain$ mean = 1.4354839

Class	Number	Posterior
0	0	0.00000
1	0	0.00000
2	19	0.30645
3	43	0.69355
4	0	0.00000

Number of training cases misclassified = 19

Predicted class is 3

Node 54: Intermediate node

A case goes into Node 108 if $ramus \leq 1.5000000$

$ramus$ mean = 1.2571429

Class	Number	Posterior
0	0	0.00000
1	0	0.00000
2	19	0.54286
3	16	0.45714
4	0	0.00000

Number of training cases misclassified = 16

Predicted class is 2

Node 108: Intermediate node

A case goes into Node 216 if $om1 \leq 1.5000000$

```

om1 mean = 1.2692308
Class      Number  Posterior
0           0      0.00000
1           0      0.00000
2          19      0.73077
3           7      0.26923
4           0      0.00000
Number of training cases misclassified = 7
Predicted class is 2
-----
Node 216: Terminal node
Class      Number  Posterior
0           0      0.00000
1           0      0.00000
2          19      1.00000
3           0      0.00000
4           0      0.00000
Number of training cases misclassified = 0
Predicted class is 2
-----
Node 217: Terminal node
Class      Number  Posterior
0           0      0.00000
1           0      0.00000
2           0      0.00000
3           7      1.00000
4           0      0.00000
Number of training cases misclassified = 0
Predicted class is 3
-----
Node 109: Terminal node
Class      Number  Posterior
0           0      0.00000
1           0      0.00000
2           0      0.00000
3           9      1.00000
4           0      0.00000
Number of training cases misclassified = 0
Predicted class is 3
-----
Node 55: Terminal node
Class      Number  Posterior
0           0      0.00000
1           0      0.00000
2           0      0.00000
3          27      1.00000

```

```

4          0      0.00000
Number of training cases misclassified = 0
Predicted class is 3
-----
Node 7: Terminal node
Class      Number  Posterior
0          1      0.03226
1          0      0.00000
2          0      0.00000
3          0      0.00000
4         30      0.96774
Number of training cases misclassified = 1
Predicted class is 4
-----

Classification matrix for training sample:
Predicted   True class
class       0         1         2         3         4
0          246         7         0         0         0
1           0        128        13         1         0
2           0         3         73         5         0
3           0         3         13        94         0
4           1         0         0         0        30
Total      247        141         99        100        30

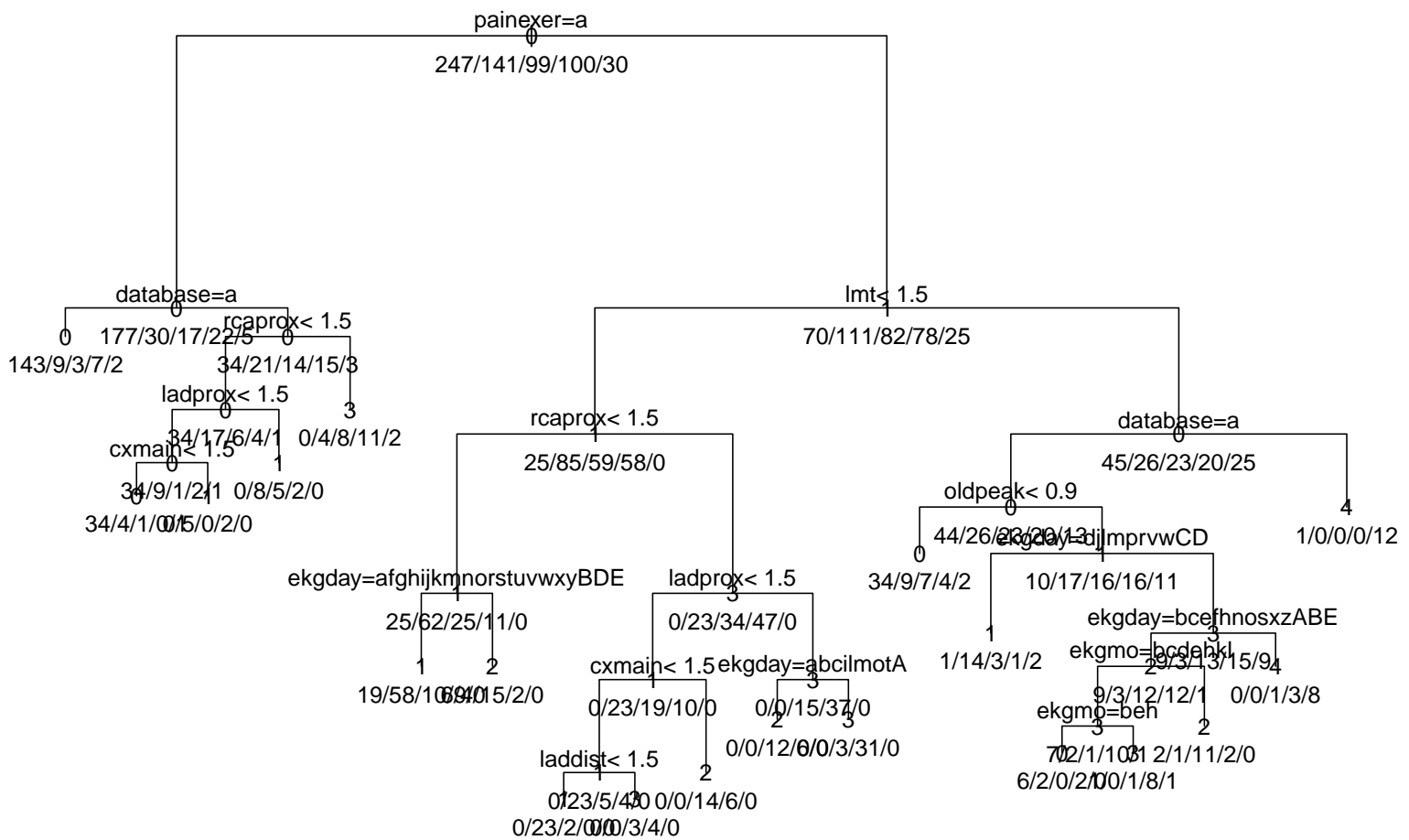
Number of cases used for tree construction: 617
Number misclassified: 46
Resubstitution est. of mean misclassification cost: 0.74554295E-001

Observed and fitted values are stored in heart.fit
LaTeX code for tree is in heart.tex

```

4.7.3 RPART model

The GUIDE model in Figure 7 took 3 sec. to construct on a Linux computer. In comparison, RPART (Therneau et al., 2017) took more than 3.5 hrs, due primarily to the presence of the categorical variables `ekgmo` and `ekgday`. The result is shown in Figure 8. It splits repeatedly on `ekgmo` and `ekgday`.



5 Regression

GUIDE can fit least-squares (LS), quantile, Poisson, proportional hazards, and least-median-of-squares (LMS) regression tree models. We use the birthweight data in files `birthwt.dat` and `birthwt.dsc` to demonstrate LS models. The data consist of observations from 50,000 live births. They are a subset of a larger dataset analyzed in [Koenker and Hallock \(2001\)](#); see also [Koenker \(2005\)](#). The variables are `weight` (infant birth weight), `black` (indicator of black mother), `married` (indicator of married mother), `boy` (indicator of boy), `visit` (prenatal visit: 0 = no visits, 1 = visit in 2nd trimester, 2 = visit in last trimester, 3 = visit in 1st trimester), `ed` (Mother's education level: 0 = high school, 1 = some college, 2 = college, 3 = less than high school), `smoke` (indicator of smoking mother), `cigsper` (number of cigarettes smoked per day), `age` (mother's age), and `wtgain` (mother's weight gain during pregnancy). The contents of `birthwt.dsc` are:

```
birthwt.dat
NA
1
1 weight d
2 black c
3 married c
4 boy c
5 age n
6 smoke c
7 cigsper n
8 wtgain n
9 visit c
10 ed c
11 lowbwt x
```

The last variable `lowbwt` is a derived indicator of low birthweight not used here.

5.1 Least squares constant: birthwt data

5.1.1 Input file creation

The input file `cons.in` is obtained as follows. We select the non-default option to enable more selections to be provided.

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: cons.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
```

```

Name of batch output file: cons.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
  5=multiresponse or itemresponse, 6=longitudinal data (with 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:
Choose 0 for stepwise linear regression
Choose 1 for multiple regression (recommended if R variable is present,
  unless there are too many N, F or B variables when stepwise is better)
Choose 2 for best polynomial in one N or F variable + R (if present)
Choose 3 to fit a constant + R (if present)
0: stepwise linear, 1: multiple linear, 2: best polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <cr>=0): 3
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
We choose 2 to allow more options below.
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 characters);
enclose with matching quotes if it has spaces: birthwt.dsc
Reading data description file ...
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is weight
Reading data file ...
Number of records in data file: 50000
Length of longest entry in data file: 4
Checking for missing values ...
Total number of cases: 50000
Col. no. Categorical variable    #levels    #missing values
      2 black                    2              0
      3 married                  2              0
      4 boy                      2              0
      6 smoke                    2              0
      9 visit                    4              0
     10 ed                      4              0
Re-checking data ...
Assigning codes to categorical and missing values
Finished processing 5000 of 50000 observations

```

```

Finished processing 10000 of 50000 observations
Finished processing 15000 of 50000 observations
Finished processing 20000 of 50000 observations
Finished processing 25000 of 50000 observations
Finished processing 30000 of 50000 observations
Finished processing 35000 of 50000 observations
Finished processing 40000 of 50000 observations
Finished processing 45000 of 50000 observations
Finished processing 50000 of 50000 observations
Data checks complete
Rereading data
      Total  #cases w/  #missing
      #cases  miss. D  ord. vals  #X-var  #N-var  #F-var  #S-var  #B-var  #C-var
      50000      0      0      1      0      0      3      0      6
No weight variable in data file
No. cases used for training: 50000
Finished reading data file
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: 30
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 250
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): cons.tex
Input 1 to include node numbers, 2 to omit them ([1:2], <cr>=1): 2
Choose 2 to omit node numbers for large trees.
Choose a color for the terminal 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): 3
Choose 3 to save split variable information to a separate file.
Input file name: cons.var
Input 2 to save regressor names in a file, 1 otherwise ([1:2], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: cons.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < cons.in

```

5.1.2 Results

The contents of cons.out follow.

```

Least squares regression tree
Pruning by cross-validation
Data description file: birthwt.dsc
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is weight
Piecewise constant model
Number of records in data file: 50000
Length of longest entry in data file: 4

```

Summary information (without x variables)

d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	weight	d	240.0	6350.		
2	black	c			2	
3	married	c			2	
4	boy	c			2	
5	age	s	18.00	45.00		
6	smoke	c			2	
7	cigsper	s	0.000	60.00		

```

      8 wtgain      s      0.000      98.00
      9 visit      c
     10 ed         c

```

```

Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var
50000      0      0      1      0      0      3
#P-var #M-var #B-var #C-var #I-var
      0      0      0      6      0

```

No weight variable in data file

No. cases used for training: 50000

Nodewise 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: 30

Min. node sample size: 250

Number of SE's for pruned tree: 0.5000

Size and CV MSE and SE of subtrees:

Tree	#Tnodes	Mean MSE	SE(Mean)	BSE(Mean)	Median MSE	BSE(Median)
1	147	2.891E+05	2.800E+03	1.678E+03	2.875E+05	2.016E+03
2	146	2.891E+05	2.800E+03	1.678E+03	2.875E+05	2.016E+03
3	145	2.891E+05	2.800E+03	1.678E+03	2.875E+05	2.016E+03
4	144	2.891E+05	2.800E+03	1.678E+03	2.875E+05	2.017E+03
5	143	2.891E+05	2.800E+03	1.678E+03	2.875E+05	2.017E+03
6	142	2.891E+05	2.800E+03	1.678E+03	2.875E+05	2.017E+03
7	141	2.891E+05	2.800E+03	1.678E+03	2.875E+05	2.017E+03
8	140	2.891E+05	2.800E+03	1.679E+03	2.875E+05	2.019E+03
9	139	2.891E+05	2.800E+03	1.678E+03	2.875E+05	2.022E+03
10	138	2.891E+05	2.800E+03	1.678E+03	2.875E+05	2.022E+03
11	137	2.891E+05	2.800E+03	1.679E+03	2.875E+05	2.026E+03
12	136	2.891E+05	2.800E+03	1.679E+03	2.875E+05	2.026E+03
13	135	2.891E+05	2.800E+03	1.679E+03	2.875E+05	2.026E+03
14	134	2.891E+05	2.800E+03	1.679E+03	2.875E+05	2.026E+03
15	133	2.891E+05	2.800E+03	1.679E+03	2.875E+05	2.026E+03
16	132	2.891E+05	2.800E+03	1.678E+03	2.875E+05	2.025E+03
17	131	2.891E+05	2.800E+03	1.681E+03	2.875E+05	2.017E+03
18	130	2.891E+05	2.800E+03	1.683E+03	2.875E+05	2.021E+03
19	129	2.891E+05	2.800E+03	1.690E+03	2.875E+05	2.025E+03
20	128	2.891E+05	2.800E+03	1.689E+03	2.875E+05	2.024E+03
21	127	2.891E+05	2.800E+03	1.689E+03	2.875E+05	2.023E+03
22	126	2.891E+05	2.800E+03	1.689E+03	2.875E+05	2.023E+03
23	125	2.891E+05	2.800E+03	1.689E+03	2.875E+05	2.023E+03
24	123	2.891E+05	2.800E+03	1.689E+03	2.875E+05	2.023E+03

25	122	2.891E+05	2.800E+03	1.689E+03	2.875E+05	2.023E+03
26	121	2.891E+05	2.800E+03	1.687E+03	2.875E+05	2.026E+03
27	119	2.891E+05	2.800E+03	1.687E+03	2.875E+05	2.027E+03
28	118	2.891E+05	2.800E+03	1.687E+03	2.875E+05	2.029E+03
29	117	2.891E+05	2.800E+03	1.687E+03	2.875E+05	2.029E+03
30	116	2.891E+05	2.800E+03	1.687E+03	2.875E+05	2.029E+03
31	114	2.891E+05	2.800E+03	1.689E+03	2.874E+05	2.042E+03
32	113	2.891E+05	2.800E+03	1.689E+03	2.874E+05	2.034E+03
33	112	2.891E+05	2.801E+03	1.696E+03	2.875E+05	2.017E+03
34	111	2.891E+05	2.801E+03	1.700E+03	2.875E+05	2.014E+03
35	110	2.891E+05	2.801E+03	1.700E+03	2.875E+05	2.014E+03
36	108	2.891E+05	2.801E+03	1.701E+03	2.875E+05	2.014E+03
37	105	2.891E+05	2.800E+03	1.710E+03	2.875E+05	2.026E+03
38	104	2.891E+05	2.801E+03	1.710E+03	2.875E+05	2.016E+03
39	103	2.891E+05	2.801E+03	1.706E+03	2.875E+05	2.006E+03
40	102	2.891E+05	2.801E+03	1.705E+03	2.875E+05	2.008E+03
41	101	2.891E+05	2.801E+03	1.702E+03	2.876E+05	2.003E+03
42	99	2.891E+05	2.801E+03	1.701E+03	2.876E+05	1.999E+03
43	98	2.891E+05	2.801E+03	1.701E+03	2.876E+05	1.990E+03
44	97	2.891E+05	2.801E+03	1.701E+03	2.876E+05	1.990E+03
45	96	2.891E+05	2.801E+03	1.708E+03	2.875E+05	1.993E+03
46	95	2.891E+05	2.801E+03	1.717E+03	2.874E+05	2.004E+03
47	93	2.891E+05	2.800E+03	1.714E+03	2.874E+05	2.006E+03
48	92	2.891E+05	2.801E+03	1.713E+03	2.874E+05	2.003E+03
49	91	2.891E+05	2.800E+03	1.714E+03	2.874E+05	2.006E+03
50	89	2.891E+05	2.800E+03	1.714E+03	2.874E+05	2.006E+03
51	88	2.891E+05	2.800E+03	1.714E+03	2.874E+05	2.006E+03
52	87	2.891E+05	2.800E+03	1.715E+03	2.874E+05	2.008E+03
53	86	2.891E+05	2.800E+03	1.715E+03	2.874E+05	1.995E+03
54	85	2.891E+05	2.800E+03	1.715E+03	2.874E+05	1.995E+03
55	84	2.891E+05	2.800E+03	1.715E+03	2.874E+05	1.995E+03
56	83	2.891E+05	2.800E+03	1.716E+03	2.875E+05	1.995E+03
57	82	2.891E+05	2.800E+03	1.712E+03	2.875E+05	1.990E+03
58	81	2.891E+05	2.800E+03	1.715E+03	2.875E+05	1.990E+03
59	80	2.891E+05	2.801E+03	1.725E+03	2.875E+05	2.001E+03
60	78	2.891E+05	2.801E+03	1.724E+03	2.875E+05	1.992E+03
61	76	2.891E+05	2.801E+03	1.724E+03	2.875E+05	1.992E+03
62	75	2.891E+05	2.801E+03	1.724E+03	2.875E+05	1.992E+03
63	74	2.891E+05	2.801E+03	1.720E+03	2.875E+05	1.987E+03
64	72	2.891E+05	2.801E+03	1.720E+03	2.875E+05	1.987E+03
65	71	2.891E+05	2.802E+03	1.750E+03	2.875E+05	2.041E+03
66	69	2.891E+05	2.801E+03	1.752E+03	2.875E+05	2.053E+03
67	68	2.891E+05	2.801E+03	1.752E+03	2.875E+05	2.053E+03
68	67	2.890E+05	2.801E+03	1.754E+03	2.875E+05	2.065E+03
69	65	2.890E+05	2.801E+03	1.753E+03	2.875E+05	2.063E+03
70	64	2.890E+05	2.801E+03	1.755E+03	2.875E+05	2.065E+03

71	62	2.890E+05	2.801E+03	1.754E+03	2.875E+05	2.065E+03
72	61	2.890E+05	2.800E+03	1.725E+03	2.875E+05	2.019E+03
73	60	2.889E+05	2.800E+03	1.735E+03	2.874E+05	2.038E+03
74	59	2.889E+05	2.800E+03	1.735E+03	2.874E+05	2.038E+03
75	57	2.889E+05	2.800E+03	1.733E+03	2.874E+05	2.038E+03
76	55	2.890E+05	2.801E+03	1.743E+03	2.874E+05	2.053E+03
77	54	2.890E+05	2.801E+03	1.743E+03	2.874E+05	2.053E+03
78	52	2.890E+05	2.801E+03	1.743E+03	2.874E+05	2.053E+03
79	51	2.890E+05	2.801E+03	1.744E+03	2.874E+05	2.062E+03
80	50	2.890E+05	2.801E+03	1.744E+03	2.874E+05	2.062E+03
81	49	2.890E+05	2.801E+03	1.744E+03	2.874E+05	2.062E+03
82	48	2.890E+05	2.801E+03	1.735E+03	2.874E+05	2.061E+03
83	46	2.889E+05	2.800E+03	1.741E+03	2.874E+05	2.061E+03
84	45	2.889E+05	2.800E+03	1.741E+03	2.874E+05	2.061E+03
85	44	2.889E+05	2.800E+03	1.740E+03	2.874E+05	2.058E+03
86	43	2.890E+05	2.801E+03	1.739E+03	2.874E+05	2.054E+03
87	41	2.889E+05	2.802E+03	1.748E+03	2.874E+05	2.079E+03
88*	40	2.889E+05	2.802E+03	1.748E+03	2.874E+05	2.079E+03
89	39	2.890E+05	2.803E+03	1.736E+03	2.873E+05	2.103E+03
90	38	2.890E+05	2.805E+03	1.715E+03	2.873E+05	2.081E+03
91	37	2.890E+05	2.807E+03	1.700E+03	2.871E+05	1.880E+03
92	36	2.890E+05	2.810E+03	1.709E+03	2.871E+05	1.913E+03
93	35	2.890E+05	2.810E+03	1.709E+03	2.871E+05	1.913E+03
94+	33	2.890E+05	2.810E+03	1.710E+03	2.871E+05	1.913E+03
95	32	2.891E+05	2.812E+03	1.734E+03	2.871E+05	1.937E+03
96++	31	2.892E+05	2.813E+03	1.790E+03	2.871E+05	1.951E+03
97	30	2.896E+05	2.818E+03	1.813E+03	2.883E+05	2.091E+03
98	29	2.896E+05	2.818E+03	1.822E+03	2.883E+05	2.103E+03
99	28	2.896E+05	2.818E+03	1.822E+03	2.883E+05	2.103E+03
100	26	2.896E+05	2.818E+03	1.830E+03	2.883E+05	2.104E+03
101--	25	2.897E+05	2.820E+03	1.801E+03	2.883E+05	2.018E+03
102	24	2.898E+05	2.820E+03	1.783E+03	2.887E+05	1.885E+03
103	23	2.900E+05	2.822E+03	1.749E+03	2.887E+05	1.921E+03
104**	22	2.901E+05	2.823E+03	1.715E+03	2.887E+05	1.935E+03
105	21	2.904E+05	2.828E+03	1.737E+03	2.887E+05	2.005E+03
106	19	2.905E+05	2.829E+03	1.750E+03	2.887E+05	1.998E+03
107	18	2.905E+05	2.829E+03	1.750E+03	2.887E+05	1.998E+03
108	17	2.905E+05	2.829E+03	1.750E+03	2.887E+05	1.998E+03
109	16	2.912E+05	2.836E+03	1.662E+03	2.900E+05	2.117E+03
110	15	2.914E+05	2.838E+03	1.655E+03	2.905E+05	2.103E+03
111	14	2.916E+05	2.838E+03	1.639E+03	2.905E+05	1.967E+03
112	13	2.916E+05	2.838E+03	1.639E+03	2.905E+05	1.967E+03
113	12	2.918E+05	2.841E+03	1.628E+03	2.914E+05	1.913E+03
114	11	2.925E+05	2.846E+03	1.681E+03	2.922E+05	2.325E+03
115	10	2.931E+05	2.846E+03	1.678E+03	2.925E+05	2.049E+03
116	9	2.947E+05	2.852E+03	1.700E+03	2.932E+05	2.062E+03

117	8	2.952E+05	2.855E+03	1.679E+03	2.932E+05	2.066E+03
118	7	2.961E+05	2.858E+03	1.599E+03	2.933E+05	2.107E+03
119	6	2.972E+05	2.861E+03	1.568E+03	2.956E+05	1.986E+03
120	5	2.999E+05	2.869E+03	1.688E+03	2.984E+05	2.189E+03
121	4	3.030E+05	2.892E+03	1.980E+03	3.020E+05	2.905E+03
122	3	3.065E+05	2.909E+03	1.652E+03	3.059E+05	2.312E+03
123	2	3.106E+05	2.954E+03	1.517E+03	3.094E+05	1.738E+03
124	1	3.208E+05	3.107E+03	1.608E+03	3.204E+05	2.111E+03

0-SE tree based on mean is marked with * and has 40 terminal nodes

0-SE tree based on median is marked with + and has 33 terminal nodes

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 ++

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 weight in the node

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	Matrix rank	Node D-mean	Node MSE	Split variable	Interacting variable
1	50000	50000	1	3.371E+03	3.208E+05	wtgain	
2	20241	20241	1	3.247E+03	3.707E+05	black	
4	16410	16410	1	3.295E+03	3.463E+05	smoke	
8	13965	13965	1	3.335E+03	3.306E+05	boy	
16	6976	6976	1	3.287E+03	3.073E+05	age	
32T	1613	1613	1	3.178E+03	3.060E+05	age	
33T	5363	5363	1	3.320E+03	3.031E+05	age	
17	6989	6989	1	3.383E+03	3.493E+05	age	
34T	1621	1621	1	3.273E+03	3.472E+05	married	
35T	5368	5368	1	3.417E+03	3.452E+05	married	
9T	2445	2445	1	3.064E+03	3.739E+05	boy	
5	3831	3831	1	3.041E+03	4.230E+05	smoke	
10T	3406	3406	1	3.069E+03	4.169E+05	boy	
11T	425	425	1	2.818E+03	4.159E+05	-	
3	29759	29759	1	3.455E+03	2.693E+05	married	
6	8291	8291	1	3.332E+03	2.715E+05	wtgain	
12	5399	5399	1	3.280E+03	2.658E+05	boy	
24	2616	2616	1	3.220E+03	2.497E+05	black	
48	1707	1707	1	3.268E+03	2.363E+05	smoke	
96T	1239	1239	1	3.328E+03	2.221E+05	visit	
97T	468	468	1	3.110E+03	2.399E+05	-	
49T	909	909	1	3.131E+03	2.628E+05	ed	

25T	2783	2783	1	3.336E+03	2.746E+05	black
13	2892	2892	1	3.429E+03	2.676E+05	black
26T	1977	1977	1	3.477E+03	2.499E+05	boy
27T	915	915	1	3.328E+03	2.911E+05	boy
7	21468	21468	1	3.503E+03	2.604E+05	boy
14	10148	10148	1	3.437E+03	2.425E+05	smoke
28	9290	9290	1	3.457E+03	2.379E+05	wtgain
56	4812	4812	1	3.406E+03	2.300E+05	black
112T	4460	4460	1	3.420E+03	2.168E+05	ed
113T	352	352	1	3.223E+03	3.617E+05	-
57	4478	4478	1	3.512E+03	2.406E+05	black
114T	4119	4119	1	3.528E+03	2.322E+05	ed
115T	359	359	1	3.320E+03	2.980E+05	-
29T	858	858	1	3.224E+03	2.427E+05	wtgain
15	11320	11320	1	3.561E+03	2.692E+05	smoke
30	10337	10337	1	3.580E+03	2.658E+05	wtgain
60T	6083	6083	1	3.530E+03	2.581E+05	age
61	4254	4254	1	3.652E+03	2.680E+05	black
122T	3918	3918	1	3.669E+03	2.597E+05	age
123T	336	336	1	3.451E+03	3.218E+05	-
31T	983	983	1	3.366E+03	2.640E+05	wtgain

Number of terminal nodes of final tree: 22

Total number of nodes of final tree: 43

Second best split variable (based on curvature test) at root node is black

Regression tree:

At splits on categorical variables, values not in training data go to the right

Node 1: wtgain <= 27.500000

Node 2: black = "0"

Node 4: smoke = "0"

Node 8: boy = "0"

Node 16: age <= 23.500000

Node 32: weight-mean = 3178.3850

Node 16: age > 23.500000 or NA

Node 33: weight-mean = 3319.5172

Node 8: boy /= "0"

Node 17: age <= 23.500000

Node 34: weight-mean = 3272.5725

Node 17: age > 23.500000 or NA

Node 35: weight-mean = 3416.6200

Node 4: smoke /= "0"

Node 9: weight-mean = 3064.3845

Node 2: black /= "0"

Node 5: smoke = "0"

```
Node 10: weight-mean = 3069.3679
Node 5: smoke /= "0"
Node 11: weight-mean = 2817.5129
Node 1: wtgain > 27.500000 or NA
Node 3: married = "0"
Node 6: wtgain <= 40.500000
Node 12: boy = "0"
Node 24: black = "0"
Node 48: smoke = "0"
Node 96: weight-mean = 3327.5650
Node 48: smoke /= "0"
Node 97: weight-mean = 3110.0064
Node 24: black /= "0"
Node 49: weight-mean = 3130.7426
Node 12: boy /= "0"
Node 25: weight-mean = 3336.2627
Node 6: wtgain > 40.500000 or NA
Node 13: black = "0"
Node 26: weight-mean = 3476.6783
Node 13: black /= "0"
Node 27: weight-mean = 3327.5301
Node 3: married /= "0"
Node 7: boy = "0"
Node 14: smoke = "0"
Node 28: wtgain <= 35.500000
Node 56: black = "0"
Node 112: weight-mean = 3420.1078
Node 56: black /= "0"
Node 113: weight-mean = 3222.5142
Node 28: wtgain > 35.500000 or NA
Node 57: black = "0"
Node 114: weight-mean = 3528.3700
Node 57: black /= "0"
Node 115: weight-mean = 3319.6546
Node 14: smoke /= "0"
Node 29: weight-mean = 3223.7063
Node 7: boy /= "0"
Node 15: smoke = "0"
Node 30: wtgain <= 38.500000
Node 60: weight-mean = 3529.5090
Node 30: wtgain > 38.500000 or NA
Node 61: black = "0"
Node 122: weight-mean = 3668.7920
Node 61: black /= "0"
Node 123: weight-mean = 3450.9702
Node 15: smoke /= "0"
```

Node 31: weight-mean = 3366.3713

WARNING: p-values below are not adjusted for split search. For a bootstrap solution, see Loh et al. (2016), "Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

Node 1: Intermediate node

A case goes into Node 2 if wtgain <= 27.500000

wtgain mean = 30.709220

Coefficients of least squares regression function:

Regressor	Coefficient	t-stat	p-value
Constant	3370.8	1330.8	0.0000

Mean of weight = 3370.76

Node 2: Intermediate node

A case goes into Node 4 if black = "0"

black mode = "0"

Node 4: Intermediate node

A case goes into Node 8 if smoke = "0"

smoke mode = "0"

Node 8: Intermediate node

A case goes into Node 16 if boy = "0"

boy mode = "1"

Node 16: Intermediate node

A case goes into Node 32 if age <= 23.500000

age mean = 28.196674

Node 32: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3178.4	230.76	0.0000

Mean of weight = 3178.38

Node 33: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3319.5	441.53	0.0000

Mean of weight = 3319.52

Node 17: Intermediate node

A case goes into Node 34 if age <= 23.500000
age mean = 28.249964

Node 34: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3272.6	223.60	0.0000

Mean of weight = 3272.57

Node 35: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3416.6	426.08	0.0000

Mean of weight = 3416.62

Node 9: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3064.4	247.79	0.0000

Mean of weight = 3064.38

Node 5: Intermediate node

A case goes into Node 10 if smoke = "0"
smoke mode = "0"

Node 10: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3069.4	277.42	0.0000

Mean of weight = 3069.37

Node 11: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	2817.5	90.069	0.0000

Mean of weight = 2817.51

Node 3: Intermediate node

A case goes into Node 6 if married = "0"
married mode = "1"

Node 6: Intermediate node

A case goes into Node 12 if wtgain <= 40.500000
wtgain mean = 39.896997

Node 12: Intermediate node

A case goes into Node 24 if boy = "0"

boy mode = "1"

Node 24: Intermediate node

A case goes into Node 48 if black = "0"

black mode = "0"

Node 48: Intermediate node

A case goes into Node 96 if smoke = "0"

smoke mode = "0"

Node 96: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3327.6	248.53	0.0000

Mean of weight = 3327.56

Node 97: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3110.0	137.37	0.0000

Mean of weight = 3110.01

Node 49: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3130.7	184.11	0.0000

Mean of weight = 3130.74

Node 25: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3336.3	335.85	0.11102E-15

Mean of weight = 3336.26

Node 13: Intermediate node

A case goes into Node 26 if black = "0"

black mode = "0"

Node 26: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3476.7	309.23	0.0000

Mean of weight = 3476.68

Node 27: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3327.5	186.57	0.11102E-15

Mean of weight = 3327.53

Node 7: Intermediate node

A case goes into Node 14 if boy = "0"
 boy mode = "1"

Node 14: Intermediate node

A case goes into Node 28 if smoke = "0"
 smoke mode = "0"

Node 28: Intermediate node

A case goes into Node 56 if wtgain <= 35.500000
 wtgain mean = 37.944241

Node 56: Intermediate node

A case goes into Node 112 if black = "0"
 black mode = "0"

Node 112: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3420.1	490.53	0.0000

Mean of weight = 3420.11

Node 113: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3222.5	100.53	0.0000

Mean of weight = 3222.51

Node 57: Intermediate node

A case goes into Node 114 if black = "0"
 black mode = "0"

Node 114: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value
Constant	3528.4	469.96	0.0000

Mean of weight = 3528.37

Node 115: Terminal node

Coefficients of least squares regression functions:

```

Regressor    Coefficient  t-stat      p-value
Constant     3319.7         115.22      0.11102E-15
Mean of weight = 3319.65
-----

Node 29: Terminal node
Coefficients of least squares regression functions:
Regressor    Coefficient  t-stat      p-value
Constant     3223.7         191.68      0.0000
Mean of weight = 3223.71
-----

Node 15: Intermediate node
A case goes into Node 30 if smoke = "0"
smoke mode = "0"
-----

Node 30: Intermediate node
A case goes into Node 60 if wtgain <= 38.500000
wtgain mean = 38.237206
-----

Node 60: Terminal node
Coefficients of least squares regression functions:
Regressor    Coefficient  t-stat      p-value
Constant     3529.5         541.81      0.33307E-15
Mean of weight = 3529.51
-----

Node 61: Intermediate node
A case goes into Node 122 if black = "0"
black mode = "0"
-----

Node 122: Terminal node
Coefficients of least squares regression functions:
Regressor    Coefficient  t-stat      p-value
Constant     3668.8         450.62      0.0000
Mean of weight = 3668.79
-----

Node 123: Terminal node
Coefficients of least squares regression functions:
Regressor    Coefficient  t-stat      p-value
Constant     3451.0         111.52      0.0000
Mean of weight = 3450.97
-----

Node 31: Terminal node
Coefficients of least squares regression functions:
Regressor    Coefficient  t-stat      p-value
Constant     3366.4         205.42      0.22204E-15
Mean of weight = 3366.37
-----

```

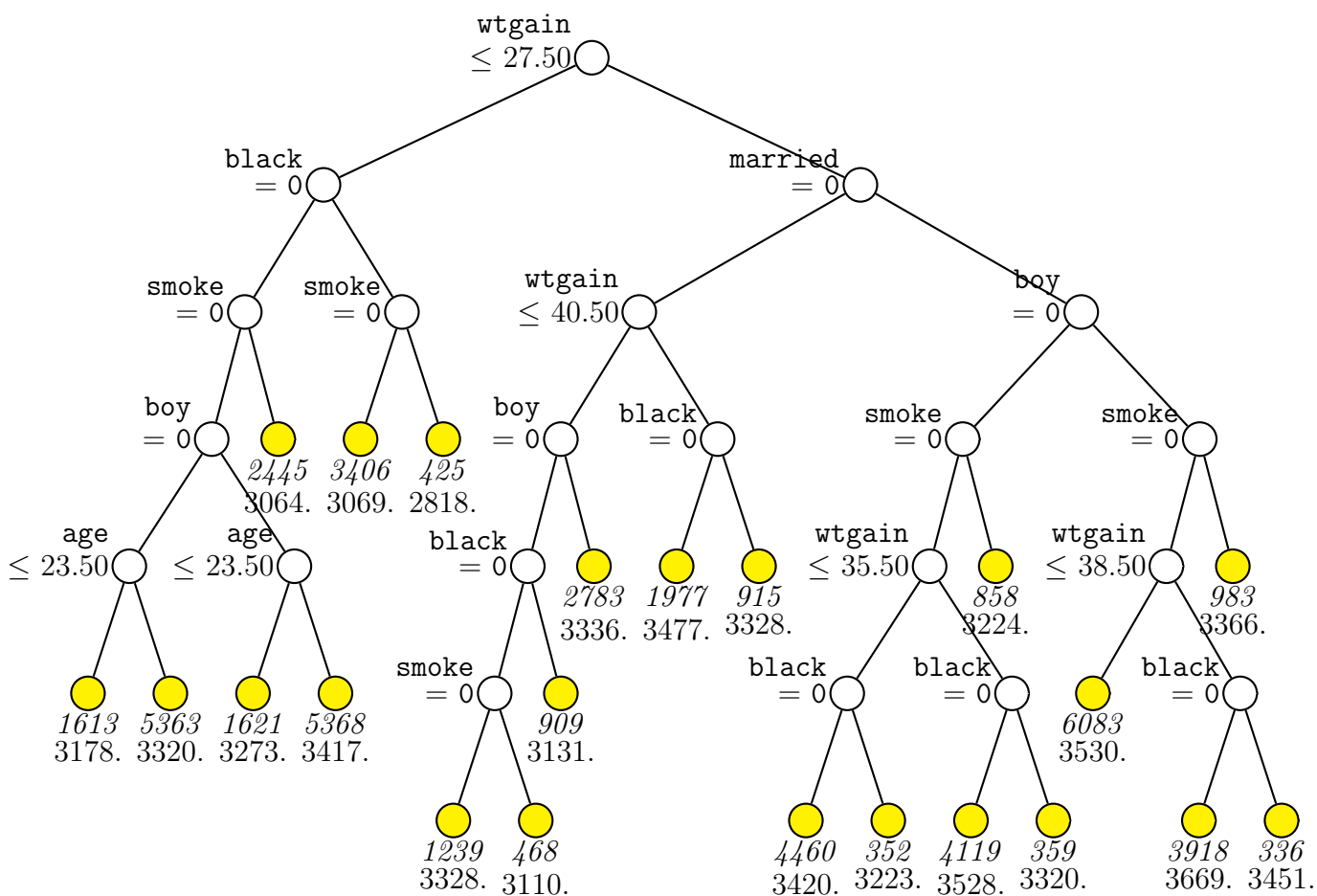
Proportion of variance (R-squared) explained by tree model: .0952

Observed and fitted values are stored in cons.fit

LaTeX code for tree is in cons.tex

Split and fit variable names are stored in cons.var

Figure 9 shows the tree diagram. It is rather large and may not be so easy to interpret. This is because the complexity of a piecewise-constant model rests completely in the tree structure. GUIDE has 4 options that will reduce the tree size by moving some of the complexity to the nodes of the tree.



5.1.3 Contents of cons.var

Following are the contents of the text file `cons.var`. Column 1 gives the node number, column 2 is a `c`, `s`, or `t`, depending on whether the split variable is `C` or `S`, or if the node is terminal. Column 3 gives the name of the split variable; if the node is terminal, the name is printed as `NONE`. Column 4 gives the name of the interacting variable if it is present; if there is no interacting variable, the split variable name is repeated. If a node is nonterminal, column 5 contains an integer indicating the number of parameter values to follow on the same line. For example, the integer is 1 for node 1 and it is followed by the value 0.2750000000E+02 which is the split point. (If a split is on a categorical variable, column 5 will give the number of categorical values defined by the split and subsequent columns will give those values.) If a node is terminal, column 5 gives the node mean of the `D` variable. The main purpose of this file is to facilitate machine extraction of the split information without parsing `cons.out`.

```

1 s wtgain wtgain      1  0.2750000000E+02
2 c black black        1  "0"
4 c smoke smoke        1  "0"
8 c boy boy            1  "0"
16 s age age           1  0.2350000000E+02
32 t age age           0.3178384997E+04
33 t age age           0.3319517248E+04
8 c boy boy            1  "0"
17 s age age           1  0.2350000000E+02
34 t married married   0.3272572486E+04
35 t married married   0.3416619970E+04
4 c smoke smoke        1  "0"
9 t boy boy            0.3064384458E+04
2 c black black        1  "0"
5 c smoke smoke        1  "0"
10 t boy boy           0.3069367880E+04
5 c smoke smoke        1  "0"
11 t NONE NONE         0.2817512941E+04
3 c married married    1  "0"
6 s wtgain wtgain      1  0.4050000000E+02
12 c boy boy           1  "0"
24 c black black       1  "0"
48 c smoke smoke       1  "0"
96 t visit visit       0.3327564972E+04
48 c smoke smoke       1  "0"
97 t NONE NONE         0.3110006410E+04
24 c black black       1  "0"
49 t ed ed             0.3130742574E+04
12 c boy boy           1  "0"
25 t black black       0.3336262666E+04
13 c black black       1  "0"

```

```

26 t boy boy      0.3476678300E+04
13 c black black      1  "0"
27 t boy boy      0.3327530055E+04
 3 c married married      1  "0"
 7 c boy boy      1  "0"
14 c smoke smoke      1  "0"
28 s wtgain wtgain      1  0.3550000000E+02
56 c black black      1  "0"
112 t ed ed      0.3420107848E+04
56 c black black      1  "0"
113 t NONE NONE      0.3222514205E+04
57 c black black      1  "0"
114 t ed ed      0.3528369993E+04
57 c black black      1  "0"
115 t NONE NONE      0.3319654596E+04
14 c smoke smoke      1  "0"
29 t wtgain wtgain      0.3223706294E+04
 7 c boy boy      1  "0"
15 c smoke smoke      1  "0"
30 s wtgain wtgain      1  0.3850000000E+02
60 t age age      0.3529508959E+04
61 c black black      1  "0"
122 t age age      0.3668791986E+04
61 c black black      1  "0"
123 t NONE NONE      0.3450970238E+04
15 c smoke smoke      1  "0"
31 t wtgain wtgain      0.3366371312E+04

```

5.2 Least squares simple linear: birthwt data

Instead of fitting a constant in each node, we can fit a simple polynomial regression model of the form $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \dots, \beta_k x^k$, where the degree of the polynomial k is pre-specified and the best predictor variable x is chosen at each node by GUIDE. We demonstrate this with $k = 1$ here.

5.2.1 Input file creation

Choose one of the following options:

0. Read the warranty disclaimer

1. Create a GUIDE input file

Input your choice: 1

Name of batch input file: lin.in

Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):


```

Input 1 for model fitting, 2 for importance or DIF scoring,
  3 for data conversion ([1:3], <cr>=1):
Name of batch output file: lin.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
  5=multiresponse or itemresponse, 6=longitudinal data (with 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:
Choose 0 for stepwise linear regression
Choose 1 for multiple regression (recommended if R variable is present,
  unless there are too many N, F or B variables when stepwise is better)
Choose 2 for simple polynomial in one N or F variable + R (if present)
Choose 3 to fit a constant + R (if present)
0: stepwise linear, 1: multiple linear, 2: simple polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <cr>=0): 2
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 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):
Choose a truncation method for predicted values:
0: none, 1: node range, 2: +10% node range, 3: global range,
4: 2-sided Winsorization Winsorization
Input 0, 1, 2, 3, or 4 ([0:4], <cr>=3):
These options allow different methods of truncating the predicted values.
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 characters);
enclose with matching quotes if it has spaces: birthwt.dsc
Reading data description file ...
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Dependent variable is weight
Reading data file ...
Number of records in data file: 50000
Length of longest entry in data file: 4
Checking for missing values ...
Total number of cases: 50000
  Column   Categorical   No. of   No. of missing
  number   variable         levels   observations

```

2	black	2	0
3	married	2	0
4	boy	2	0
6	smoke	2	0
9	visit	4	0
10	ed	4	0

Re-checking data ...

Assigning codes to categorical and missing values

Finished processing 5000 of 50000 observations

Finished processing 10000 of 50000 observations

Finished processing 15000 of 50000 observations

Finished processing 20000 of 50000 observations

Finished processing 25000 of 50000 observations

Finished processing 30000 of 50000 observations

Finished processing 35000 of 50000 observations

Finished processing 40000 of 50000 observations

Finished processing 45000 of 50000 observations

Finished processing 50000 of 50000 observations

Data checks complete

Rereading data

Total #cases	w/ miss. D	#missing ord. vals	#X-var	#N-var	#F-var	#S-var
50000	0	0	1	3	0	0
#M-var	#B-var	#C-var				
0	0	6				

No weight variable in data file

No. cases used for training: 50000

Finished reading data file

Choose how you wish to deal with missing values in training or test data:

Option 1: Fit separate models to complete and incomplete cases

Option 2: Impute missing F and N values at each node with means for regression

Option 3: Fit a piecewise constant model

Input selection: ([1:3], <cr>=2):

These options matter only if there are missing values

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 1.0000

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: 30

```

Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 2499
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): lin.tex
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 color(s) for the terminal nodes:
(1) yellow-blue-green
(2) red-green-blue
(3) magenta-yellow-green
(4) yellow
(5) green
(6) magenta
(7) cyan
(8) lightgray
(9) white
Input your choice ([1:9], <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): 3
These options allow saving of split info in a file
Input file nameL lin.var
Input 2 to save regressor names in a file, 1 otherwise ([1:2], <cr>=1): 2
Input file name: lin.reg
Saves names of regressors and their coefficients in a file
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: lin.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < lin.in

```

5.2.2 Results

Warning: The p-values produced by GUIDE are not adjusted for split selection. Therefore they are typically biased low. One way to adjust the p-values to control for split selection is with the bootstrap method in [Loh et al. \(2016\)](#). Our experience indicates, however, that any unadjusted p-value less than 0.01 is likely to be significant at level 0.05 after the bootstrap adjustment.

```

Least squares regression tree
Predictions truncated at global min. and max. of D sample values
This is the default truncation option

```

Pruning by cross-validation
 Data description file: birthwt.dsc
 Training sample file: birthwt.dat
 Missing value code: NA
 Records in data file start on line 1
 Dependent variable is weight
 Piecewise simple linear or constant model
 Powers are dropped if they are not significant at level .0500
The default option sets non-significant regression coefs to 0
 Number of records in data file: 50000
 Length of longest entry in data file: 4

Summary information (without x variables)

d=dependent, b=split and fit cat variable using indicator variables,
 c=split-only categorical, i=fit-only categorical (via indicators),
 s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
 m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	weight	d	240.0	6350.		
2	black	c			2	
3	married	c			2	
4	boy	c			2	
5	age	n	18.00	45.00		
6	smoke	c			2	
7	cigsper	n	0.000	60.00		
8	wtgain	n	0.000	98.00		
9	visit	c			4	
10	ed	c			4	

C variables are not used as predictors in the node models

Total	#cases w/ #cases	#missing miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var
50000	0	0	1	3	0	0	
#P-var	#M-var	#B-var	#C-var	#I-var			
0	0	0	6	0			

No weight variable in data file

No. cases used for training: 50000

Missing values imputed with node means for regression

Nodewise 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: 1.0000

Max. number of split levels: 30

Min. node sample size: 2499

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*	13	2.910E+05	2.797E+03	1.723E+03	2.904E+05	2.202E+03
2	12	2.910E+05	2.798E+03	1.722E+03	2.904E+05	2.207E+03
3+	11	2.910E+05	2.797E+03	1.727E+03	2.904E+05	2.193E+03
4	10	2.910E+05	2.798E+03	1.771E+03	2.904E+05	2.304E+03
5	9	2.910E+05	2.798E+03	1.771E+03	2.904E+05	2.304E+03
6	8	2.910E+05	2.798E+03	1.771E+03	2.904E+05	2.304E+03
7--	7	2.910E+05	2.798E+03	1.771E+03	2.904E+05	2.304E+03
8	6	2.918E+05	2.803E+03	1.820E+03	2.911E+05	2.268E+03
9	5	2.918E+05	2.803E+03	1.820E+03	2.911E+05	2.268E+03
10**	4	2.918E+05	2.803E+03	1.820E+03	2.911E+05	2.268E+03
11	3	2.944E+05	2.805E+03	1.897E+03	2.934E+05	2.381E+03
12	2	2.994E+05	2.833E+03	1.756E+03	2.989E+05	1.995E+03
13	1	3.069E+05	2.887E+03	1.553E+03	3.059E+05	1.759E+03

0-SE tree based on mean is marked with * and has 13 terminal nodes

0-SE tree based on median is marked with + and has 11 terminal nodes

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.

D-mean is mean of weight 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	Other variables
1	50000	50000	2	3.371E+03	3.069E+05	0.0432	married	+wtgain
2T	14369	14369	2	3.234E+03	3.166E+05	0.0558	black	+wtgain
3	35631	35631	2	3.426E+03	2.925E+05	0.0394	smoke	+wtgain
6	32318	32318	2	3.449E+03	2.856E+05	0.0366	boy	+wtgain
12T	15610	15610	2	3.388E+03	2.681E+05	0.0336	age	+wtgain
13T	16708	16708	2	3.506E+03	2.958E+05	0.0380	age	+wtgain
7T	3313	3313	2	3.198E+03	3.063E+05	0.0587	-	+wtgain

Number of terminal nodes of final tree: 4

Total number of nodes of final tree: 7

Second best split variable (based on curvature test) at root node is black

Regression tree:

At splits on categorical variables, values not in training data go to the right

Node 1: married = "0"

Node 2: weight-mean = 3234.4287

Node 1: married /= "0"

Node 3: smoke = "0"

Node 6: boy = "0"

Node 12: weight-mean = 3387.7463

Node 6: boy /= "0"

Node 13: weight-mean = 3506.3593

Node 3: smoke /= "0"

Node 7: weight-mean = 3198.1147

WARNING: p-values below are not adjusted for split search. For a bootstrap solution, see Loh et al. (2016), "Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

Node 1: Intermediate node

A case goes into Node 2 if married = "0"

married mode = "1"

Coefficients of least squares regression function:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	3090.0	482.24	0.0000			
wtgain	9.1433	47.517	0.22204E-14	0.0000	30.709	98.000

Mean of weight = 3370.76

Predicted values truncated at 240.000 & 6350.00

Node 2: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	2934.6	259.41	0.0000			
wtgain	9.7545	29.130	0.0000	0.0000	30.737	98.000

Mean of weight = 3234.43

Predicted values truncated at 240.000 & 6350.00

Node 3: Intermediate node

A case goes into Node 6 if smoke = "0"

smoke mode = "0"

Node 6: Intermediate node

A case goes into Node 12 if boy = "0"

```

boy mode = "1"
-----
Node 12: Terminal node
Coefficients of least squares regression functions:
Regressor   Coefficient  t-stat    p-value    Minimum      Mean      Maximum
Constant    3147.6         283.21    0.11102E-15  0.0000      30.429    98.000
wtgain       7.8926         23.289    0.11102E-14  0.0000      30.429    98.000
Mean of weight = 3387.75
Predicted values truncated at 240.000 & 6350.00
-----
Node 13: Terminal node
Coefficients of least squares regression functions:
Regressor   Coefficient  t-stat    p-value    Minimum      Mean      Maximum
Constant    3229.6         279.14    0.0000      0.0000      31.149    98.000
wtgain       8.8854         25.680    0.31086E-14  0.0000      31.149    98.000
Mean of weight = 3506.36
Predicted values truncated at 240.000 & 6350.00
-----
Node 7: Terminal node
Coefficients of least squares regression functions:
Regressor   Coefficient  t-stat    p-value    Minimum      Mean      Maximum
Constant    2906.0         129.19    0.0000      0.0000      29.688    98.000
wtgain       9.8392         14.364    0.0000      0.0000      29.688    98.000
Mean of weight = 3198.11
Predicted values truncated at 240.000 & 6350.00
-----

Proportion of variance (R-squared) explained by tree model: .0842

Observed and fitted values are stored in lin.fit
Regressor names and coefficients are stored in lin.reg
LaTeX code for tree is in lin.tex
Split and fit variable names are stored in lin.var

```

The tree model is shown in Figure 10. Besides being much smaller than the piecewise-constant model, it shows that `wtgain` (mother's weight gain) is the best linear predictor in every node.

5.2.3 Contents of `lin.var`

The contents of `lin.var` follow. Their interpretation is the same as for the piecewise constant model above.

```
1 c married married      1  "0"
```

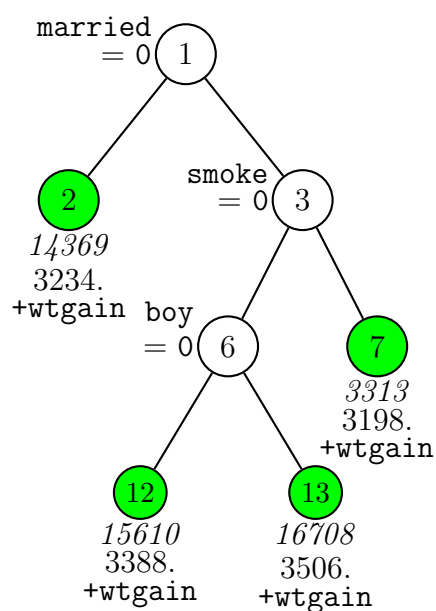


Figure 10: GUIDE v.29.6 0.50-SE pruned piecewise simple linear least-squares regression tree for predicting `weight`. Maximum number of split levels is 30 and minimum node sample size is 2499. At each split, an observation goes to the left branch if and only if the condition is satisfied. For splits on categorical variables, values not present in the training sample go to the right. Sample size (*in italics*), mean of `weight`, and sign and name of regressor variables printed below nodes. Second best split variable at root node is `black`.


```

2 t black black      0.3234428701E+04
1 c married married      1  "0"
3 c smoke smoke      1  "0"
6 c boy boy          1  "0"
12 t age age         0.3387746316E+04
6 c boy boy          1  "0"
13 t age age         0.3506359349E+04
3 c smoke smoke      1  "0"
7 t NONE NONE        0.3198114700E+04

```

5.2.4 Contents of lin.reg

The first row of the file contains the column names. Column 1 gives the node number and column 2 the linear predictor variable in the node. Columns 3 and 4 give the regression coefficients (intercept followed by slope) and columns 5 and 6 the lower and upper truncation points for the predicted D values. This file is useful for machine extraction of the regression information in each node.

```

node variable 0 1 lower upper
2 wtgain 2935. 9.754 240.0 6350.
12 wtgain 3148. 7.893 240.0 6350.
13 wtgain 3230. 8.885 240.0 6350.
7 wtgain 2906. 9.839 240.0 6350.

```

5.3 Multiple linear: birthwt data

The tree structure complexity may be reduced further by fitting a multiple linear regression in each node as follows.

5.3.1 Input file creation

We again use non-defaults to allow more options.

```

0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: mul.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: mul.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)

```

```

Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
  5=multiresponse or itemresponse, 6=longitudinal data (with 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:
Choose 0 for stepwise linear regression
Choose 1 for multiple regression (recommended if R variable is present,
  unless there are too many N, F or B variables when stepwise is better)
Choose 2 for simple polynomial in one N or F variable + R (if present)
Choose 3 to fit a constant + R (if present)
0: stepwise linear, 1: multiple linear, 2: simple polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <cr>=0): 1
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
  Option 2 opens more choices
Input 2 for no intercept term, 1 otherwise ([1:2], <cr>=1):
Choose a truncation method for predicted values:
0: none, 1: node range, 2: +10% node range, 3: global range
Input 0, 1, 2, or 3 ([0:3], <cr>=3):
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 characters);
enclose with matching quotes if it has spaces: birthwt.dsc
Reading data description file ...
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Dependent variable is weight
Reading data file ...
Number of records in data file: 50000
Length of longest entry in data file: 4
Checking for missing values ...
Total number of cases: 50000

```

Column number	Categorical variable	No. of levels	No. of missing observations
2	black	2	0
3	married	2	0
4	boy	2	0
6	smoke	2	0
9	visit	4	0
10	ed	4	0

```

Re-checking data ...
Assigning codes to categorical and missing values

```

```

Finished processing 5000 of 50000 observations
Finished processing 10000 of 50000 observations
Finished processing 15000 of 50000 observations
Finished processing 20000 of 50000 observations
Finished processing 25000 of 50000 observations
Finished processing 30000 of 50000 observations
Finished processing 35000 of 50000 observations
Finished processing 40000 of 50000 observations
Finished processing 45000 of 50000 observations
Finished processing 50000 of 50000 observations
Data checks complete
Rereading data
      Total  #cases w/  #missing
#cases  miss. D  ord. vals  #X-var  #N-var  #F-var  #S-var
 50000         0        0        1        3        0        0
#M-var  #B-var  #C-var
    0         0         6
No weight variable in data file
No. cases used for training: 50000
Finished reading data file
Choose how you wish to deal with missing values in training or test data:
Option 1: Fit separate models to complete and incomplete cases
Option 2: Impute missing F and N values at each node with means for regression
Option 3: Fit a piecewise constant model
Input selection: ([1:3], <cr>=2):
  These options are relevant only if there are missing values
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: 30
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 2499
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): mul.tex
A file by that name exists
Input 1 to overwrite it, 2 to choose another name ([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 terminal 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): 3
Input file name: mul.var
A file by that name exists
Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):
Input 2 to save truncation limits and regression coefficients in a file, 1 otherwise ([1:2], <cr>=1):
Input file name: mul.reg
A file by that name exists
Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: mul.fit
A file by that name exists
Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < mul.in

```

5.3.2 Results

```

Least squares regression tree
Predictions truncated at global min. and max. of D sample values
Truncation of predicted values can be changed by selecting a non-default option
Pruning by cross-validation
Data description file: birthwt.dsc
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Dependent variable is weight
Piecewise linear model
Number of records in data file: 50000
Length of longest entry in data file: 4

```

Summary information (without x variables)

d=dependent, b=split and fit cat variable using indicator variables,
 c=split-only categorical, i=fit-only categorical (via indicators),
 s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
 m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	weight	d	240.0	6350.		
2	black	c			2	
3	married	c			2	
4	boy	c			2	
5	age	n	18.00	45.00		
6	smoke	c			2	
7	cigsper	n	0.000	60.00		
8	wtgain	n	0.000	98.00		
9	visit	c			4	
10	ed	c			4	

Total	#cases w/	#missing					
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var	
50000	0	0	1	3	0	0	
#P-var	#M-var	#B-var	#C-var	#I-var			
0	0	0	6	0			

No weight variable in data file

No. cases used for training: 50000

Missing values imputed with node means for regression

Nodewise 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: 30

Min. node sample size: 2499

100 bootstrap calibration replicates

Scaling for N variables after bootstrap calibration: 1.000

Number of SE's for pruned tree: 0.5000

Size and CV MSE and SE of subtrees:

Tree	#Tnodes	Mean MSE	SE(Mean)	BSE(Mean)	Median MSE	BSE(Median)
1	13	2.879E+05	2.772E+03	1.866E+03	2.878E+05	1.972E+03
2	12	2.879E+05	2.772E+03	1.866E+03	2.878E+05	1.981E+03
3	11	2.879E+05	2.772E+03	1.865E+03	2.878E+05	1.980E+03
4	10	2.879E+05	2.772E+03	1.866E+03	2.879E+05	1.989E+03
5	9	2.879E+05	2.772E+03	1.862E+03	2.879E+05	1.984E+03

6	8	2.879E+05	2.772E+03	1.834E+03	2.877E+05	2.016E+03
7*	7	2.879E+05	2.770E+03	1.853E+03	2.877E+05	1.978E+03
8+	6	2.879E+05	2.771E+03	1.853E+03	2.876E+05	1.992E+03
9--	5	2.882E+05	2.775E+03	1.832E+03	2.880E+05	1.811E+03
10**	4	2.889E+05	2.781E+03	1.842E+03	2.884E+05	1.960E+03
11	3	2.893E+05	2.782E+03	1.807E+03	2.889E+05	1.938E+03
12	2	2.919E+05	2.781E+03	1.888E+03	2.905E+05	2.256E+03
13	1	2.989E+05	2.859E+03	1.653E+03	2.972E+05	2.119E+03

0-SE tree based on mean is marked with * and has 7 terminal nodes

0-SE tree based on median is marked with + and has 6 terminal nodes

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.

D-mean is mean of weight 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	Other variables
1	50000	50000	4	3.371E+03	2.989E+05	0.0683	black	
2	41858	41858	4	3.411E+03	2.797E+05	0.0673	boy	
4T	20229	20229	4	3.352E+03	2.598E+05	0.0679	married	
5	21629	21629	4	3.467E+03	2.923E+05	0.0671	married	
10T	4610	4610	4	3.354E+03	2.911E+05	0.0647	-	
11T	17019	17019	4	3.497E+03	2.911E+05	0.0600	age	
3T	8142	8142	4	3.163E+03	3.541E+05	0.0600	boy	

Number of terminal nodes of final tree: 4

Total number of nodes of final tree: 7

Second best split variable (based on curvature test) at root node is boy

Regression tree:

At splits on categorical variables, values not in training data go to the right

Node 1: black = "0"

Node 2: boy = "0"

Node 4: weight-mean = 3351.6212

Node 2: boy /= "0"

Node 5: married = "0"

Node 10: weight-mean = 3354.4534

```

Node 5: married /= "0"
Node 11: weight-mean = 3497.4645
Node 1: black /= "0"
Node 3: weight-mean = 3162.6759

```

```

*****

```

WARNING: p-values below are not adjusted for split search. For a bootstrap solution, see Loh et al. (2016), "Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

Node 1: Intermediate node

A case goes into Node 2 if black = "0"
black mode = "0"

Coefficients of least squares regression function:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	2827.1	206.14	0.0000			
age	10.226	23.882	0.41078E-14	18.000	27.416	45.000
cigsper	-13.956	-26.509	0.12657E-13	0.0000	1.4766	60.000
wtgain	9.2434	48.567	0.15543E-14	0.0000	30.709	98.000

Mean of weight = 3370.76

Predicted values truncated at 240.000 & 6350.00

Node 2: Intermediate node

A case goes into Node 4 if boy = "0"
boy mode = "1"

Node 4: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	2901.6	140.32	0.0000			
age	8.3560	13.138	0.0000	18.000	27.715	45.000
cigsper	-16.363	-21.803	0.0000	0.0000	1.5738	60.000
wtgain	7.9611	27.979	0.0000	0.0000	30.667	98.000

Mean of weight = 3351.62

Predicted values truncated at 240.000 & 6350.00

Node 5: Intermediate node

A case goes into Node 10 if married = "0"
married mode = "1"

Node 10: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
-----------	-------------	--------	---------	---------	------	---------

```

Constant      2976.3      70.584      0.0000
age            4.5938      3.1108      0.18773E-02   18.000      24.000      45.000
cigsper       -8.8992     -7.4306      0.12623E-12   0.0000      3.3685      50.000
wtgain        9.3517      15.927      0.17764E-14   0.0000      31.855      98.000

```

Mean of weight = 3354.45

Predicted values truncated at 240.000 & 6350.00

Node 11: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	2997.8	115.13	0.0000			
age	8.0315	10.210	0.0000	18.000	28.718	45.000
cigsper	-14.397	-14.739	0.0000	0.0000	1.1626	60.000
wtgain	9.1581	26.992	0.0000	0.0000	31.203	98.000

Mean of weight = 3497.46

Predicted values truncated at 240.000 & 6350.00

Node 3: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	2757.3	82.479	0.0000			
age	5.8561	5.1506	0.26581E-06	18.000	25.886	45.000
cigsper	-20.286	-9.7933	0.11213E-13	0.0000	0.82031	40.000
wtgain	9.2828	19.903	0.16653E-14	0.0000	29.133	98.000

Mean of weight = 3162.68

Predicted values truncated at 240.000 & 6350.00

Proportion of variance (R-squared) explained by tree model: .1004

Observed and fitted values are stored in mul.fit

Regressor names and coefficients are stored in mul.reg

LaTeX code for tree is in mul.tex

Split and fit variable names are stored in mul.var

Figure 11 shows the piecewise multiple linear model. Although it is slightly smaller than the piecewise best simple linear model, it conveys less information because the multiple linear regression coefficients in the nodes are not shown.

5.3.3 Contents of mul.var

The contents of mul.var follow.

```
1 c black black      1  "0"
```

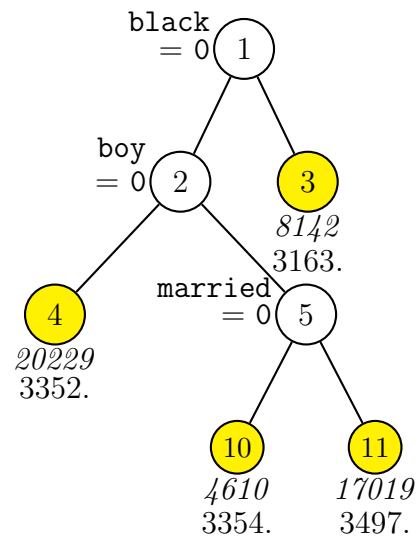



Figure 11: GUIDE v.29.6 0.50-SE pruned least-squares multiple linear regression tree for predicting **weight**. Maximum number of split levels is 30 and minimum node sample size is 2499. At each split, an observation goes to the left branch if and only if the condition is satisfied. For splits on categorical variables, values not present in the training sample go to the right. Sample size (*in italics*) and mean of **weight** printed below nodes. Second best split variable at root node is **boy**.

```

2 c boy boy      1  "0"
4 t married married 0.3351621237E+04
2 c boy boy      1  "0"
5 c married married 1  "0"
10 t NONE NONE    0.3354453362E+04
5 c married married 1  "0"
11 t age age      0.3497464540E+04
1 c black black   1  "0"
3 t boy boy      0.3162675878E+04

```

5.3.4 Contents of mul.reg

The file mul.reg give the node number and the regression coefficients in each node.

```

Node Constant age cigspers wtgain
4 2901.6 8.3560 -16.363 7.9611
10 2976.3 4.5938 -8.8992 9.3517
11 2997.8 8.0315 -14.397 9.1581
3 2757.3 5.8561 -20.286 9.2828

```

5.4 Stepwise linear: birthwt data

Yet another option is to fit a stepwise linear regression model in each node. This may be better than the piecewise multiple linear model if it reduces the number of linear predictors in some of the nodes.

5.4.1 Input file creation

```

0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: step.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: step.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
    1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
    5=multiresponse or itemresponse, 6=longitudinal data (with 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:
 Choose 0 for stepwise linear regression
 Choose 1 for multiple regression (recommended if R variable is present,
 unless there are too many N, F or B variables when stepwise is better)
 Choose 2 for simple polynomial in one N or F variable + R (if present)
 Choose 3 to fit a constant + R (if present)
 0: stepwise linear, 1: multiple linear, 2: simple polynomial, 3: constant,
 4: stepwise simple ANCOVA ([0:4], <cr>=0):
 Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
 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 a truncation method for predicted values:
 0: none, 1: node range, 2: +10% node range, 3: global range
 Input 0, 1, 2, or 3 ([0:3], <cr>=3):
 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 characters);
 enclose with matching quotes if it has spaces: birthwt.dsc
 Reading data description file ...
 Training sample file: birthwt.dat
 Missing value code: NA
 Records in data file start on line 1
 Dependent variable is weight
 Reading data file ...
 Number of records in data file: 50000
 Length of longest entry in data file: 4
 Checking for missing values ...
 Total number of cases: 50000

Column number	Categorical variable	No. of levels	No. of missing observations
2	black	2	0
3	married	2	0
4	boy	2	0
6	smoke	2	0
9	visit	4	0
10	ed	4	0

Re-checking data ...
 Assigning codes to categorical and missing values
 Finished processing 5000 of 50000 observations

```

Finished processing 10000 of 50000 observations
Finished processing 15000 of 50000 observations
Finished processing 20000 of 50000 observations
Finished processing 25000 of 50000 observations
Finished processing 30000 of 50000 observations
Finished processing 35000 of 50000 observations
Finished processing 40000 of 50000 observations
Finished processing 45000 of 50000 observations
Finished processing 50000 of 50000 observations
Data checks complete
Rereading data
      Total  #cases w/  #missing
#cases  miss. D  ord. vals  #X-var  #N-var  #F-var  #S-var
50000      0      0      1      3      0      0
#M-var  #B-var  #C-var
0      0      6
No weight variable in data file
No. cases used for training: 50000
Finished reading data file
Choose how you wish to deal with missing values in training or test data:
Option 1: Fit separate models to complete and incomplete cases
Option 2: Impute missing F and N values at each node with means for regression
Option 3: Fit a piecewise constant model
Input selection: ([1:3], <cr>=2):
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 1.0000
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: 30
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 2499
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): step.tex
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 terminal 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 regressor names in a file, 1 otherwise ([1:2], <cr>=1): 2
Input file name: step.reg
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: step.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < step.in

```

The result is the same as that for the multiple linear regression option in this example.

5.4.2 Contents of step.reg

The contents of `step.reg` are slightly different from that of `mul.reg`. Instead of giving the estimated regression coefficients in each node, it gives the names of the variables selected to fit each node. The node number is given in column 1 and the lower and upper truncation limits in columns 2 and 3.

```

node lower upper variables
4 240.0 6350. age cigsper wtgain
10 240.0 6350. age cigsper wtgain
11 240.0 6350. age cigsper wtgain
3 240.0 6350. age cigsper wtgain

```

5.5 Best ANCOVA: birthwt data

In the best simple polynomial model, categorical variables that are specified as `C` are used only to split the nodes. Sometimes, it may be desired to let them also serve as linear predictors by means of their dummy variables. This can be done in the multiple linear and stepwise linear options by simply specifying the categorical

variables as B instead of C. The same can also be done in the best simple polynomial model, but this has the undesirable effect that a single dummy variable may be chosen as the best linear predictor in a node. A better alternative is the *best simple ANCOVA* option, where at each node, (i) a single N or F variable is selected as the best linear predictor and (ii) stepwise regression is used to select a subset of the dummy variables as additional predictors. We demonstrate this by first editing the description file so that C variables are changed to B as follows. The resulting file is named `birthwtancova.dsc`.

```
birthwt.dat
NA
1
1 weight d
2 black b
3 married b
4 boy b
5 age n
6 smoke b
7 cigsper n
8 wtgain n
9 visit b
10 ed b
11 lowbwt x
```

5.5.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: ancova.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: ancova.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
    1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
    5=multiresponse or itemresponse, 6=longitudinal data (with 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:
Choose 0 for stepwise linear regression
Choose 1 for multiple regression (recommended if R variable is present,
```

unless there are too many N, F or B variables when stepwise is better)
 Choose 2 for simple polynomial in one N or F variable + R (if present)
 Choose 3 to fit a constant + R (if present)
 0: stepwise linear, 1: multiple linear, 2: simple polynomial, 3: constant,
 4: stepwise simple ANCOVA ([0:4], <cr>=0): 4
 Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
 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
 Input 0, 1, 2, 3, or 4 ([0:4], <cr>=3):
 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 characters);
 enclose with matching quotes if it has spaces: birthwtancova.dsc
 Reading data description file ...
 Training sample file: birthwt.dat
 Missing value code: NA
 Records in data file start on line 1
 Dependent variable is weight
 Reading data file ...
 Number of records in data file: 50000
 Length of longest entry in data file: 4
 Checking for missing values ...
 Total number of cases: 50000

Column number	Categorical variable	No. of levels	No. of missing observations
2	black	2	0
3	married	2	0
4	boy	2	0
6	smoke	2	0
9	visit	4	0
10	ed	4	0

Re-checking data ...
 Assigning codes to categorical and missing values
 Finished processing 5000 of 50000 observations
 Finished processing 10000 of 50000 observations
 Finished processing 15000 of 50000 observations
 Finished processing 20000 of 50000 observations
 Finished processing 25000 of 50000 observations

```

Finished processing 30000 of 50000 observations
Finished processing 35000 of 50000 observations
Finished processing 40000 of 50000 observations
Finished processing 45000 of 50000 observations
Finished processing 50000 of 50000 observations
Data checks complete
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 10
Creating dummy variables
Rereading data
      Total  #cases w/  #missing
#cases  miss. D  ord. vals  #X-var  #N-var  #F-var  #S-var
50000      0      0      1      3      0      0
#M-var  #B-var  #C-var
0        6        0
No weight variable in data file
No. cases used for training: 50000
Finished reading data file
Choose how you wish to deal with missing values in training or test data:
Option 1: Fit separate models to complete and incomplete cases
Option 2: Impute missing F and N values at each node with means for regression
Option 3: Fit a piecewise constant model
Input selection: ([1:3], <cr>=2):
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 1.0000
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: 30
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 2499
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): ancova.tex
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 color(s) for the terminal nodes:
(1) yellow-blue-green
(2) red-green-blue
(3) magenta-yellow-green

```



```

(4) yellow
(5) green
(6) magenta
(7) cyan
(8) lightgray
(9) white
Input your choice ([1:9], <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): 3
Input file name: ancova.var
Input 2 to save truncation limits and regression coefficients in a file, 1 otherwise ([1:2], <cr>=1):
Input file name: ancova.reg
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: ancova.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < ancova.in

```

5.5.2 Contents of output file

```

Least squares regression tree
Predictions truncated at global min. and max. of D sample values
Pruning by cross-validation
Data description file: birthwtancova.dsc
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Dependent variable is weight
Piecewise simple linear ANCOVA model
F-to-enter and F-to-delete: 4.000 3.990
Number of records in data file: 50000
Length of longest entry in data file: 4
Number of dummy variables created: 10

Summary information (without x variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight

```

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	weight	d	240.0	6350.		

2	black	b			2
3	married	b			2
4	boy	b			2
5	age	n	18.00	45.00	
6	smoke	b			2
7	cigsper	n	0.000	60.00	
8	wtgain	n	0.000	98.00	
9	visit	b			4
10	ed	b			4

===== Constructed variables =====

12	black.1	f	0.000	1.000
13	married.1	f	0.000	1.000
14	boy.1	f	0.000	1.000
15	smoke.1	f	0.000	1.000
16	visit.1	f	0.000	1.000
17	visit.2	f	0.000	1.000
18	visit.3	f	0.000	1.000
19	ed.1	f	0.000	1.000
20	ed.2	f	0.000	1.000
21	ed.3	f	0.000	1.000

*Indicator F variables are created for the B variables,
with the alphabetically first category of each variable set as reference level.*

Total	#cases w/	#missing					
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var	
50000	0	0	1	3	0	0	
#P-var	#M-var	#B-var	#C-var	#I-var			
0	0	6	0	0			

No weight variable in data file

No. cases used for training: 50000

Missing values imputed with node means for regression

Nodewise 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: 1.0000

Max. number of split levels: 30

Min. node sample size: 2499

Number of SE's for pruned tree: 0.5000

Size and CV MSE and SE of subtrees:

Tree	#Tnodes	Mean MSE	SE(Mean)	BSE(Mean)	Median MSE	BSE(Median)
1	14	2.869E+05	2.778E+03	1.777E+03	2.853E+05	2.154E+03
2	13	2.869E+05	2.778E+03	1.777E+03	2.853E+05	2.154E+03
3	12	2.869E+05	2.778E+03	1.789E+03	2.852E+05	2.157E+03
4	11	2.867E+05	2.774E+03	1.793E+03	2.851E+05	2.173E+03

5	10	2.867E+05	2.774E+03	1.797E+03	2.851E+05	2.158E+03
6	9	2.866E+05	2.773E+03	1.772E+03	2.851E+05	2.150E+03
7+	8	2.866E+05	2.772E+03	1.781E+03	2.850E+05	2.157E+03
8	7	2.866E+05	2.771E+03	1.823E+03	2.851E+05	1.942E+03
9	5	2.867E+05	2.769E+03	1.871E+03	2.852E+05	2.031E+03
10	4	2.866E+05	2.768E+03	1.909E+03	2.852E+05	2.032E+03
11++	3	2.865E+05	2.765E+03	1.838E+03	2.855E+05	2.045E+03
12	2	2.867E+05	2.763E+03	1.936E+03	2.861E+05	2.370E+03
13**	1	2.873E+05	2.766E+03	1.904E+03	2.868E+05	2.273E+03

0-SE tree based on mean is marked with * and has 3 terminal nodes

0-SE tree based on median is marked with + and has 8 terminal nodes

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-mean is mean of weight 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	Total	Cases	Matrix	Node	Node	Node	Split	Other
label	cases	fit	rank	D-mean	MSE	R ²	variable	variables
1T	50000	50000	9	3.371E+03	2.873E+05	0.1047	age	+wtgain

Best split at root node is on age at 20.500000000000000

Number of terminal nodes of final tree: 1

Total number of nodes of final tree: 1

Second best split variable (based on curvature test) at root node is wtgain

Regression tree:

Node 1: weight-mean = 3370.7566

Node 1: Terminal node

Coefficients of least squares regression functions:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	3035.4	336.72	0.0000			
wtgain	8.5430	45.723	0.0000	0.0000	30.709	98.000

black.1	-193.97	-27.627	0.0000	0.0000	0.16284	1.0000
boy.1	109.80	22.884	0.0000	0.0000	0.51584	1.0000
ed.1	23.350	3.6480	0.0000	0.0000	0.24258	1.0000
ed.2	47.962	7.2611	0.0000	0.0000	0.24898	1.0000
ed.3	-29.972	-4.0837	0.0000	0.0000	0.15946	1.0000
married.1	86.933	14.278	0.0000	0.0000	0.71262	1.0000
smoke.1	-205.40	-27.714	0.0000	0.0000	0.13066	1.0000

Mean of weight = 3370.76

Predicted values truncated at 240.000 & 6350.00

Proportion of variance (R-squared) explained by tree model: .1047

Observed and fitted values are stored in ancova.fit

Regressor names and coefficients are stored in ancova.reg

LaTeX code for tree is in ancova.tex

Split and fit variable names are stored in ancova.var

The results show that the tree is trivial, with no splits after pruning. The model is linear in `wtgain` and all indicator variables except those for `visit`.

5.5.3 Contents of ancova.reg

This file gives the estimated regression coefficients in each node. Variables not included in the regression have value 0.

node	selected	lower	upper	constant	age	cigsper	wtgain	black.1	boy.1
1	wtgain	2.4000E+02	6.3500E+03	3.0354E+03	0.0000E+00	0.0000E+00	8.5430E+00	-1.9397E+02	1.0980E+01

5.6 Quantile regression: birthwt data

Low birthweight is a term used to describe babies who are born weighing less than 2,500 grams (5 pounds, 8 ounces). In contrast, the average newborn weighs about 8 pounds. Over 8 percent of all newborn babies in the United States have low birthweight. We can use GUIDE to estimate conditional 0.08 quantiles ([Chaudhuri and Loh, 2002](#); [Koenker and Bassett, 1978](#)) for the `birthwt` data.

5.6.1 Piecewise constant: 1 quantile

First we fit a 0.08-quantile piecewise constant model.

5.6.2 Input file creation

```

0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: q08con.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: q08con.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
    1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
    5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended if R variable is present,
    unless there are too many N, F or B variables)
Choose 2 for best polynomial in one N or F variable + R (if present)
Choose 3 to fit a constant + R (if present)
1: multiple linear, 2: best polynomial, 3: constant ([1:3], <cr>=1): 3
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input 1 for 1 quantile, 2 for 2 quantiles ([1:2], <cr>=1):
    We fit two quantiles in the next section.
Input quantile probability ([0.00:1.00], <cr>=0.50): 0.08

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: birthwt.dsc
Reading data description file ...
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is weight
Reading data file ...
Number of records in data file: 50000
Length of longest entry in data file: 4
Checking for missing values ...
Total number of cases: 50000
Col. no. Categorical variable    #levels    #missing values
      2 black                    2              0
      3 married                  2              0
      4 boy                      2              0
      6 smoke                    2              0

```

```

      9 visit                      4          0
     10 ed                        4          0
Re-checking data ...
Assigning codes to categorical and missing values
Finished processing 5000 of 50000 observations
Finished processing 10000 of 50000 observations
Finished processing 15000 of 50000 observations
Finished processing 20000 of 50000 observations
Finished processing 25000 of 50000 observations
Finished processing 30000 of 50000 observations
Finished processing 35000 of 50000 observations
Finished processing 40000 of 50000 observations
Finished processing 45000 of 50000 observations
Finished processing 50000 of 50000 observations
Data checks complete
Rereading data
      Total  #cases w/  #missing
      #cases  miss. D  ord. vals  #X-var  #N-var  #F-var  #S-var  #B-var  #C-var
      50000      0      0      1      0      0      3      0      6
No. cases used for training: 50000
Finished reading data file
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): q08con.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: q08con.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < q08con.in

```

5.6.3 Results

```

Quantile regression tree with quantile probability 0.0800
Pruning by cross-validation
Data description file: birthwt.dsc
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is weight
Piecewise constant model
Number of records in data file: 50000
Length of longest entry in data file: 4

Summary information (without x variables)
d=dependent, b=split and fit cat variable using indicator variables,

```

c=split-only categorical, i=fit-only categorical (via indicators),
 s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
 m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	weight	d	240.0	6350.		
2	black	c			2	
3	married	c			2	
4	boy	c			2	
5	age	s	18.00	45.00		
6	smoke	c			2	
7	cigsper	s	0.000	60.00		
8	wtgain	s	0.000	98.00		
9	visit	c			4	
10	ed	c			4	

Total	#cases w/	#missing					
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var	
50000	0	0	1	0	0	3	
#P-var	#M-var	#B-var	#C-var	#I-var			
0	0	0	6	0			

No. cases used for training: 50000

Nodewise 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: 1.0000

Max. number of split levels: 30

Min. node sample size: 250

Number of SE's for pruned tree: 0.5000

Size and CV Loss and SE of subtrees:

Tree	#Tnodes	Mean Loss	SE(Mean)	BSE(Mean)	Median Loss	BSE(Median)
1	124	9.064E+01	6.772E-01	5.109E-01	9.094E+01	5.239E-01
2	123	9.064E+01	6.772E-01	5.109E-01	9.094E+01	5.239E-01
3	122	9.064E+01	6.772E-01	5.109E-01	9.094E+01	5.239E-01
4	121	9.064E+01	6.772E-01	5.109E-01	9.094E+01	5.239E-01
5	120	9.064E+01	6.772E-01	5.104E-01	9.094E+01	5.234E-01
6	119	9.064E+01	6.772E-01	5.111E-01	9.094E+01	5.243E-01
7	117	9.064E+01	6.772E-01	5.111E-01	9.093E+01	5.251E-01
8	115	9.064E+01	6.772E-01	5.113E-01	9.093E+01	5.271E-01
9	114	9.063E+01	6.772E-01	5.103E-01	9.093E+01	5.269E-01
10	112	9.064E+01	6.772E-01	5.094E-01	9.093E+01	5.283E-01
11	111	9.064E+01	6.772E-01	5.102E-01	9.093E+01	5.284E-01
12	110	9.064E+01	6.772E-01	5.102E-01	9.093E+01	5.284E-01

13	109	9.064E+01	6.772E-01	5.102E-01	9.093E+01	5.284E-01
14	108	9.064E+01	6.772E-01	5.078E-01	9.093E+01	5.230E-01
15	107	9.065E+01	6.772E-01	5.070E-01	9.093E+01	5.181E-01
16	106	9.065E+01	6.772E-01	5.078E-01	9.093E+01	5.181E-01
17	105	9.065E+01	6.772E-01	5.068E-01	9.093E+01	5.148E-01
18	103	9.065E+01	6.772E-01	5.068E-01	9.093E+01	5.148E-01
19	102	9.065E+01	6.773E-01	5.073E-01	9.093E+01	5.171E-01
20	100	9.065E+01	6.773E-01	5.072E-01	9.093E+01	5.171E-01
21	98	9.065E+01	6.773E-01	5.083E-01	9.093E+01	5.184E-01
22	97	9.065E+01	6.773E-01	5.080E-01	9.093E+01	5.168E-01
23	96	9.063E+01	6.773E-01	5.085E-01	9.092E+01	5.070E-01
24	95	9.064E+01	6.773E-01	5.074E-01	9.092E+01	4.985E-01
25	94	9.063E+01	6.772E-01	5.026E-01	9.091E+01	4.994E-01
26	93	9.061E+01	6.771E-01	5.022E-01	9.089E+01	4.942E-01
27	91	9.061E+01	6.771E-01	5.022E-01	9.088E+01	4.974E-01
28	90	9.061E+01	6.771E-01	5.022E-01	9.088E+01	4.974E-01
29	89	9.061E+01	6.771E-01	5.024E-01	9.088E+01	4.975E-01
30	88	9.061E+01	6.771E-01	5.024E-01	9.088E+01	4.975E-01
31	87	9.061E+01	6.771E-01	5.024E-01	9.088E+01	4.975E-01
32	86	9.061E+01	6.771E-01	5.032E-01	9.088E+01	5.003E-01
33	85	9.061E+01	6.771E-01	5.020E-01	9.088E+01	5.027E-01
34	84	9.061E+01	6.771E-01	5.030E-01	9.088E+01	4.997E-01
35	83	9.061E+01	6.771E-01	5.058E-01	9.090E+01	5.079E-01
36	80	9.062E+01	6.771E-01	5.081E-01	9.092E+01	5.209E-01
37	78	9.062E+01	6.770E-01	5.059E-01	9.090E+01	5.093E-01
38	77	9.061E+01	6.768E-01	5.101E-01	9.090E+01	5.141E-01
39	76	9.061E+01	6.766E-01	5.096E-01	9.088E+01	5.076E-01
40	74	9.058E+01	6.761E-01	4.982E-01	9.088E+01	5.027E-01
41	73	9.056E+01	6.761E-01	4.996E-01	9.088E+01	5.184E-01
42	72	9.057E+01	6.762E-01	4.993E-01	9.090E+01	5.120E-01
43	71	9.056E+01	6.761E-01	4.967E-01	9.086E+01	5.172E-01
44	70	9.056E+01	6.760E-01	4.974E-01	9.086E+01	5.172E-01
45	69	9.055E+01	6.760E-01	4.973E-01	9.085E+01	5.193E-01
46	68	9.056E+01	6.758E-01	4.993E-01	9.086E+01	5.312E-01
47	67	9.057E+01	6.759E-01	5.003E-01	9.089E+01	5.304E-01
48	64	9.058E+01	6.760E-01	4.954E-01	9.088E+01	5.233E-01
49	62	9.055E+01	6.758E-01	5.072E-01	9.090E+01	5.255E-01
50	61	9.055E+01	6.757E-01	5.092E-01	9.086E+01	5.323E-01
51	59	9.056E+01	6.758E-01	5.036E-01	9.086E+01	5.225E-01
52	58	9.055E+01	6.758E-01	5.055E-01	9.084E+01	5.230E-01
53	56	9.056E+01	6.762E-01	5.086E-01	9.083E+01	5.376E-01
54	55	9.054E+01	6.766E-01	5.043E-01	9.083E+01	5.504E-01
55	53	9.053E+01	6.767E-01	5.013E-01	9.083E+01	5.325E-01
56	51	9.053E+01	6.767E-01	5.013E-01	9.083E+01	5.325E-01
57	50	9.052E+01	6.770E-01	4.945E-01	9.080E+01	4.907E-01
58	49	9.055E+01	6.771E-01	5.073E-01	9.080E+01	4.951E-01

59	47	9.056E+01	6.770E-01	5.068E-01	9.080E+01	4.929E-01
60	45	9.056E+01	6.770E-01	5.104E-01	9.085E+01	5.153E-01
61	44	9.056E+01	6.770E-01	5.104E-01	9.085E+01	5.153E-01
62	43	9.056E+01	6.770E-01	5.104E-01	9.085E+01	5.153E-01
63	40	9.055E+01	6.769E-01	5.111E-01	9.085E+01	5.203E-01
64	39	9.055E+01	6.773E-01	5.069E-01	9.080E+01	5.324E-01
65	38	9.051E+01	6.773E-01	5.100E-01	9.079E+01	5.434E-01
66	37	9.052E+01	6.773E-01	5.100E-01	9.079E+01	5.347E-01
67	34	9.053E+01	6.776E-01	5.168E-01	9.079E+01	5.351E-01
68	30	9.050E+01	6.766E-01	5.185E-01	9.071E+01	5.226E-01
69+	27	9.050E+01	6.766E-01	5.155E-01	9.070E+01	5.112E-01
70	25	9.048E+01	6.770E-01	5.083E-01	9.076E+01	4.492E-01
71	23	9.045E+01	6.762E-01	5.145E-01	9.077E+01	4.365E-01
72	21	9.045E+01	6.761E-01	5.177E-01	9.079E+01	4.462E-01
73*	20	9.044E+01	6.761E-01	5.141E-01	9.079E+01	4.368E-01
74	18	9.045E+01	6.759E-01	5.143E-01	9.082E+01	4.441E-01
75	17	9.048E+01	6.763E-01	5.089E-01	9.083E+01	4.454E-01
76	16	9.046E+01	6.769E-01	5.027E-01	9.083E+01	4.363E-01
77	15	9.050E+01	6.799E-01	5.129E-01	9.081E+01	5.069E-01
78	12	9.050E+01	6.799E-01	5.129E-01	9.081E+01	5.069E-01
79	11	9.058E+01	6.810E-01	5.202E-01	9.086E+01	4.236E-01
80	10	9.062E+01	6.827E-01	5.340E-01	9.089E+01	4.472E-01
81**	9	9.060E+01	6.833E-01	5.108E-01	9.094E+01	4.929E-01
82	7	9.083E+01	6.832E-01	5.020E-01	9.118E+01	4.396E-01
83	6	9.086E+01	6.843E-01	4.902E-01	9.118E+01	4.229E-01
84	5	9.086E+01	6.843E-01	4.902E-01	9.118E+01	4.229E-01
85	4	9.190E+01	6.947E-01	5.337E-01	9.222E+01	4.955E-01
86	3	9.209E+01	6.959E-01	4.561E-01	9.222E+01	4.304E-01
87	2	9.323E+01	7.052E-01	5.111E-01	9.346E+01	5.032E-01
88	1	9.572E+01	7.520E-01	5.063E-01	9.610E+01	6.260E-01

0-SE tree based on mean is marked with * and has 20 terminal nodes

0-SE tree based on median is marked with + and has 27 terminal nodes

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-quant is quantile of weight in the node

Cases fit give the number of cases used to fit node

Node label	Total cases	Cases fit	Matrix rank	Node D-quant	Split variable	Other variables
1	50000	50000	1	2.637E+03	wtgain	
2	17934	17934	1	2.438E+03	black	
4	3536	3536	1	2.155E+03	wtgain	
8T	1107	1107	1	1.786E+03	married	
9T	2429	2429	1	2.268E+03	smoke	
5	14398	14398	1	2.523E+03	smoke	
10T	2222	2222	1	2.240E+03	visit	
11T	12176	12176	1	2.580E+03	married	
3	32066	32066	1	2.760E+03	black	
6T	4606	4606	1	2.580E+03	smoke	
7	27460	27460	1	2.807E+03	smoke	
14T	3519	3519	1	2.608E+03	wtgain	
15	23941	23941	1	2.835E+03	married	
30T	4242	4242	1	2.750E+03	visit :wtgain	
31	19699	19699	1	2.863E+03	wtgain	
62T	10795	10795	1	2.826E+03	ed	
63T	8904	8904	1	2.925E+03	boy	

Number of terminal nodes of final tree: 9

Total number of nodes of final tree: 17

Second best split variable (based on curvature test) at root node is black

Regression tree:

At splits on categorical variables, values not in training data go to the right

Node 1: wtgain <= 25.500000

Node 2: black = "1"

Node 4: wtgain <= 14.500000

Node 8: weight sample quantile = 1786.0000

Node 4: wtgain > 14.500000 or NA

Node 9: weight sample quantile = 2268.0000

Node 2: black /= "1"

Node 5: smoke = "1"

Node 10: weight sample quantile = 2240.0000

Node 5: smoke /= "1"

Node 11: weight sample quantile = 2580.0000

Node 1: wtgain > 25.500000 or NA

Node 3: black = "1"

Node 6: weight sample quantile = 2580.0000

Node 3: black /= "1"

Node 7: smoke = "1"

Node 14: weight sample quantile = 2608.0000

Node 7: smoke /= "1"

Node 15: married = "0"

```

Node 30: weight sample quantile = 2750.0000
Node 15: married /= "0"
Node 31: wtgain <= 35.500000
Node 62: weight sample quantile = 2826.0000
Node 31: wtgain > 35.500000 or NA
Node 63: weight sample quantile = 2925.0000

```

```

*****

```

WARNING: p-values below are not adjusted for split search. For a bootstrap solution, see Loh et al. (2016), "Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables", *Statistics in Medicine*, v.35, 4837-4855.

```

Node 1: Intermediate node
A case goes into Node 2 if wtgain <= 25.500000
wtgain mean = 30.709220
Predicted quantile = 2637.00
-----

```

```

Node 2: Intermediate node
A case goes into Node 4 if black = "1"
black mode = "0"
-----

```

```

Node 4: Intermediate node
A case goes into Node 8 if wtgain <= 14.500000
wtgain mean = 16.785633
-----

```

```

Node 8: Terminal node
Predicted quantile = 1786.00
-----

```

```

Node 9: Terminal node
Predicted quantile = 2268.00
-----

```

```

Node 5: Intermediate node
A case goes into Node 10 if smoke = "1"
smoke mode = "0"
-----

```

```

Node 10: Terminal node
Predicted quantile = 2240.00
-----

```

```

Node 11: Terminal node
Predicted quantile = 2580.00
-----

```

```

Node 3: Intermediate node
A case goes into Node 6 if black = "1"

```

```

black mode = "0"
-----
Node 6: Terminal node
Predicted quantile = 2580.00
-----
Node 7: Intermediate node
A case goes into Node 14 if smoke = "1"
smoke mode = "0"
-----
Node 14: Terminal node
Predicted quantile = 2608.00
-----
Node 15: Intermediate node
A case goes into Node 30 if married = "0"
married mode = "1"
-----
Node 30: Terminal node
Predicted quantile = 2750.00
-----
Node 31: Intermediate node
A case goes into Node 62 if wtgain <= 35.500000
wtgain mean = 37.130057
-----
Node 62: Terminal node
Predicted quantile = 2826.00
-----
Node 63: Terminal node
Predicted quantile = 2925.00
-----

Observed and fitted values are stored in q08con.fit
LaTeX code for tree is in q08con.tex

```

Figure 12 shows the tree model.

5.6.4 Piecewise constant: 2 quantiles

Now we fit a model to simultaneously predict two quantiles. We demonstrate this for the 0.08 and 0.12 quantiles.

5.6.5 Input file creation

0. Read the warranty disclaimer
1. Create a GUIDE input file

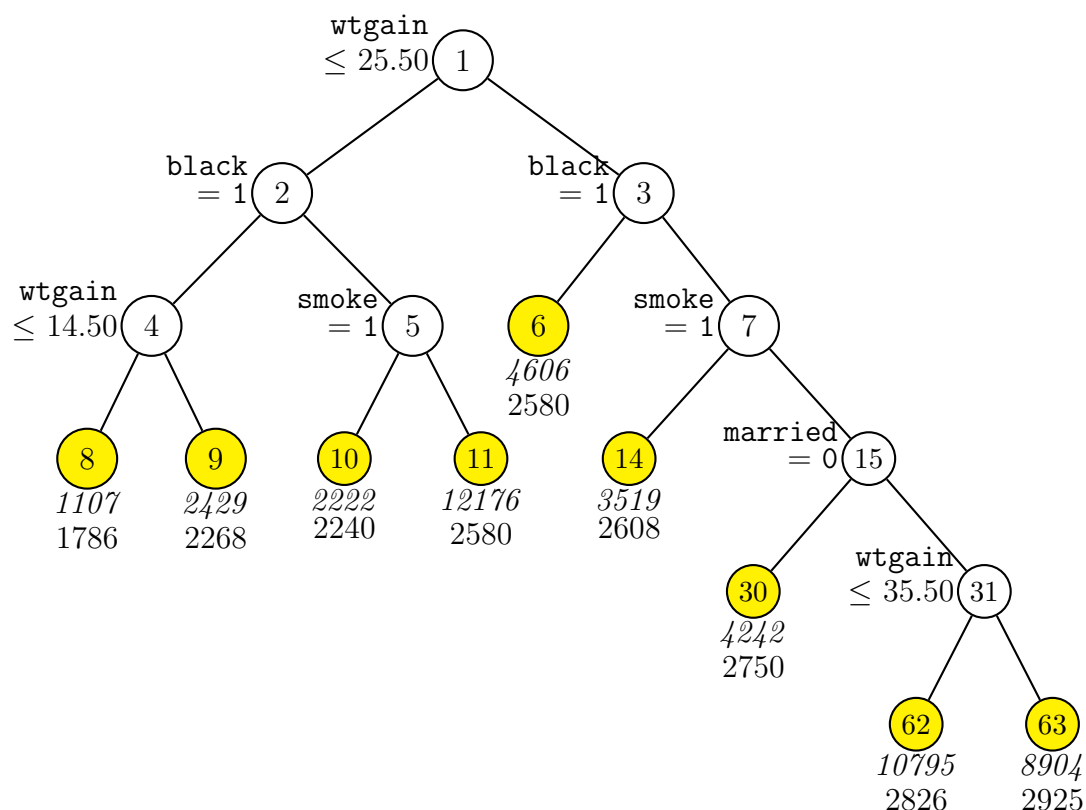


Figure 12: GUIDE v.29.6 0.50-SE pruned piecewise constant .08-quantile regression tree for predicting **weight**. Maximum number of split levels is 30 and minimum node sample size is 250. At each split, an observation goes to the left branch if and only if the condition is satisfied. For splits on categorical variables, values not present in the training sample go to the right. Sample size (*in italics*) and 0.080-quantiles of **weight** printed below nodes. Second best split variable at root node is **black**.

```

Input your choice: 1
Name of batch input file: qcon2.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: qcon2.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
    1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
    5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended if R variable is present,
    unless there are too many N, F or B variables)
Choose 2 for best polynomial in one N or F variable + R (if present)
Choose 3 to fit a constant + R (if present)
1: multiple linear, 2: best polynomial, 3: constant ([1:3], <cr>=1): 3
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input 1 for 1 quantile, 2 for 2 quantiles ([1:2], <cr>=1): 2
    Choose two quantiles here.
Input 1st quantile probability ([0.00:1.00], <cr>=0.25): 0.08
Input 2nd quantile probability ([0.00:1.00], <cr>=0.75): 0.12
    Specify the 0.08 and 0.12 quantiles here.

```

```

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: birthwt.dsc
Reading data description file ...
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is weight
Reading data file ...
Number of records in data file: 50000
Length of longest entry in data file: 4
Checking for missing values ...
Total number of cases: 50000

```

Col. no.	Categorical variable	#levels	#missing values
2	black	2	0
3	married	2	0
4	boy	2	0
6	smoke	2	0
9	visit	4	0
10	ed	4	0

```

Re-checking data ...
Assigning codes to categorical and missing values
Finished processing 5000 of 50000 observations
Finished processing 10000 of 50000 observations
Finished processing 15000 of 50000 observations
Finished processing 20000 of 50000 observations
Finished processing 25000 of 50000 observations
Finished processing 30000 of 50000 observations
Finished processing 35000 of 50000 observations
Finished processing 40000 of 50000 observations
Finished processing 45000 of 50000 observations
Finished processing 50000 of 50000 observations
Data checks complete
Rereading data
      Total #cases w/ #missing
      #cases  miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
      50000      0      0      1      0      0      3      0      6
No. cases used for training: 50000
Finished reading data file
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): qcon2.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: qcon2.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < qcon2.in

```

5.6.6 Results

```

Dual-quantile regression tree with 0.0800 and 0.1200 quantiles
Pruning by cross-validation
Data description file: birthwt.dsc
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is weight
Piecewise constant model
Number of records in data file: 50000
Length of longest entry in data file: 4

Summary information (without x variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,

```

m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	weight	d	240.0	6350.		
2	black	c			2	
3	married	c			2	
4	boy	c			2	
5	age	s	18.00	45.00		
6	smoke	c			2	
7	cigsper	s	0.000	60.00		
8	wtgain	s	0.000	98.00		
9	visit	c			4	
10	ed	c			4	

Total	#cases	w/ miss.	D	#missing	ord.	vals	#X-var	#N-var	#F-var	#S-var
50000		0		0			1	0	0	3
#P-var	#M-var	#B-var	#C-var	#I-var						
0	0	0	6	0						

No. cases used for training: 50000

Nodewise 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: 1.0000

Max. number of split levels: 30

Min. node sample size: 250

Number of SE's for pruned tree: 0.5000

Size and CV Loss and SE of subtrees:

Tree	#Tnodes	Mean Loss	SE(Mean)	BSE(Mean)	Median Loss	BSE(Median)
1	133	2.062E+02	1.435E+00	1.035E+00	2.064E+02	7.620E-01
2	132	2.062E+02	1.435E+00	1.036E+00	2.064E+02	7.661E-01
3	131	2.062E+02	1.435E+00	1.038E+00	2.064E+02	7.646E-01
4	130	2.062E+02	1.435E+00	1.038E+00	2.064E+02	7.646E-01
5	129	2.062E+02	1.435E+00	1.036E+00	2.064E+02	7.625E-01
6	128	2.062E+02	1.435E+00	1.035E+00	2.064E+02	7.589E-01
7	127	2.062E+02	1.435E+00	1.035E+00	2.064E+02	7.578E-01
8	126	2.062E+02	1.435E+00	1.035E+00	2.064E+02	7.581E-01
9	125	2.062E+02	1.435E+00	1.034E+00	2.064E+02	7.579E-01
10	124	2.062E+02	1.435E+00	1.033E+00	2.064E+02	7.568E-01
11	123	2.062E+02	1.435E+00	1.033E+00	2.064E+02	7.558E-01
12	122	2.062E+02	1.435E+00	1.033E+00	2.064E+02	7.548E-01
13	121	2.062E+02	1.435E+00	1.033E+00	2.064E+02	7.548E-01
14	120	2.062E+02	1.435E+00	1.033E+00	2.064E+02	7.545E-01

15	119	2.062E+02	1.435E+00	1.033E+00	2.064E+02	7.560E-01
16	118	2.062E+02	1.435E+00	1.030E+00	2.064E+02	7.557E-01
17	117	2.062E+02	1.435E+00	1.029E+00	2.064E+02	7.548E-01
18	116	2.062E+02	1.435E+00	1.029E+00	2.064E+02	7.499E-01
19	114	2.062E+02	1.435E+00	1.029E+00	2.064E+02	7.509E-01
20	113	2.062E+02	1.435E+00	1.029E+00	2.064E+02	7.532E-01
21	112	2.062E+02	1.435E+00	1.029E+00	2.064E+02	7.572E-01
22	110	2.062E+02	1.435E+00	1.029E+00	2.064E+02	7.572E-01
23	109	2.062E+02	1.435E+00	1.029E+00	2.064E+02	7.541E-01
24	108	2.062E+02	1.435E+00	1.024E+00	2.064E+02	7.549E-01
25	107	2.062E+02	1.435E+00	1.025E+00	2.064E+02	7.507E-01
26	106	2.062E+02	1.435E+00	1.025E+00	2.065E+02	7.491E-01
27	105	2.062E+02	1.435E+00	1.028E+00	2.065E+02	7.443E-01
28	104	2.062E+02	1.435E+00	1.029E+00	2.064E+02	7.523E-01
29	103	2.062E+02	1.434E+00	1.022E+00	2.065E+02	7.681E-01
30	102	2.062E+02	1.434E+00	1.021E+00	2.065E+02	7.693E-01
31	100	2.062E+02	1.434E+00	1.024E+00	2.065E+02	7.761E-01
32	99	2.062E+02	1.434E+00	1.026E+00	2.065E+02	7.846E-01
33	98	2.062E+02	1.434E+00	1.028E+00	2.064E+02	8.083E-01
34	97	2.061E+02	1.434E+00	1.032E+00	2.065E+02	8.204E-01
35	96	2.061E+02	1.434E+00	1.029E+00	2.064E+02	8.326E-01
36	94	2.061E+02	1.434E+00	1.031E+00	2.064E+02	8.308E-01
37	93	2.061E+02	1.434E+00	1.029E+00	2.064E+02	8.256E-01
38	90	2.061E+02	1.434E+00	1.037E+00	2.063E+02	7.912E-01
39	89	2.061E+02	1.434E+00	1.037E+00	2.063E+02	7.910E-01
40	88	2.061E+02	1.434E+00	1.039E+00	2.063E+02	7.919E-01
41	87	2.061E+02	1.434E+00	1.039E+00	2.063E+02	7.919E-01
42	86	2.061E+02	1.434E+00	1.041E+00	2.063E+02	7.981E-01
43	85	2.061E+02	1.434E+00	1.040E+00	2.063E+02	8.099E-01
44	83	2.060E+02	1.434E+00	1.048E+00	2.063E+02	8.169E-01
45	80	2.060E+02	1.434E+00	1.048E+00	2.063E+02	8.169E-01
46	77	2.060E+02	1.434E+00	1.051E+00	2.063E+02	8.121E-01
47	73	2.060E+02	1.434E+00	1.042E+00	2.062E+02	8.317E-01
48	70	2.060E+02	1.434E+00	1.037E+00	2.062E+02	8.338E-01
49	69	2.060E+02	1.434E+00	1.041E+00	2.062E+02	8.682E-01
50	64	2.059E+02	1.434E+00	1.043E+00	2.061E+02	9.118E-01
51	63	2.059E+02	1.434E+00	1.043E+00	2.061E+02	9.118E-01
52	61	2.059E+02	1.434E+00	1.043E+00	2.061E+02	9.118E-01
53	59	2.059E+02	1.433E+00	1.043E+00	2.061E+02	9.176E-01
54	58	2.059E+02	1.432E+00	1.035E+00	2.059E+02	9.166E-01
55	57	2.058E+02	1.432E+00	1.036E+00	2.059E+02	9.087E-01
56	54	2.058E+02	1.432E+00	1.022E+00	2.058E+02	8.863E-01
57	52	2.058E+02	1.432E+00	1.026E+00	2.058E+02	9.306E-01
58	51	2.057E+02	1.432E+00	1.031E+00	2.058E+02	9.181E-01
59	50	2.058E+02	1.433E+00	1.033E+00	2.058E+02	9.378E-01
60	49	2.057E+02	1.433E+00	1.046E+00	2.058E+02	9.368E-01

61	48	2.057E+02	1.432E+00	1.040E+00	2.058E+02	8.968E-01
62	45	2.057E+02	1.432E+00	1.034E+00	2.057E+02	9.038E-01
63	44	2.056E+02	1.432E+00	1.036E+00	2.057E+02	9.304E-01
64	43	2.056E+02	1.434E+00	1.026E+00	2.058E+02	9.081E-01
65	42	2.056E+02	1.435E+00	1.000E+00	2.057E+02	8.986E-01
66	39	2.056E+02	1.435E+00	9.852E-01	2.058E+02	8.817E-01
67	37	2.056E+02	1.435E+00	1.013E+00	2.058E+02	9.134E-01
68	36	2.056E+02	1.435E+00	1.013E+00	2.057E+02	8.966E-01
69	35	2.055E+02	1.436E+00	1.016E+00	2.057E+02	9.221E-01
70	34	2.055E+02	1.435E+00	1.016E+00	2.057E+02	9.202E-01
71	33	2.055E+02	1.435E+00	1.016E+00	2.057E+02	9.202E-01
72	30	2.055E+02	1.435E+00	1.024E+00	2.057E+02	9.379E-01
73	29	2.055E+02	1.436E+00	1.035E+00	2.057E+02	9.408E-01
74	27	2.055E+02	1.436E+00	1.035E+00	2.058E+02	9.398E-01
75	26	2.055E+02	1.435E+00	1.031E+00	2.057E+02	9.754E-01
76*	25	2.054E+02	1.436E+00	1.037E+00	2.056E+02	1.005E+00
77	24	2.055E+02	1.435E+00	1.043E+00	2.058E+02	9.983E-01
78	23	2.055E+02	1.437E+00	1.037E+00	2.058E+02	9.924E-01
79	21	2.055E+02	1.437E+00	1.040E+00	2.058E+02	9.895E-01
80	20	2.055E+02	1.437E+00	1.039E+00	2.059E+02	9.618E-01
81	19	2.055E+02	1.437E+00	1.035E+00	2.059E+02	9.507E-01
82++	17	2.056E+02	1.436E+00	1.006E+00	2.059E+02	8.860E-01
83--	16	2.058E+02	1.437E+00	9.774E-01	2.062E+02	7.834E-01
84	15	2.060E+02	1.439E+00	1.006E+00	2.061E+02	8.704E-01
85	14	2.060E+02	1.439E+00	1.003E+00	2.061E+02	8.772E-01
86	13	2.061E+02	1.439E+00	9.893E-01	2.065E+02	7.846E-01
87**	12	2.062E+02	1.439E+00	1.010E+00	2.065E+02	7.680E-01
88	11	2.063E+02	1.439E+00	1.003E+00	2.067E+02	8.334E-01
89	10	2.063E+02	1.439E+00	1.006E+00	2.069E+02	8.480E-01
90	8	2.064E+02	1.439E+00	1.007E+00	2.069E+02	8.577E-01
91	7	2.064E+02	1.438E+00	9.954E-01	2.069E+02	9.102E-01
92	6	2.071E+02	1.445E+00	9.577E-01	2.073E+02	8.106E-01
93	5	2.077E+02	1.452E+00	9.947E-01	2.077E+02	7.435E-01
94	4	2.088E+02	1.455E+00	1.032E+00	2.093E+02	1.269E+00
95	3	2.106E+02	1.467E+00	1.117E+00	2.107E+02	9.685E-01
96	2	2.122E+02	1.486E+00	1.002E+00	2.128E+02	9.149E-01
97	1	2.173E+02	1.576E+00	1.015E+00	2.178E+02	1.229E+00

0-SE tree based on mean is marked with * and has 25 terminal nodes

0-SE tree based on median is marked with + and has 25 terminal nodes

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

Column labeled 'Split variable' gives median if node is terminal

Node label	Total cases	Cases fit	Matrix rank	Node median	Split variable	Other variables
1	50000	50000	1	2.637E+03	wtgain	
2	17934	17934	1	2.438E+03	black	
4	14398	14398	1	2.523E+03	smoke	
8T	12176	12176	1	2.580E+03	2.750E+03	married
9T	2222	2222	1	2.240E+03	2.438E+03	visit
5	3536	3536	1	2.155E+03	smoke	
10	3134	3134	1	2.211E+03	wtgain	
20T	879	879	1	1.785E+03	2.183E+03	wtgain
21T	2255	2255	1	2.320E+03	2.495E+03	married
11T	402	402	1	1.830E+03	2.070E+03	-
3	32066	32066	1	2.760E+03	black	
6	27460	27460	1	2.807E+03	smoke	
12	23941	23941	1	2.835E+03	married	
24	19699	19699	1	2.863E+03	boy	
48T	10388	10388	1	2.920E+03	3.033E+03	wtgain
49	9311	9311	1	2.835E+03	wtgain	
98T	3981	3981	1	2.778E+03	2.889E+03	ed
99T	5330	5330	1	2.892E+03	2.990E+03	ed
25T	4242	4242	1	2.750E+03	2.863E+03	visit :wtgain
13	3519	3519	1	2.608E+03	wtgain	
26T	2035	2035	1	2.523E+03	2.649E+03	boy
27T	1484	1484	1	2.745E+03	2.863E+03	boy
7T	4606	4606	1	2.580E+03	2.693E+03	smoke

Number of terminal nodes of final tree: 12

Total number of nodes of final tree: 23

Second best split variable (based on curvature test) at root node is black

Regression tree:

At splits on categorical variables, values not in training data go to the right

Node 1: wtgain <= 25.500000

Node 2: black = "0"

Node 4: smoke = "0"

Node 8: weight sample quantiles = 2580.0000, 2750.0000

Node 4: smoke /= "0"

Node 9: weight sample quantiles = 2240.0000, 2438.0000

Node 2: black /= "0"

Node 5: smoke = "0"

```

Node 10: wtgain <= 13.500000
  Node 20: weight sample quantiles = 1785.0000, 2183.0000
Node 10: wtgain > 13.500000 or NA
  Node 21: weight sample quantiles = 2320.0000, 2495.0000
Node 5: smoke /= "0"
  Node 11: weight sample quantiles = 1830.0000, 2070.0000
Node 1: wtgain > 25.500000 or NA
  Node 3: black = "0"
    Node 6: smoke = "0"
      Node 12: married = "1"
        Node 24: boy = "1"
          Node 48: weight sample quantiles = 2920.0000, 3033.0000
Node 24: boy /= "1"
  Node 49: wtgain <= 33.500000
    Node 98: weight sample quantiles = 2778.0000, 2889.0000
Node 49: wtgain > 33.500000 or NA
  Node 99: weight sample quantiles = 2892.0000, 2990.0000
Node 12: married /= "1"
  Node 25: weight sample quantiles = 2750.0000, 2863.0000
Node 6: smoke /= "0"
  Node 13: wtgain <= 39.500000
    Node 26: weight sample quantiles = 2523.0000, 2649.0000
Node 13: wtgain > 39.500000 or NA
  Node 27: weight sample quantiles = 2745.0000, 2863.0000
Node 3: black /= "0"
  Node 7: weight sample quantiles = 2580.0000, 2693.0000

```

WARNING: p-values below are not adjusted for split search. For a bootstrap solution, see Loh et al. (2016), "Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

```

Node 1: Intermediate node
A case goes into Node 2 if wtgain <= 25.500000
wtgain mean = 30.709220
Sample lower quantile, median, and upper quantile:
  2.6370E+03    2.8000E+03    3.4020E+03
The three numbers above are the 0.08, 0.12 and 0.50 quantiles.
-----

```

```

Node 2: Intermediate node
A case goes into Node 4 if black = "0"
black mode = "0"
-----

```

```

Node 4: Intermediate node
A case goes into Node 8 if smoke = "0"
smoke mode = "0"
-----
Node 8: Terminal node
Sample lower quantile, median, and upper quantile:
      2.5800E+03      2.7500E+03      3.3740E+03
-----
Node 9: Terminal node
Sample lower quantile, median, and upper quantile:
      2.2400E+03      2.4380E+03      3.1050E+03
-----
Node 5: Intermediate node
A case goes into Node 10 if smoke = "0"
smoke mode = "0"
-----
Node 10: Intermediate node
A case goes into Node 20 if wtgain <= 13.500000
wtgain mean = 16.885769
-----
Node 20: Terminal node
Sample lower quantile, median, and upper quantile:
      1.7850E+03      2.1830E+03      3.0620E+03
-----
Node 21: Terminal node
Sample lower quantile, median, and upper quantile:
      2.3200E+03      2.4950E+03      3.1470E+03
-----
Node 11: Terminal node
Sample lower quantile, median, and upper quantile:
      1.8300E+03      2.0700E+03      2.8920E+03
-----
Node 3: Intermediate node
A case goes into Node 6 if black = "0"
black mode = "0"
-----
Node 6: Intermediate node
A case goes into Node 12 if smoke = "0"
smoke mode = "0"
-----
Node 12: Intermediate node
A case goes into Node 24 if married = "1"
married mode = "1"
-----
Node 24: Intermediate node
A case goes into Node 48 if boy = "1"

```

```

boy mode = "1"
-----
Node 48: Terminal node
Sample lower quantile, median, and upper quantile:
    2.9200E+03    3.0330E+03    3.6000E+03
-----
Node 49: Intermediate node
A case goes into Node 98 if wtgain <= 33.500000
wtgain mean = 36.951240
-----
Node 98: Terminal node
Sample lower quantile, median, and upper quantile:
    2.7780E+03    2.8890E+03    3.4020E+03
-----
Node 99: Terminal node
Sample lower quantile, median, and upper quantile:
    2.8920E+03    2.9900E+03    3.5150E+03
-----
Node 25: Terminal node
Sample lower quantile, median, and upper quantile:
    2.7500E+03    2.8630E+03    3.4285E+03
-----
Node 13: Intermediate node
A case goes into Node 26 if wtgain <= 39.500000
wtgain mean = 38.884342
-----
Node 26: Terminal node
Sample lower quantile, median, and upper quantile:
    2.5230E+03    2.6490E+03    3.2040E+03
-----
Node 27: Terminal node
Sample lower quantile, median, and upper quantile:
    2.7450E+03    2.8630E+03    3.3740E+03
-----
Node 7: Terminal node
Sample lower quantile, median, and upper quantile:
    2.5800E+03    2.6930E+03    3.2890E+03
-----

Observed and fitted values are stored in qcon2.fit
LaTeX code for tree is in qcon2.tex

```

Figure 13 shows the tree. The sample size and 0.08 and 0.12 quantiles are printed below each terminal node.

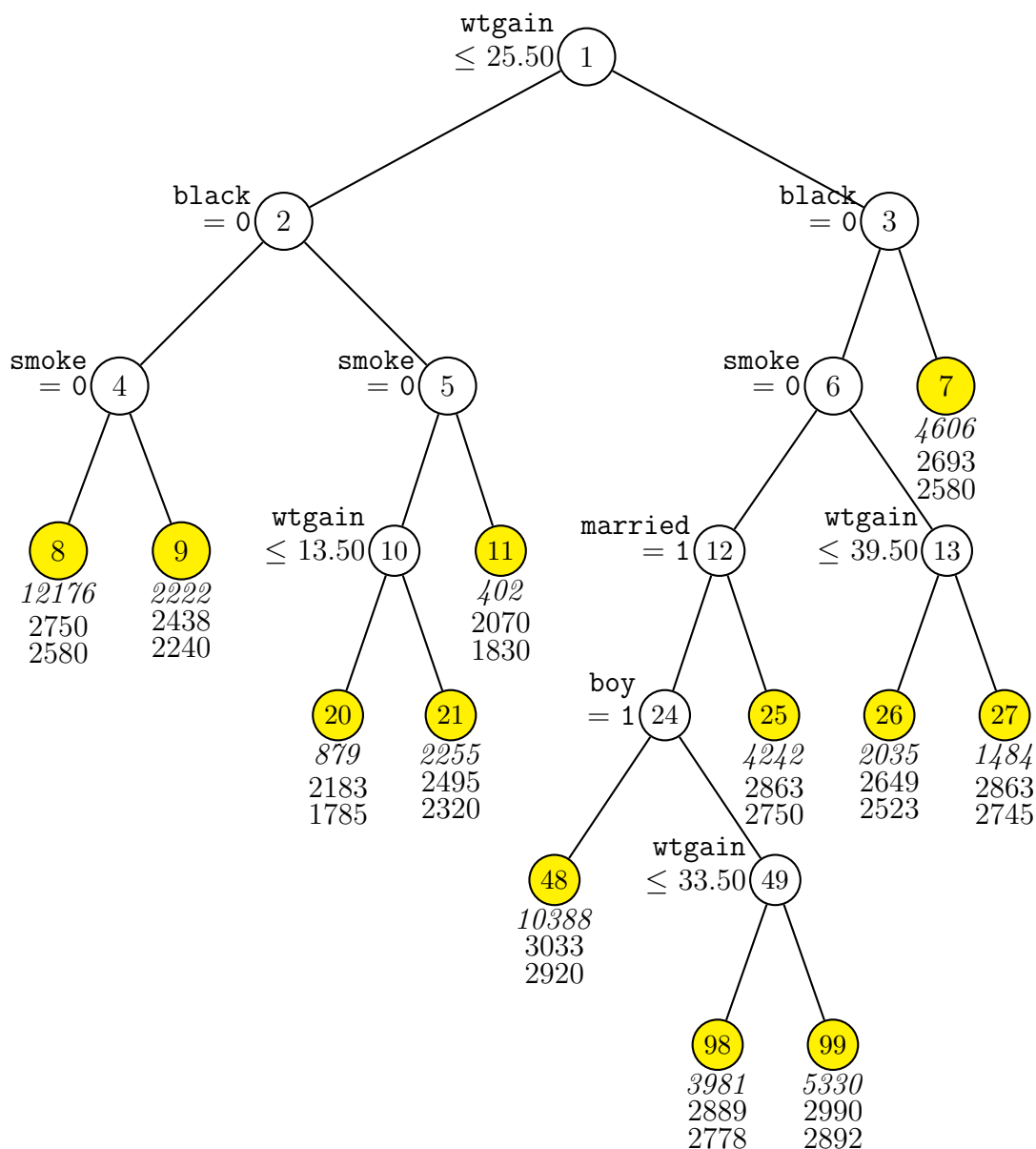


Figure 13: GUIDE v.29.6 0.50-SE pruned piecewise constant 0.080 and 0.120-quantile regression tree for predicting **weight**. Maximum number of split levels is 30 and minimum node sample size is 250. At each split, an observation goes to the left branch if and only if the condition is satisfied. For splits on categorical variables, values not present in the training sample go to the right. Sample size (*in italics*) and sample 0.120 and 0.080-quantiles of **weight** printed below nodes. Second best split variable at root node is **black**.

5.6.7 Piecewise simple linear

Next we fit a piecewise best simple linear 0.08-quantile model.

5.6.8 Input file creation

```

0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: q08lin.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: q08lin.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
    1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
    5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended if R variable is present,
    unless there are too many N, F or B variables)
Choose 2 for best polynomial in one N or F variable + R (if present)
Choose 3 to fit a constant + R (if present)
1: multiple linear, 2: best polynomial, 3: constant ([1:3], <cr>=1): 2
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input quantile probability ([0.00:1.00], <cr>=0.50): 0.08

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: birthwt.dsc
Reading data description file ...
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Dependent variable is weight
Reading data file ...
Number of records in data file: 50000
Length of longest entry in data file: 4
Checking for missing values ...
Total number of cases: 50000
Col. no. Categorical variable    #levels    #missing values
      2 black                    2              0
      3 married                  2              0
      4 boy                      2              0

```



```

        6 smoke                2                0
        9 visit                4                0
       10 ed                   4                0
Re-checking data ...
Assigning codes to categorical and missing values
Finished processing 5000 of 50000 observations
Finished processing 10000 of 50000 observations
Finished processing 15000 of 50000 observations
Finished processing 20000 of 50000 observations
Finished processing 25000 of 50000 observations
Finished processing 30000 of 50000 observations
Finished processing 35000 of 50000 observations
Finished processing 40000 of 50000 observations
Finished processing 45000 of 50000 observations
Finished processing 50000 of 50000 observations
Data checks complete
Rereading data
      Total #cases w/   #missing
      #cases  miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
      50000      0      0      1      3      0      0      0      6
No. cases used for training: 50000
Finished reading data file
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): q08lin.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: q08lin.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < q08lin.in

```

5.6.9 Results

```

Quantile regression tree with quantile probability 0.0800
No truncation of predicted values
Pruning by cross-validation
Data description file: birthwt.dsc
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Dependent variable is weight
Piecewise simple linear or constant model
Powers are dropped if they are not significant at level 1.0000
Number of records in data file: 50000
Length of longest entry in data file: 4

```

Summary information (without x variables)

d=dependent, b=split and fit cat variable using indicator variables,
 c=split-only categorical, i=fit-only categorical (via indicators),
 s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
 m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	weight	d	240.0	6350.		
2	black	c			2	
3	married	c			2	
4	boy	c			2	
5	age	n	18.00	45.00		
6	smoke	c			2	
7	cigsper	n	0.000	60.00		
8	wtgain	n	0.000	98.00		
9	visit	c			4	
10	ed	c			4	

Total	#cases w/	#missing					
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var	
50000	0	0	1	3	0	0	
#P-var	#M-var	#B-var	#C-var	#I-var			
0	0	0	6	0			

No. cases used for training: 50000

Missing values imputed with node means for regression

Nodewise 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: 1.0000

Max. number of split levels: 30

Min. node sample size: 2499

Number of SE's for pruned tree: 0.5000

Size and CV Loss and SE of subtrees:

Tree	#Tnodes	Mean Loss	SE(Mean)	BSE(Mean)	Median Loss	BSE(Median)
1*	16	9.015E+01	6.743E-01	4.721E-01	9.034E+01	4.225E-01
2	15	9.017E+01	6.742E-01	4.729E-01	9.039E+01	4.157E-01
3	14	9.019E+01	6.743E-01	4.819E-01	9.039E+01	4.163E-01
4	13	9.019E+01	6.745E-01	4.783E-01	9.039E+01	4.245E-01
5	12	9.017E+01	6.743E-01	4.918E-01	9.034E+01	4.384E-01
6	10	9.020E+01	6.746E-01	4.912E-01	9.034E+01	4.060E-01
7	9	9.020E+01	6.746E-01	4.912E-01	9.034E+01	4.060E-01
8+	8	9.020E+01	6.746E-01	4.912E-01	9.034E+01	4.060E-01
9	7	9.023E+01	6.743E-01	4.808E-01	9.035E+01	4.097E-01

10	6	9.023E+01	6.756E-01	4.861E-01	9.035E+01	4.311E-01
11	5	9.031E+01	6.804E-01	4.813E-01	9.048E+01	3.611E-01
12**	4	9.031E+01	6.811E-01	4.813E-01	9.048E+01	3.610E-01
13	3	9.070E+01	6.870E-01	4.899E-01	9.110E+01	4.495E-01
14	2	9.171E+01	6.875E-01	5.288E-01	9.195E+01	3.940E-01
15	1	9.308E+01	7.061E-01	5.258E-01	9.322E+01	4.314E-01

0-SE tree based on mean is marked with * and has 16 terminal nodes

0-SE tree based on median is marked with + and has 8 terminal nodes

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-quant is quantile of weight in the node

Cases fit give the number of cases used to fit node

Node label	Total cases	Cases fit	Matrix rank	Node D-quant	Split variable	Other variables
1	50000	50000	2	2.637E+03	black	
2T	8142	8142	2	2.381E+03	wtgain	
3	41858	41858	2	2.710E+03	smoke	
6T	5741	5741	2	2.466E+03	boy	
7	36117	36117	2	2.750E+03	married	
14T	6484	6484	2	2.637E+03	wtgain	
15T	29633	29633	2	2.778E+03	age	

Number of terminal nodes of final tree: 4

Total number of nodes of final tree: 7

Second best split variable (based on curvature test) at root node is married

Regression tree:

At splits on categorical variables, values not in training data go to the right

Node 1: black = "1"

Node 2: weight sample quantile = 2381.0000

Node 1: black /= "1"

Node 3: smoke = "1"

Node 6: weight sample quantile = 2466.0000

Node 3: smoke /= "1"

Node 7: married = "0"

```

Node 14: weight sample quantile = 2637.0000
Node 7: married /= "0"
Node 15: weight sample quantile = 2778.0000

```

```

*****

```

WARNING: p-values below are not adjusted for split search. For a bootstrap solution, see Loh et al. (2016), "Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables", *Statistics in Medicine*, v.35, 4837-4855.

Node 1: Intermediate node

A case goes into Node 2 if black = "1"

black mode = "0"

Coefficients of quantile regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	2302.7			
wtgain	11.333	0.0000	30.709	98.000

```

-----

```

Node 2: Terminal node

Coefficients of quantile regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	1977.9			
wtgain	13.899	0.0000	29.133	98.000

```

-----

```

Node 3: Intermediate node

A case goes into Node 6 if smoke = "1"

smoke mode = "0"

```

-----

```

Node 6: Terminal node

Coefficients of quantile regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	2040.7			
wtgain	14.179	0.0000	30.430	98.000

```

-----

```

Node 7: Intermediate node

A case goes into Node 14 if married = "0"

married mode = "1"

```

-----

```

Node 14: Terminal node

Coefficients of quantile regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	2372.7			
wtgain	8.9143	0.0000	31.852	98.000

```

-----

```

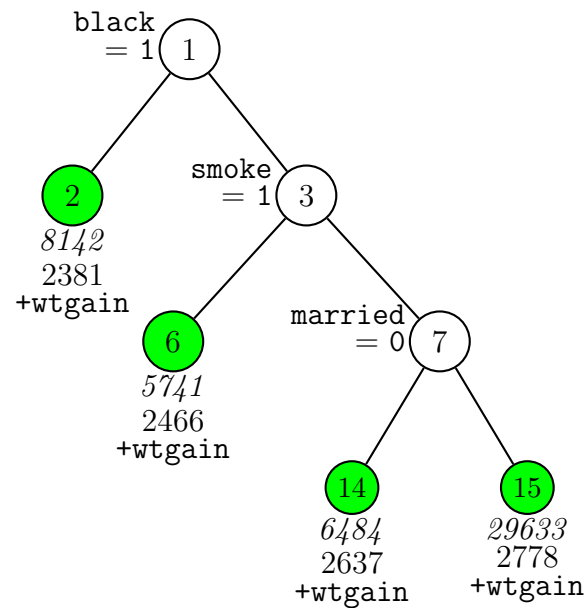


Figure 14: GUIDE v.29.6 0.50-SE pruned piecewise simple linear 0.080-quantile regression tree for predicting **weight**. Maximum number of split levels is 30 and minimum node sample size is 2499. At each split, an observation goes to the left branch if and only if the condition is satisfied. For splits on categorical variables, values not present in the training sample go to the right. Sample size (*in italics*), 0.080-quantile of **weight**, and sign and name of best regressor printed below nodes. Second best split variable at root node is **married**.

Node 15: Terminal node

Coefficients of quantile regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	2479.1			
wtgain	10.071	0.0000	30.946	98.000

Observed and fitted values are stored in q08lin.fit

LaTeX code for tree is in q08lin.tex

The tree is shown in Figure 14. Piecewise linear quantile regression with two quantiles simultaneously is not available at the present time.

5.6.10 Piecewise multiple linear

Next we fit a piecewise multiple linear 0.80-quantile model.

5.6.11 Input file creation

```

0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: q08mul.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: q08mul.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
    1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
    5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended if R variable is present,
    unless there are too many N, F or B variables)
Choose 2 for best polynomial in one N or F variable + R (if present)
Choose 3 to fit a constant + R (if present)
1: multiple linear, 2: best polynomial, 3: constant ([1:3], <cr>=1): 1
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input quantile probability ([0.00:1.00], <cr>=0.50): 0.08

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: birthwt.dsc
Reading data description file ...
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Dependent variable is weight
Reading data file ...
Number of records in data file: 50000
Length of longest entry in data file: 4
Checking for missing values ...
Total number of cases: 50000
Col. no. Categorical variable    #levels    #missing values
      2 black                    2              0
      3 married                  2              0
      4 boy                      2              0

```

```

      6 smoke                2                0
      9 visit                4                0
     10 ed                   4                0
Re-checking data ...
Assigning codes to categorical and missing values
Finished processing 5000 of 50000 observations
Finished processing 10000 of 50000 observations
Finished processing 15000 of 50000 observations
Finished processing 20000 of 50000 observations
Finished processing 25000 of 50000 observations
Finished processing 30000 of 50000 observations
Finished processing 35000 of 50000 observations
Finished processing 40000 of 50000 observations
Finished processing 45000 of 50000 observations
Finished processing 50000 of 50000 observations
Data checks complete
Rereading data
      Total #cases w/ #missing
      #cases  miss. D ord. vals #X-var #N-var #F-var #S-var #B-var #C-var
      50000      0      0      1      3      0      0      0      6
No. cases used for training: 50000
Finished reading data file
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): q08mul.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: q08mul.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < q08mul.in

```

5.6.12 Results

```

Quantile regression tree with quantile probability 0.0800
No truncation of predicted values
Pruning by cross-validation
Data description file: birthwt.dsc
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Dependent variable is weight
Piecewise linear model
Number of records in data file: 50000
Length of longest entry in data file: 4

Summary information (without x variables)

```

d=dependent, b=split and fit cat variable using indicator variables,
 c=split-only categorical, i=fit-only categorical (via indicators),
 s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
 m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	weight	d	240.0	6350.		
2	black	c			2	
3	married	c			2	
4	boy	c			2	
5	age	n	18.00	45.00		
6	smoke	c			2	
7	cigsper	n	0.000	60.00		
8	wtgain	n	0.000	98.00		
9	visit	c			4	
10	ed	c			4	

Total	#cases w/	#missing					
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var	
50000	0	0	1	3	0	0	
#P-var	#M-var	#B-var	#C-var	#I-var			
0	0	0	6	0			

No. cases used for training: 50000

Missing values imputed with node means for regression

Nodewise 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: 1.0000

Max. number of split levels: 30

Min. node sample size: 2499

100 bootstrap calibration replicates

Scaling for N variables after bootstrap calibration: 1.300

Number of SE's for pruned tree: 0.5000

Size and CV Loss and SE of subtrees:

Tree	#Tnodes	Mean Loss	SE(Mean)	BSE(Mean)	Median Loss	BSE(Median)
1	13	9.018E+01	6.757E-01	4.592E-01	9.063E+01	3.413E-01
2	10	9.018E+01	6.757E-01	4.651E-01	9.063E+01	3.393E-01
3	9	9.020E+01	6.763E-01	4.649E-01	9.066E+01	3.405E-01
4	8	9.018E+01	6.765E-01	4.786E-01	9.066E+01	3.568E-01
5	7	9.016E+01	6.762E-01	4.770E-01	9.060E+01	3.342E-01
6	6	9.019E+01	6.763E-01	4.728E-01	9.058E+01	3.210E-01
7*	5	9.015E+01	6.752E-01	4.713E-01	9.055E+01	3.366E-01
8	4	9.020E+01	6.763E-01	4.770E-01	9.060E+01	3.605E-01

9**	3	9.034E+01	6.809E-01	5.328E-01	9.060E+01	3.909E-01
10	2	9.059E+01	6.872E-01	5.121E-01	9.085E+01	3.997E-01
11	1	9.213E+01	7.065E-01	5.197E-01	9.224E+01	3.786E-01

0-SE tree based on mean is marked with * and has 5 terminal nodes
 0-SE tree based on median is marked with + and has 5 terminal nodes
 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.

D-quant is quantile of weight in the node

Cases fit give the number of cases used to fit node

Node label	Total cases	Cases Matrix fit rank	Node D-quant	Split variable	Other variables
1	50000	50000 4	2.637E+03	black	
2T	8142	8142 4	2.381E+03	wtgain	
3	41858	41858 4	2.710E+03	married	
6T	9053	9053 4	2.580E+03	boy	
7T	32805	32805 4	2.750E+03	wtgain	

Number of terminal nodes of final tree: 3

Total number of nodes of final tree: 5

Second best split variable (based on curvature test) at root node is married

Regression tree:

At splits on categorical variables, values not in training data go to the right

Node 1: black = "1"

Node 2: weight sample quantile = 2381.0000

Node 1: black /= "1"

Node 3: married = "0"

Node 6: weight sample quantile = 2580.0000

Node 3: married /= "0"

Node 7: weight sample quantile = 2750.0000

WARNING: p-values below are not adjusted for split search. For a bootstrap solution, see Loh et al. (2016), "Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables", *Statistics in Medicine*, v.35, 4837-4855.

Node 1: Intermediate node

A case goes into Node 2 if black = "1"

black mode = "0"

Coefficients of quantile regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	2176.5			
age	6.0183	18.000	27.416	45.000
cigsper	-18.493	0.0000	1.4766	60.000
wtgain	11.288	0.0000	30.709	98.000

Node 2: Terminal node

Coefficients of quantile regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	2071.3			
age	-2.0740	18.000	25.886	45.000
cigsper	-35.349	0.0000	0.82031	40.000
wtgain	13.420	0.0000	29.133	98.000

Node 3: Intermediate node

A case goes into Node 6 if married = "0"

married mode = "1"

Node 6: Terminal node

Coefficients of quantile regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	2425.4			
age	-4.0598	18.000	24.050	45.000
cigsper	-11.956	0.0000	3.3095	60.000
wtgain	9.8152	0.0000	31.661	98.000

Node 7: Terminal node

Coefficients of quantile regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	2323.5			
age	4.3902	18.000	28.725	45.000
cigsper	-20.623	0.0000	1.1337	60.000
wtgain	10.854	0.0000	30.838	98.000

Observed and fitted values are stored in q08mul.fit

LaTeX code for tree is in q08mul.tex

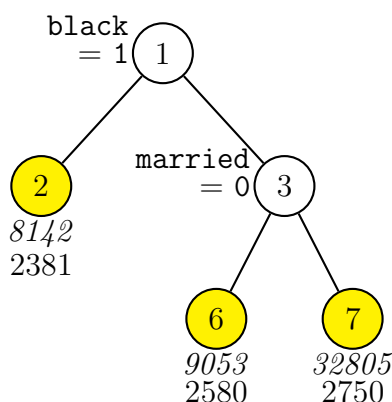


Figure 15: GUIDE v.29.6 0.50-SE pruned multiple linear 0.080-quantile regression tree for predicting **weight**. Maximum number of split levels is 30 and minimum node sample size is 2499. At each split, an observation goes to the left branch if and only if the condition is satisfied. For splits on categorical variables, values not present in the training sample go to the right. Sample size (*in italics*) and 0.080-quantiles of **weight** printed below nodes. Second best split variable at root node is **married**.

R code is stored in `q08mul.r`

Figure 15 shows the tree.

5.7 Least median of squares: birthwt data

Although median regression may be preferred to least-squares regression if there are large outliers in a data set, an alternative that is even more robust to outliers is *least median of squares* regression (Rousseeuw and Leroy, 1987). GUIDE can construct tree models using this criterion. We use the birthwt data for illustration. A session log of the input file generation is below, followed by the results and the L^AT_EX tree diagram in Figure 16.

```

0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: lms.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: lms.out
  
```

```

Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1): 2
  This is where the option for least median of squares is selected.
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended if R variable is present,
  unless there are too many N, F or B variables)
Choose 2 for simple polynomial in one N or F variable + R (if present)
Choose 3 to fit a constant + R (if present)
1: multiple linear, 2: simple polynomial, 3: constant ([1:3], <cr>=2):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: birthwt.dsc
Reading data description file ...
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Dependent variable is weight
Reading data file ...
Number of records in data file: 50000
Length of longest entry in data file: 4
Checking for missing values ...
Total number of cases: 50000
  Column  Categorical      No. of  No. of missing
  number  variable        levels  observations
      2  black             2           0
      3  married           2           0
      4  boy               2           0
      6  smoke             2           0
      9  visit             4           0
     10  ed                4           0

Re-checking data ...
Assigning codes to categorical and missing values
Finished processing 5000 of 50000 observations
Finished processing 10000 of 50000 observations
Finished processing 15000 of 50000 observations
Finished processing 20000 of 50000 observations
Finished processing 25000 of 50000 observations

```

```

Finished processing 30000 of 50000 observations
Finished processing 35000 of 50000 observations
Finished processing 40000 of 50000 observations
Finished processing 45000 of 50000 observations
Finished processing 50000 of 50000 observations
Data checks complete
Rereading data
  Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var
 50000      0      0      1      3      0      0
#M-var #B-var #C-var
      0      0      6
No weight variable in data file
No. cases used for training: 50000
Finished reading data file
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): lms.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: lms.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < lms.in

```

5.7.1 Results

```

Least median of squares regression tree
Predictions truncated at global min. and max. of D sample values
Pruning by cross-validation
Data description file: birthwt.dsc
Training sample file: birthwt.dat
Missing value code: NA
Records in data file start on line 1
Dependent variable is weight
Piecewise simple linear or constant model
Powers are dropped if they are not significant at level 1.0000
Number of records in data file: 50000
Length of longest entry in data file: 4

Summary information (without x variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                Codes/
                                Levels/

```

Column	Name		Minimum	Maximum	Periods	#Missing
1	weight	d	240.0	6350.		
2	black	c			2	
3	married	c			2	
4	boy	c			2	
5	age	n	18.00	45.00		
6	smoke	c			2	
7	cigsper	n	0.000	60.00		
8	wtgain	n	0.000	98.00		
9	visit	c			4	
10	ed	c			4	

Total #cases	w/ miss.	#missing D	ord. vals	#X-var	#N-var	#F-var	#S-var
50000	0	0	0	1	3	0	0
#P-var	#M-var	#B-var	#C-var	#I-var			
0	0	0	6	0			

No weight variable in data file

No. cases used for training: 50000

Missing values imputed with node means for regression

Nodewise 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: 1.0000

Max. number of split levels: 30

Min. node sample size: 2499

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

Size and CV median absolute residual (MAR) and SE of subtrees:

Tree	#Tnodes	Mean MAR	BSE(Mean)	Median MAR	BSE(Median)
1	13	1.591E+06	2.021E+00	3.170E+02	5.113E+00
2	11	1.591E+06	1.985E+00	3.174E+02	5.168E+00
3+	9	1.589E+06	1.951E+00	3.160E+02	4.823E+00
4	8	1.591E+06	2.031E+00	3.163E+02	5.124E+00
5--	6	1.588E+06	1.787E+00	3.164E+02	3.193E+00
6++	5	1.600E+06	2.239E+00	3.184E+02	3.990E+00
7	4	1.598E+06	2.069E+00	3.202E+02	4.519E+00
8	3	1.623E+06	2.715E+00	3.260E+02	3.087E+00
9	1	1.640E+06	9.479E-01	3.263E+02	9.920E-01

The selected tree is marked by two dashes --.

0-SE tree based on mean is marked with * and has 6 terminal nodes

0-SE tree based on median is marked with + and has 9 terminal nodes

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 bootstrap SE estimate (--).

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

D-mean is mean of weight in the node

Cases fit give the number of cases used to fit node

MAR is median of absolute residuals

Node label	Total cases	Cases Matrix fit rank	Node D-median	Node MAR	Split variable	Other variables
1	50000	50000	2	3.402E+03	3.260E+02 smoke	+cigsper
2	43467	43467	2	3.430E+03	3.241E+02 black	+wtgain
4	36117	36117	1	3.459E+03	3.120E+02 wtgain	*Constant*
8T	13965	13965	1	3.374E+03	3.150E+02 boy	*Constant*
9T	22152	22152	2	3.515E+03	3.048E+02 boy	+wtgain
5	7350	7350	2	3.231E+03	3.256E+02 boy	+wtgain
10T	3584	3584	1	3.158E+03	3.116E+02 -	*Constant*
11T	3766	3766	2	3.289E+03	3.262E+02 -	+wtgain
3	6533	6533	2	3.203E+03	3.202E+02 boy	+wtgain
6T	3148	3148	1	3.119E+03	2.976E+02 -	*Constant*
7T	3385	3385	2	3.260E+03	3.202E+02 -	+wtgain

Number of terminal nodes of final tree: 6

Total number of nodes of final tree: 11

Second best split variable (based on curvature test) at root node is cigsper

Regression tree:

At splits on categorical variables, values not in training data go to the right

Node 1: smoke = "0"

Node 2: black = "0"

Node 4: wtgain <= 27.500000

Node 8: weight-mean = 3374.0000

Node 4: wtgain > 27.500000 or NA

Node 9: weight-mean = 3515.0000

Node 2: black /= "0"

Node 5: boy = "0"

Node 10: weight-mean = 3157.5000

Node 5: boy /= "0"

Node 11: weight-mean = 3289.0000

Node 1: smoke /= "0"

Node 3: boy = "0"

Node 6: weight-mean = 3119.0000

Node 3: boy /= "0"

Node 7: weight-mean = 3260.0000

WARNING: p-values below are not adjusted for split search. For a bootstrap solution, see Loh et al. (2016), "Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

Node 1: Intermediate node

A case goes into Node 2 if smoke = "0"

smoke mode = "0"

Coefficients of least median of squares regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	3416.0			
cigsper	7.8333	0.0000	1.4766	60.000

Mean of weight = 3402.00

Predicted values truncated at 240.000 & 6350.00

Node 2: Intermediate node

A case goes into Node 4 if black = "0"

black mode = "0"

Node 4: Intermediate node

A case goes into Node 8 if wtgain <= 27.500000

wtgain mean = 31.108868

Node 8: Terminal node

Coefficients of least median of squares regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	3399.0			
age	0.0000	18.000	28.223	45.000

Mean of weight = 3374.00

Predicted values truncated at 240.000 & 6350.00

Node 9: Terminal node

Coefficients of least median of squares regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	3211.3			
wtgain	8.5000	28.000	38.380	98.000

Mean of weight = 3515.00

Predicted values truncated at 240.000 & 6350.00

Node 5: Intermediate node

A case goes into Node 10 if boy = "0"

boy mode = "1"

Node 10: Terminal node

Coefficients of least median of squares regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	3203.5			
wtgain	0.0000	0.0000	28.907	98.000

Mean of weight = 3157.50

Predicted values truncated at 240.000 & 6350.00

Node 11: Terminal node

Coefficients of least median of squares regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	3161.0			
wtgain	5.6500	0.0000	29.746	98.000

Mean of weight = 3289.00

Predicted values truncated at 240.000 & 6350.00

Node 3: Intermediate node

A case goes into Node 6 if boy = "0"

boy mode = "1"

Node 6: Terminal node

Coefficients of least median of squares regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	3104.5			
cigsper	0.0000	1.0000	11.185	60.000

Mean of weight = 3119.00

Predicted values truncated at 240.000 & 6350.00

Node 7: Terminal node

Coefficients of least median of squares regression function:

Regressor	Coefficient	Minimum	Mean	Maximum
Constant	2949.6			
wtgain	10.078	0.0000	30.341	98.000

Mean of weight = 3260.00

Predicted values truncated at 240.000 & 6350.00

Proportion of deviance explained by tree model: .0735

Observed and fitted values are stored in lms.fit

LaTeX code for tree is in lms.tex

The tree is shown in Figure 16.

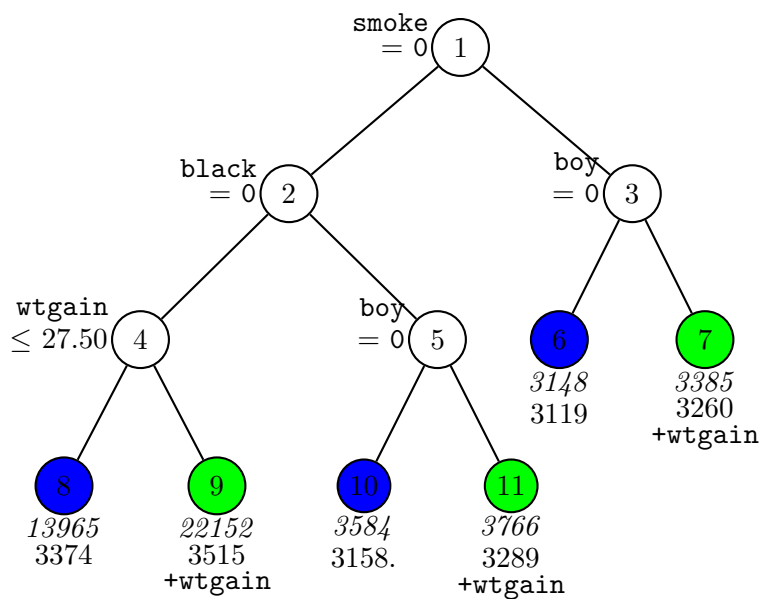


Figure 16: GUIDE v.29.6 0.50-SE pruned piecewise simple linear least-median-of-squares regression tree for predicting **weight**. Maximum number of split levels is 30 and minimum node sample size is 2499. At each split, an observation goes to the left branch if and only if the condition is satisfied. For splits on categorical variables, values not present in the training sample go to the right. Sample size (*in italics*), mean of **weight**, and sign and name of regressor variables printed below nodes. Second best split variable at root node is **cigsper**.

5.8 Poisson regression with offset: lung cancer data

We use a data set from an epidemiological study of the effect of public drinking water on cancer mortality in Missouri (Choi et al., 2005). Our data file `lungcancer.txt` gives the number of deaths (`deaths`) from lung cancer among 115 counties (`county`) during the period 1972–1981 for both sexes (`sex`) and four age groups (`agegp`): 45–54, 55–64, 65–74, and over 75. The description file `lungcancer.dsc` below lists the variables together with the county population (`pop`) and the natural log of `pop` (`logpop`). The latter is specified as `z` to serve as an offset variable and the former is excluded (`x`) from the analysis. For the purpose of illustration, we specify `sex` as `b` to allow its dummy indicator variable to serve as a linear predictor in the node Poisson models. The contents of `lungcancer.dsc` are:

```
lungcancer.txt
NA
1
1 county c
2 sex b
3 agegp c
4 deaths d
5 pop x
6 logpop z
```

Our goal is to construct a Poisson regression tree for the gender-specific rate of lung cancer deaths, where rate is the expected number of deaths in a county divided by its population size for each gender. That is, letting μ denote the expected number of gender-specific deaths in a county, we fit this model in each node of the tree:

$$\log(\mu/\text{pop}) = \beta_0 + \beta_1 I(\text{sex} = \text{M})$$

or, equivalently,

$$\log(\mu) = \beta_0 + \beta_1 I(\text{sex} = \text{M}) + \log\text{pop}.$$

5.8.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: poi.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: poi.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
```

```

(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 3
Choose Poisson regression here
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended if R variable is present,
  unless there are too many N, F or B variables)
Choose 2 for simple polynomial in one N or F variable + R (if present)
Choose 3 to fit a constant + R (if present)
1: multiple linear, 2: simple polynomial, 3: constant ([1:3], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: lungcancer.dsc
Reading data description file ...
Training sample file: lungcancer.txt
Missing value code: NA
Records in data file start on line 1
Dependent variable is deaths
Reading data file ...
Number of records in data file: 920
Length of longest entry in data file: 8
Checking for missing values ...
Total number of cases: 920

```

Column number	Categorical variable	No. of levels	No. of missing observations
1	county	115	0
2	sex	2	0
3	agegp	4	0

```

Re-checking data ...
Assigning codes to categorical and missing values
Data checks complete
Number of cases with positive D values: 869
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 1
Creating dummy variables
Rereading data

```

Total #cases	#cases w/ miss. D	#missing ord. vals	#X-var	#N-var	#F-var	#S-var
920	0	0	1	0	0	0
#M-var	#B-var	#C-var				

```

      0      1      2
Offset variable in column:      6
No. cases used for training: 920
Finished reading data file
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): poi.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: poi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < poi.in

```

5.8.2 Results

```

Poisson regression tree
No truncation of predicted values
Pruning by cross-validation
Data description file: lungcancer.dsc
Training sample file: lungcancer.txt
Missing value code: NA
Records in data file start on line 1
Dependent variable is deaths
Piecewise linear model
Number of records in data file: 920
Length of longest entry in data file: 8
Number of cases with positive D values: 869
Number of dummy variables created: 1

```

```

Summary information (without x variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
z=offset variable

```

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	county	c			115	
2	sex	b			2	
3	agegp	c			4	
4	deaths	d	0.000	1046.		
6	logpop	z	4.828	10.96		
===== Constructed variables =====						
7	sex.M	f	0.000	1.000		

```

      Total #cases w/ #missing
#cases  miss. D ord. vals #X-var #N-var #F-var #S-var
      920      0      0      1      0      0      0
#P-var #M-var #B-var #C-var #I-var
      0      0      1      2      0
Offset variable in column 6
No. cases used for training: 920

```

```

Missing values imputed with node means for regression
Nodewise 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: 1.0000
Max. number of split levels: 10
Min. node sample size: 45
Number of SE's for pruned tree: 0.5000

```

Size and CV Loss and SE of subtrees:

Tree	#Tnodes	Mean Loss	SE(Mean)	BSE(Mean)	Median Loss	BSE(Median)
1	7	2.254E+00	3.289E-01	2.746E-01	1.776E+00	4.222E-01
2	6	2.254E+00	3.289E-01	2.746E-01	1.776E+00	4.222E-01
3	5	2.254E+00	3.289E-01	2.746E-01	1.776E+00	4.222E-01
4	4	2.254E+00	3.289E-01	2.746E-01	1.776E+00	4.222E-01
5**	3	2.235E+00	3.278E-01	2.766E-01	1.721E+00	4.175E-01
6	2	4.857E+00	8.176E-01	5.168E-01	4.388E+00	7.094E-01
7	1	9.439E+00	1.406E+00	1.131E+00	9.048E+00	1.683E+00

```

0-SE tree based on mean is marked with * and has 3 terminal nodes
0-SE tree based on median is marked with + and has 3 terminal nodes
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.

Rate is mean of Y/exp(offset)

Cases fit give the number of cases used to fit node

Deviance is mean residual deviance for all cases in node

Node label	Total cases	Cases fit	Matrix rank	Node rate	Node deviance	Split variable	Other variables
1	920	920	2	1.382E-02	9.179E+00	agegp	
2T	230	230	2	5.493E-03	1.863E+00	sex	
3	690	690	2	1.763E-02	4.357E+00	agegp	

```

6T      230      230      2  1.339E-02  3.003E+00  sex
7T      460      460      2  2.093E-02  1.802E+00  agegp

```

Number of terminal nodes of final tree: 3

Total number of nodes of final tree: 5

Second best split variable (based on curvature test) at root node is sex

Regression tree:

At splits on categorical variables, values not in training data go to the right

Node 1: agegp = "45-54"

Node 2: deaths sample rate = 0.54928582E-002

Node 1: agegp /= "45-54"

Node 3: agegp = "55-64"

Node 6: deaths sample rate = 0.13389777E-001

Node 3: agegp /= "55-64"

Node 7: deaths sample rate = 0.20932715E-001

WARNING: p-values below are not adjusted for split search. For a bootstrap solution, see Loh et al. (2016), "Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

Node 1: Intermediate node

A case goes into Node 2 if agegp = "45-54"

agegp mode = "45-54"

Coefficients of loglinear regression function:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	-5.1717	-366.86	0.0000			
sex.M	1.4370	89.637	0.0000	0.0000	0.50000	1.0000

Node mean for offset variable = 6.7275

Node 2: Terminal node

Coefficients of loglinear regression function:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	-5.8337	-161.46	0.33307E-15			
sex.M	1.0384	24.437	0.22204E-15	0.0000	0.50000	1.0000

Node mean for offset variable = 6.8567

Node 3: Intermediate node

A case goes into Node 6 if agegp = "55-64"

agegp mode = "55-64"

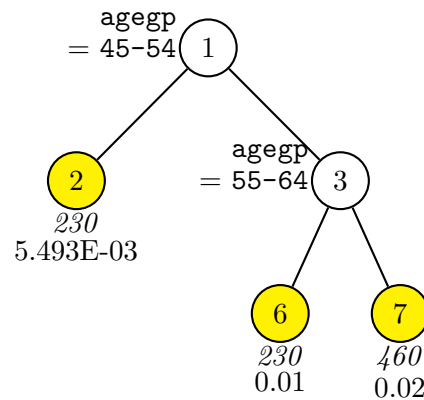


Figure 17: GUIDE v.29.6 0.50-SE pruned multiple linear Poisson regression tree for predicting rate of **deaths**. Maximum number of split levels is 10 and minimum node sample size is 45. At each split, an observation goes to the left branch if and only if the condition is satisfied. For splits on categorical variables, values not present in the training sample go to the right. Sample size (*in italics*) and sample rate printed below nodes. Second best split variable at root node is **sex**.

Node 6: Terminal node

Coefficients of loglinear regression function:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	-5.1175	-199.84	0.0000			
sex.M	1.2854	43.868	0.0000	0.0000	0.50000	1.0000

Node mean for offset variable = 6.9199

Node 7: Terminal node

Coefficients of loglinear regression function:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	-4.9065	-256.88	0.0000			
sex.M	1.7137	79.680	0.22204E-15	0.0000	0.50000	1.0000

Node mean for offset variable = 6.5666

Observed and fitted values are stored in poi.fit

LaTeX code for tree is in poi.tex

The results show that the death rate increases with age and that the rate for males is consistently higher than that for females. The tree diagram is given in Figure 17.

5.9 Censored response: heart attack data

GUIDE can fit a piecewise-constant, piecewise-simple linear, or piecewise multiple linear proportional hazards regression model to censored response data. Using usual notation, let $\lambda(\mathbf{x}, t)$ denote the hazard rate at time t for a subject with covariate vector \mathbf{x} . In a proportional hazards model, the hazard rate can be factored as $\lambda(\mathbf{x}, t) = \lambda_0(t)f(\mathbf{x}, \boldsymbol{\beta})$, where $\lambda_0(t)$ is a “baseline” hazard rate that is independent of the covariates and $f(\mathbf{x}, \boldsymbol{\beta})$ is a function of \mathbf{x} and some coefficients $\boldsymbol{\beta}$, independent of t . The Cox proportional hazards model uses $\lambda(\mathbf{x}, t) = \lambda_0(t) \exp(\boldsymbol{\beta}'\mathbf{x})$. GUIDE fits the more general model

$$\lambda(\mathbf{x}, t) = \lambda_0(t) \sum_i I(\mathbf{x} \in S_i) \exp(\boldsymbol{\beta}_i'\mathbf{x}),$$

where S_i is a set corresponding node i and $\boldsymbol{\beta}_i$ is its associated coefficient vector. See [Loh et al. \(2015\)](#) for more details.

We illustrate the piecewise-constant model $\lambda(\mathbf{x}, t) = \lambda_0(t) \sum_i I(\mathbf{x} \in S_i) \exp(\boldsymbol{\beta}_i'\mathbf{x})$ with a data set from the Worcester Heart Attack Study analyzed in [Hosmer et al. \(2008\)](#). The data are in the file `whas500.csv` and the description file in `whas500.dsc` whose contents are repeated below.

```
whas500.csv
NA
1
1 id x
2 age n
3 gender c
4 hr n
5 sysbp n
6 diasbp n
7 bmi n
8 cvd c
9 afb c
10 sho c
11 chf c
12 av3 c
13 miord c
14 mitype c
15 year c
16 admitdate x
17 disdate x
18 fdate x
19 los n
20 dstat x
21 lenfol t
```

22 fstat d

The goal of the study is to observe survival rates following hospital admission for acute myocardial infarction. The response variable is `lenfol`, which stands for total length of follow-up in days. Variable `fstat` is status at last follow-up (0=alive, 1=dead) and variable `chf` is congestive heart complications (0=no, 1=yes).

```

0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: whas500.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: whas500.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
    1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
    5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 4
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended if R variable is present,
    unless there are too many N, F or B variables)
Choose 2 for simple linear in one N or F variable + R (if present)
Choose 3 to fit a constant + R (if present)
1: multiple linear, 2: simple linear, 3: constant ([1:3], <cr>=3):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: whas500.dsc
Reading data description file ...
Training sample file: whas500.csv
Missing value code: NA
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is fstat
Reading data file ...
Number of records in data file: 500
Length of longest entry in data file: 10
Checking for missing values ...
Total number of cases: 500
    Column  Categorical    No. of    No. of missing
    number  variable      levels    observations
        3  gender          2             0

```

8	cvd	2	0
9	afb	2	0
10	sho	2	0
11	chf	2	0
12	av3	2	0
13	miord	2	0
14	mitype	2	0
15	year	3	0

Re-checking data ...

Assigning codes to categorical and missing values

Data checks complete

Smallest uncensored T: 1.0000

No. complete cases excluding censored T < smallest uncensored T: 500

No. cases used to compute baseline hazard: 500

No. cases with D=1 and T >= smallest uncensored: 215

Rereading data

Total #cases	#cases w/ miss. D	#missing ord. vals	#X-var	#N-var	#F-var	#S-var
500	0	0	5	0	0	6
#M-var	#B-var	#C-var				
0	0	9				

Survival time variable in column: 21

Event indicator variable in column: 22

Proportion uncensored among nonmissing T and D variables: .430

No. cases used for training: 500

Finished reading data file

Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):

Input file name to store LaTeX code (use .tex as suffix): whas500.tex

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

Input name of file to store node ID and fitted value of each case: whas500.fit

Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):

Input file is created!

Run GUIDE with the command: guide < whas500.in

5.9.1 Results

Proportional hazards regression with relative risk estimates

Pruning by cross-validation

Data description file: whas500.dsc

Training sample file: whas500.csv

Missing value code: NA

Records in data file start on line 1

Warning: N variables changed to S

Dependent variable is fstat

Piecewise constant model

Number of records in data file: 500
 Length of longest entry in data file: 10
 Smallest uncensored T: 1.0000
 No. complete cases excluding censored T < smallest uncensored T: 500
 No. cases used to compute baseline hazard: 500
 No. cases with D=1 and T >= smallest uncensored: 215

Summary information (without x variables)

d=dependent, b=split and fit cat variable using indicator variables,
 c=split-only categorical, i=fit-only categorical (via indicators),
 s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
 m=missing-value flag variable, p=periodic variable, w=weight
 t=survival time variable

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
2	age	s	30.00	104.0		
3	gender	c			2	
4	hr	s	35.00	186.0		
5	sysbp	s	57.00	244.0		
6	diasbp	s	6.000	198.0		
7	bmi	s	13.05	44.84		
8	cvd	c			2	
9	afb	c			2	
10	sho	c			2	
11	chf	c			2	
12	av3	c			2	
13	miord	c			2	
14	mitype	c			2	
15	year	c			3	
19	los	s	0.000	47.00		
21	lenfol	t	1.000	2358.		
22	fstat	d	0.000	1.000		
===== Constructed variables =====						
23	lnbasehaz0	z	-4.135	0.9755		

Total #cases	#cases w/ miss. D	#missing ord. vals	#X-var	#N-var	#F-var	#S-var
500	0	0	5	0	0	6
#P-var	#M-var	#B-var	#C-var	#I-var		
0	0	0	9	0		

Survival time variable in column: 21

Event indicator variable in column: 22

Proportion uncensored among nonmissing T and D variables: 0.430

No. cases used for training: 500

Nodewise 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: 1.0000
 Max. number of split levels: 10
 Min. node sample size: 2
 Number of iterations: 5
 Number of SE's for pruned tree: 0.5000

Size and CV Loss and SE of subtrees:

Tree	#Tnodes	Mean Loss	SE(Mean)	BSE(Mean)	Median Loss	BSE(Median)
1	88	1.823E+00	1.271E-01	8.283E-02	1.755E+00	1.177E-01
2	87	1.823E+00	1.271E-01	8.283E-02	1.755E+00	1.177E-01
3	86	1.823E+00	1.271E-01	8.283E-02	1.755E+00	1.177E-01
4	85	1.823E+00	1.271E-01	8.283E-02	1.755E+00	1.177E-01
5	84	1.823E+00	1.271E-01	8.207E-02	1.755E+00	1.175E-01
6	83	1.820E+00	1.270E-01	8.132E-02	1.755E+00	1.138E-01
7	82	1.818E+00	1.269E-01	8.223E-02	1.755E+00	1.162E-01
8	81	1.818E+00	1.269E-01	8.223E-02	1.755E+00	1.162E-01
9	80	1.818E+00	1.269E-01	8.223E-02	1.755E+00	1.162E-01
10	79	1.818E+00	1.269E-01	8.218E-02	1.755E+00	1.160E-01
11	78	1.818E+00	1.269E-01	8.218E-02	1.755E+00	1.160E-01
12	77	1.818E+00	1.269E-01	8.218E-02	1.755E+00	1.160E-01
13	75	1.818E+00	1.269E-01	8.225E-02	1.755E+00	1.164E-01
14	74	1.818E+00	1.269E-01	8.221E-02	1.755E+00	1.161E-01
15	73	1.817E+00	1.269E-01	8.203E-02	1.755E+00	1.152E-01
16	72	1.817E+00	1.269E-01	8.203E-02	1.755E+00	1.152E-01
17	71	1.817E+00	1.269E-01	8.203E-02	1.755E+00	1.152E-01
18	69	1.816E+00	1.276E-01	8.091E-02	1.755E+00	1.093E-01
19	67	1.814E+00	1.271E-01	8.103E-02	1.767E+00	1.032E-01
20	66	1.815E+00	1.273E-01	8.310E-02	1.762E+00	1.111E-01
21	65	1.804E+00	1.254E-01	8.580E-02	1.762E+00	1.086E-01
22	63	1.791E+00	1.250E-01	8.434E-02	1.762E+00	1.095E-01
23	62	1.780E+00	1.225E-01	8.660E-02	1.739E+00	1.208E-01
24	61	1.774E+00	1.215E-01	8.856E-02	1.739E+00	1.226E-01
25	60	1.774E+00	1.215E-01	8.856E-02	1.739E+00	1.226E-01
26	57	1.754E+00	1.186E-01	7.356E-02	1.752E+00	1.184E-01
27	56	1.754E+00	1.186E-01	7.356E-02	1.752E+00	1.184E-01
28	54	1.754E+00	1.186E-01	7.356E-02	1.752E+00	1.184E-01
29	52	1.731E+00	1.167E-01	6.183E-02	1.752E+00	1.150E-01
30	51	1.731E+00	1.167E-01	6.183E-02	1.752E+00	1.150E-01
31	42	1.730E+00	1.167E-01	6.189E-02	1.748E+00	1.159E-01
32	39	1.729E+00	1.167E-01	6.282E-02	1.748E+00	1.159E-01
33	38	1.729E+00	1.167E-01	6.282E-02	1.748E+00	1.159E-01
34	37	1.709E+00	1.162E-01	6.447E-02	1.742E+00	1.313E-01
35	36	1.679E+00	1.145E-01	6.345E-02	1.682E+00	1.449E-01

36	34	1.669E+00	1.137E-01	6.907E-02	1.676E+00	1.449E-01
37	31	1.642E+00	1.102E-01	7.362E-02	1.613E+00	1.505E-01
38	29	1.605E+00	1.104E-01	6.757E-02	1.578E+00	1.190E-01
39	28	1.601E+00	1.102E-01	6.973E-02	1.577E+00	1.329E-01
40	26	1.544E+00	1.064E-01	6.661E-02	1.567E+00	1.227E-01
41	25	1.539E+00	1.044E-01	6.517E-02	1.538E+00	1.096E-01
42	22	1.539E+00	1.044E-01	6.517E-02	1.538E+00	1.096E-01
43	19	1.459E+00	9.271E-02	5.145E-02	1.407E+00	8.773E-02
44	18	1.446E+00	9.228E-02	5.220E-02	1.398E+00	8.546E-02
45	17	1.409E+00	8.944E-02	6.156E-02	1.388E+00	1.198E-01
46	16	1.409E+00	8.960E-02	6.062E-02	1.401E+00	1.174E-01
47	12	1.389E+00	8.864E-02	6.014E-02	1.395E+00	1.094E-01
48*	8	1.217E+00	7.530E-02	3.647E-02	1.243E+00	5.111E-02
49--	6	1.230E+00	7.255E-02	3.753E-02	1.244E+00	5.228E-02
50**	5	1.236E+00	7.117E-02	3.825E-02	1.261E+00	5.070E-02
51++	4	1.271E+00	7.080E-02	3.980E-02	1.266E+00	3.752E-02
52	3	1.300E+00	7.015E-02	3.646E-02	1.319E+00	3.915E-02
53	2	1.325E+00	6.506E-02	2.839E-02	1.313E+00	3.556E-02
54	1	1.503E+00	5.698E-02	2.544E-02	1.484E+00	3.699E-02

0-SE tree based on mean is marked with * and has 8 terminal nodes

0-SE tree based on median is marked with + and has 8 terminal nodes

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.

Rel. risk is mean risk relative to sample average ignoring covariates

Cases fit give the number of cases used to fit node

Deviance is mean residual deviance for all cases in node

Node label	Total cases	Cases fit	Matrix rank	Node rel.risk	Node deviance	Split variable	Other variables
1	500	500	1	1.000E+00	1.505E+00	age	
2	244	244	1	3.726E-01	9.913E-01	chf	
4T	195	195	1	2.124E-01	7.383E-01	year	
5T	49	49	1	1.110E+00	1.413E+00	miord	
3	256	256	1	1.890E+00	1.526E+00	chf	
6	150	150	1	1.365E+00	1.469E+00	age	
12T	120	120	1	1.063E+00	1.360E+00	cvd	
13T	30	30	1	3.322E+00	1.278E+00	year	
7T	106	106	1	3.028E+00	1.372E+00	sho	

```

Number of terminal nodes of final tree: 5
Total number of nodes of final tree: 9
Second best split variable (based on curvature test) at root node is chf

Regression tree:
At splits on categorical variables, values not in training data go to the right

Node 1: age <= 71.500000
  Node 2: chf = "0"
    Node 4: Risk relative to sample average ignoring covariates = 0.21235169
    Node 2: chf /= "0"
      Node 5: Risk relative to sample average ignoring covariates = 1.1095575
  Node 1: age > 71.500000 or NA
    Node 3: chf = "0"
      Node 6: age <= 85.500000
        Node 12: Risk relative to sample average ignoring covariates = 1.0633351
        Node 6: age > 85.500000 or NA
          Node 13: Risk relative to sample average ignoring covariates = 3.3221527
      Node 3: chf /= "0"
        Node 7: Risk relative to sample average ignoring covariates = 3.0276016

```

```

*****

```

WARNING: p-values below are not adjusted for split search. For a bootstrap solution, see Loh et al. (2016), "Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables", *Statistics in Medicine*, v.35, 4837-4855.

```

Node 1: Intermediate node
A case goes into Node 2 if age <= 71.500000
age mean = 69.846000
Coefficients of log-relative risk function:
Regressor    Coefficient  t-stat      p-value
Constant    -0.35381E-01
Predicted relative risk = 1.0000000
-----
Node 2: Intermediate node
A case goes into Node 4 if chf = "0"
chf mode = "0"
-----
Node 4: Terminal node
Coefficients of log-relative risk function:
Regressor    Coefficient  t-stat      p-value
Constant     -1.5849

```

```

Predicted relative risk = 0.21235169
-----
Node 5: Terminal node
Coefficients of log-relative risk function:
Regressor    Coefficient  t-stat      p-value
Constant     0.68580E-01
Predicted relative risk = 1.1095575
-----
Node 3: Intermediate node
A case goes into Node 6 if chf = "0"
chf mode = "0"
-----
Node 6: Intermediate node
A case goes into Node 12 if age <= 85.500000
age mean = 80.666667
-----
Node 12: Terminal node
Coefficients of log-relative risk function:
Regressor    Coefficient  t-stat      p-value
Constant     0.26029E-01
Predicted relative risk = 1.0633351
-----
Node 13: Terminal node
Coefficients of log-relative risk function:
Regressor    Coefficient  t-stat      p-value
Constant     1.1652
Predicted relative risk = 3.3221527
-----
Node 7: Terminal node
Coefficients of log-relative risk function:
Regressor    Coefficient  t-stat      p-value
Constant     1.0724
Predicted relative risk = 3.0276016
-----

Observed and fitted values are stored in whas500.fit
LaTeX code for tree is in whas500.tex

```

The tree model, given in Figure 18, shows that risk of death is lowest (0.21 relative to the sample average for the whole data set) for those younger than 72 with no congestive heart complications. The groups with the highest risks (3.03–3.32 relative to average) are those older than 71 with congestive heart complications and those older than 85 without congestive heart complications.

The top few lines of the file `whas500.fit` and its column definitions are:

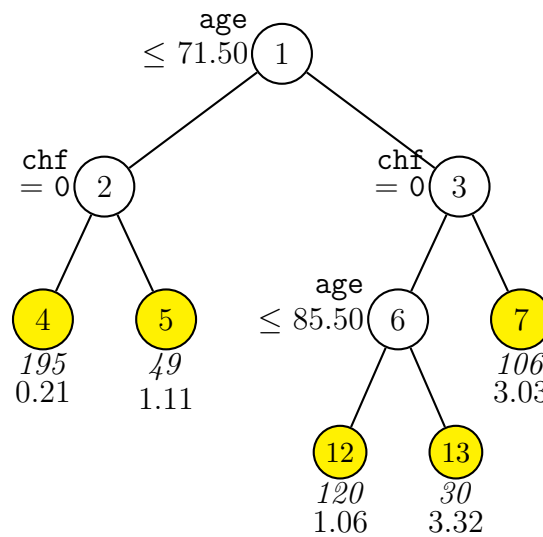


Figure 18: GUIDE v.29.6 0.50-SE pruned piecewise constant relative risk regression tree for predicting **fstat**. Maximum number of split levels is 10 and minimum node sample size is 2. At each split, an observation goes to the left branch if and only if the condition is satisfied. For splits on categorical variables, values not present in the training sample go to the right. Sample size (*in italics*) and mean relative risks (relative to sample average ignoring covariates) printed below nodes. Second best split variable at root node is **chf**.

train	node	survivaltime	logbasecumhaz	relativerisk	survivalprob	mediansurvtime
y	12	2.178000E+03	-7.667985E-02	1.063335E+00	3.865048E-01	1.553833E+03
y	4	2.172000E+03	-7.667985E-02	2.123517E-01	8.270912E-01	2.354277E+03
y	4	2.190000E+03	-7.667985E-02	2.123517E-01	8.270912E-01	2.354277E+03
y	5	2.970000E+02	-1.320296E+00	1.109557E+00	7.512523E-01	1.534963E+03

The columns are:

train: “y” if the observation is used for model fitting, “n” if not.

node: terminal node label of observation.

survivaltime: observed survival time t .

logbasecumhaz: log of the estimated baseline cumulative hazard function $\log \Lambda_0(t) = \log \int_0^t \lambda_0(u) du$ at observed time t .

relativerisk: $\exp(\beta' \mathbf{x} - \beta_*)$, risk of death relative to the average for the sample, where \mathbf{x} is the covariate vector of the observation, β is the estimated regression coefficient vector in the node, and β_* is the coefficient in the constant model $\lambda_0(t) \exp(\beta_*)$ fitted to all the training cases in the root node. Because a constant is fitted to each node here, $\beta_* = -0.035381$ is the value of β at the root node. For example, the first subject, which is in node 12, has $\beta = 0.026029$ and so **relativerisk** $= \exp(\beta - \beta_*) = \exp(0.026029 + 0.035381) = 1.063335$.

survivalprob: probability that the subject survives up to observed time t . For the first subject, this is

$$\begin{aligned}
 \exp\{-\Lambda_0(t) \exp(\beta' \mathbf{x})\} &= \exp\{-\exp(\beta_* + \text{logbasecumhaz}) \times \text{relativerisk}\} \\
 &= \exp(-\exp(-0.035381 - 0.07667985) \times 1.063335) \\
 &= 0.3865049.
 \end{aligned}$$

mediansurvtime: estimated median survival time t such that $\exp\{-\Lambda_0(t) \exp(\beta' \mathbf{x})\} = 0.5$, or, equivalently, $\Lambda_0(t) \exp(\beta' \mathbf{x}) = -\log(0.5)$, or $\text{logbasecumhaz}(t) = \log \log(2) - \beta' \mathbf{x}$, using linear interpolation of $\Lambda_0(t)$. Median survival times greater than the largest observed time have a trailing plus (+) sign.

5.10 Multi-response: public health data

GUIDE can fit a piecewise-constant regression model for two or more dependent variables simultaneously (Loh and Zheng, 2013). We demonstrate this with the data set `phs.dat` from a public health survey of about 120,000 respondents. There are three D variables, namely, total restricted activity days in the past 2 week (`raday`), number of doctor visits in the past 12 months (`visit`), and number of short-stay hospital days in the past 12 months (`hda12`). The description files `phs.dsc` given below lists 6 numeric and 9 categorical variables.

```
phs.dat
NA
1
1 phone c
2 sex c
3 age n
4 race c
5 marstat c
6 educ n
7 income n
8 poverty c
9 famsize n
10 condlist c
11 health n
12 latotal n
13 wkclass c
14 indus c
15 occup c
16 raday d
17 visit d
18 nacute x
19 hda12 d
20 lnvisit x
```

5.10.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: mult.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: mult.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
```

```

(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 5
  Option 5 is for multiresponse data
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: phs.dsc
Reading data description file ...
Training sample file: phs.dat
Missing value code: NA
Records in data file start on line 1
Warning: N variables changed to S
Number of D variables = 3
D variables are:
raday
visit
hda12
Multivariate or univariate split variable selection:
Choose multivariate if there is an order among the D variables; otherwise choose univariate
Input 1 for multivariate, 2 for univariate ([1:2], <cr>=2):
  Choose 2 because there is no order among the D variables
Input 1 to normalize D variables, 2 for no normalization ([1:2], <cr>=1):
Input 1 for equal, 2 for unequal weighting of D variables ([1:2], <cr>=1):
Reading data file ...
Number of records in data file: 119579
Length of longest entry in data file: 17
Checking for missing values ...
Total number of cases: 119579
Missing values found among categorical variables
Separate categories will be created for missing categorical variables

```

Column number	Categorical variable	No. of levels	No. of missing observations
1	phone	4	0
2	sex	2	0
4	race	3	0
5	marstat	7	720
8	poverty	2	11400
10	condlist	6	754
13	wkclass	8	63793
14	indus	14	64059
15	occup	14	64085

```

Re-checking data ...
Allocating missing value information
Assigning codes to categorical and missing values
Finished processing 5000 of 119579 observations
Finished processing 10000 of 119579 observations
Finished processing 15000 of 119579 observations
Finished processing 20000 of 119579 observations
Finished processing 25000 of 119579 observations
Finished processing 30000 of 119579 observations
Finished processing 35000 of 119579 observations
Finished processing 40000 of 119579 observations
Finished processing 45000 of 119579 observations
Finished processing 50000 of 119579 observations
Finished processing 55000 of 119579 observations
Finished processing 60000 of 119579 observations
Finished processing 65000 of 119579 observations
Finished processing 70000 of 119579 observations
Finished processing 75000 of 119579 observations
Finished processing 80000 of 119579 observations
Finished processing 85000 of 119579 observations
Finished processing 90000 of 119579 observations
Finished processing 95000 of 119579 observations
Finished processing 100000 of 119579 observations
Finished processing 105000 of 119579 observations
Finished processing 110000 of 119579 observations
Finished processing 115000 of 119579 observations
Data checks complete
Normalizing data
Creating missing value indicators
Some D variables have missing values
You can use all the data or only those with complete D values
Using only cases with complete D values will reduce the sample size
but allows the option of using PCA for split selection
Input 1 to use all obs, 2 to use obs with complete D values ([1:2], <cr>=2):
  Using only observations with complete D values allows more options
Rereading data
PCA can be used for variable selection
Do not use PCA if differential item functioning (DIF) scores are wanted
Input 1 to use PCA, 2 otherwise ([1:2], <cr>=2):
#cases w/ miss. D = number of cases with all D values missing
  Total #cases w/ #missing
  #cases   miss. D  ord. vals  #X-var  #N-var  #F-var  #S-var
119579      0      30722      2        0        0        6
  #M-var  #B-var  #C-var
      0      0      9
No. cases used for training: 60000

```

```

No. cases excluded due to 0 weight or missing D: 59579
Finished reading data file
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): mult.tex
A file by that name exists
Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):
Input 2 to save node IDs of observations, 1 otherwise ([1:2], <cr>=2):
Input name of file to store terminal node ID of each case: mult.nid
A file by that name exists
Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):
Input 2 to save fitted values at each terminal node; 1 otherwise ([1:2], <cr>=1): 2
Input name of file to store node fitted values: mult.fit
A file by that name exists
Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < mult.in

```

5.10.2 Results

```

Multi-response or longitudinal data without T variables
Pruning by cross-validation
Data description file: phs.dsc
Training sample file: phs.dat
Missing value code: NA
Records in data file start on line 1
Warning: N variables changed to S
Number of D variables = 3
Univariate split variable selection method
Mean-squared errors (MSE) are calculated from normalized D variables
D variables equally weighted
Piecewise constant model
Number of records in data file: 119579
Length of longest entry in data file: 17
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Some D variables have missing values
Model fitted to subset of observations with complete D values
Neither LDA nor PCA used

Summary information (without x variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight

```

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	phone	c			4	
2	sex	c			2	
3	age	s	0.000	99.00		
4	race	c			3	
5	marstat	c			7	720
6	educ	s	0.000	18.00		11162
7	income	s	0.000	26.00		21547
8	poverty	c			2	11400
9	famsize	s	1.000	26.00		
10	condlist	c			6	754
11	health	s	1.000	5.000		678
12	latotal	s	1.000	4.000		
13	wkclass	c			8	63793
14	indus	c			14	64059
15	occup	c			14	64085
16	raday	d	0.000	14.00		
17	visit	d	0.000	637.0		59579
19	hda12	d	0.000	286.0		

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

Total	#cases w/	#missing					
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var	
119579	0	30722	2	0	0	6	
#P-var	#M-var	#B-var	#C-var	#I-var			
0	0	0	9	0			

No. cases used for training: 60000

No. cases excluded due to 0 weight or missing D: 59579

No nodewise interaction tests

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: 30

Min. node sample size: 3000

Number of SE's for pruned tree: 0.5000

Size and CV Loss and SE of subtrees:

Tree	#Tnodes	Mean Loss	SE(Mean)	BSE(Mean)	Median Loss	BSE(Median)
1	14	1.451E+00	7.378E-02	5.390E-02	1.467E+00	5.753E-02
2	12	1.451E+00	7.378E-02	5.390E-02	1.467E+00	5.753E-02
3	11	1.509E+00	7.378E-02	5.067E-02	1.525E+00	2.597E-02
4	9	1.577E+00	7.379E-02	5.566E-02	1.587E+00	5.903E-02
5	8	1.624E+00	7.379E-02	5.075E-02	1.644E+00	5.546E-02

6	7	1.624E+00	7.379E-02	5.075E-02	1.644E+00	5.546E-02
7	6	1.590E+00	7.367E-02	5.661E-02	1.566E+00	6.778E-02
8**	5	1.424E+00	7.369E-02	6.384E-02	1.449E+00	8.818E-02
9	3	1.640E+00	7.380E-02	7.607E-02	1.632E+00	1.304E-01
10	2	1.706E+00	7.383E-02	6.099E-02	1.707E+00	7.733E-02
11	1	1.899E+00	7.710E-02	5.641E-02	1.925E+00	6.427E-02

0-SE tree based on mean is marked with * and has 5 terminal nodes

0-SE tree based on median is marked with + and has 5 terminal nodes

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.

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
1	60000	60000	1.310E+00	latotal
2T	5936	5936	7.660E+00	-
3	54064	54064	5.212E-01	health
6	50875	50875	4.330E-01	educ
12	42463	42463	4.373E-01	health
24T	31417	31417	2.844E-01	sex
25T	11046	11046	8.621E-01	age
13T	8412	8412	4.077E-01	age
7T	3189	3189	1.787E+00	-

Number of terminal nodes of final tree: 5

Total number of nodes of final tree: 9

Second best split variable (based on curvature test) at root node is health

Regression tree for multi-response data:

Node 1: lattotal <= 2.5000000

Node 2: Mean cost = 3.8429747

Node 1: lattotal > 2.5000000 or NA

Node 3: health <= 3.5000000 or NA

Node 6: educ <= 16.500000

Node 12: health <= 2.5000000

Node 24: Mean cost = 0.14269907

Node 12: health > 2.5000000 or NA


```

Node 25: Mean cost = 0.43253208
Node 6: educ > 16.500000 or NA
Node 13: Mean cost = 0.20454223
Node 3: health > 3.5000000
Node 7: Mean cost = 0.89637215

```

```

*****

```

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

```

Node 1: Intermediate node
A case goes into Node 2 if latotal <= 2.5000000
latotal mean = 3.7126500
Means of raday, visit, and hda12
  6.1067E-01  4.1094E+00  6.0488E-01
The above 3 numbers are the mean values of the 3 D variables.
-----
Node 2: Terminal node
Means of raday, visit, and hda12
  2.8630E+00  1.2474E+01  3.0076E+00
-----
Node 3: Intermediate node
A case goes into Node 6 if health <= 3.5000000 or NA
health mean = 1.9395511
-----
Node 6: Intermediate node
A case goes into Node 12 if educ <= 16.500000
educ mean = 10.876582
-----
Node 12: Intermediate node
A case goes into Node 24 if health <= 2.5000000
health mean = 1.8331401
-----
Node 24: Terminal node
Means of raday, visit, and hda12
  2.4687E-01  2.4306E+00  2.1584E-01
-----
Node 25: Terminal node
Means of raday, visit, and hda12
  4.5854E-01  3.8409E+00  4.4858E-01
-----
Node 13: Terminal node
Means of raday, visit, and hda12
  3.2834E-01  3.7666E+00  3.0777E-01
-----
Node 7: Terminal node

```

```
Means of raday, visit, and hda12
  1.2738E+00   6.9116E+00   1.2904E+00
-----
```

```
Case and node IDs are in file: mult.nid
Node fitted values are in file: mult.fit
LaTeX code for tree is in mult.tex
```

The tree is shown in Figure 19. The file `mult.fit` saves the mean values of the dependent variables in each terminal node:

node	raday	visit	hda12
2	0.28630E+01	0.12474E+02	0.30076E+01
24	0.24687E+00	0.24306E+01	0.21584E+00
25	0.45854E+00	0.38409E+01	0.44858E+00
13	0.32834E+00	0.37666E+01	0.30777E+00
7	0.12738E+01	0.69116E+01	0.12904E+01

The file `mult.nid` gives the terminal node number for each observation, *including* those that are not used to construct the tree (indicated by the letter “n” in the `train` column of the file).

5.11 Longitudinal response with varying time: wage data

The data come from a longitudinal study on the hourly wage of 888 male high-school dropouts (246 black, 204 Hispanic, 438 white), where the observation time points as well as their number (1–13) varied across individuals (Murnane et al., 1999; Singer and Willett, 2003). An earlier version of GUIDE was used to analyze the data in Loh and Zheng (2013).

The response variable is hourly wage (in 1990 dollars) and the predictor variables are `hgc` (highest grade completed; 6–12), `exper` (years in labor force; 0.001–12.7 yrs), and `race` (Black, Hispanic, and White). The data file `wagedat.txt` is in *wide format*, where each record refers to one individual. The description file `wagedsc.txt` is given below. Note that observation time points are marked as `t`.

```
wagedat.txt
NA
1
1 id x
2 hgc n
3 exper1 t
4 exper2 t
5 exper3 t
```

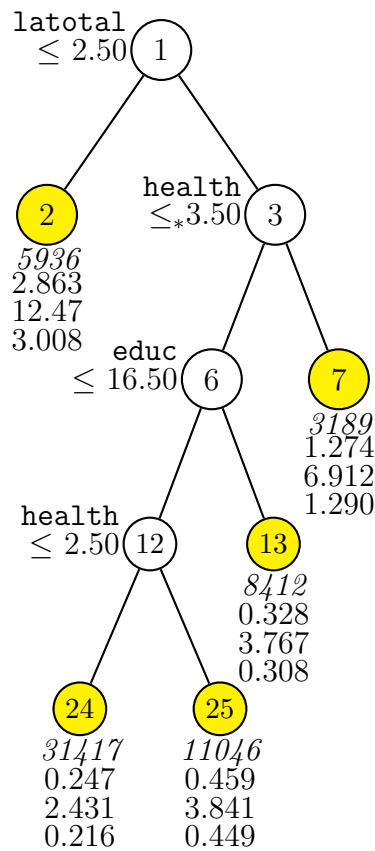


Figure 19: GUIDE v.29.6 0.50-SE pruned regression tree for predicting response variables `raday`, `visit`, and `hda12`, without using PCA at each node. Maximum number of split levels is 30 and minimum node sample size is 3000. At each split, an observation goes to the left branch if and only if the condition is satisfied. The symbol ' \leq_* ' stands for ' \leq or missing'. Sample sizes (*in italics*) and predicted values of `raday`, `visit`, and `hda12` printed below nodes. Second best split variable at root node is `health`.

```
6 exper4 t
7 exper5 t
8 exper6 t
9 exper7 t
10 exper8 t
11 exper9 t
12 exper10 t
13 exper11 t
14 exper12 t
15 exper13 t
16 postexp1 x
17 postexp2 x
18 postexp3 x
19 postexp4 x
20 postexp5 x
21 postexp6 x
22 postexp7 x
23 postexp8 x
24 postexp9 x
25 postexp10 x
26 postexp11 x
27 postexp12 x
28 postexp13 x
29 wage1 d
30 wage2 d
31 wage3 d
32 wage4 d
33 wage5 d
34 wage6 d
35 wage7 d
36 wage8 d
37 wage9 d
38 wage10 d
39 wage11 d
40 wage12 d
41 wage13 d
42 ged1 x
43 ged2 x
44 ged3 x
45 ged4 x
46 ged5 x
47 ged6 x
48 ged7 x
49 ged8 x
50 ged9 x
51 ged10 x
```

```

52 ged11 x
53 ged12 x
54 ged13 x
55 uerate1 x
56 uerate2 x
57 uerate3 x
58 uerate4 x
59 uerate5 x
60 uerate6 x
61 uerate7 x
62 uerate8 x
63 uerate9 x
64 uerate10 x
65 uerate11 x
66 uerate12 x
67 uerate13 x
68 race c

```

5.11.1 Input file creation

Because the default 0.5-SE rule yields a trivial tree with no splits, we show how the options can be changed to produce a tree with the 0-SE rule. Following is a session log.

```

0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: wage.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: wage.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
    1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
    5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 6
Input 1 for lowess smoothing, 2 for spline smoothing ([1:2], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
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 characters);

```

```

enclose with matching quotes if it has spaces: wagedsc.txt
Reading data description file ...
Training sample file: wagedat.txt
Missing value code: NA
Records in data file start on line 1
Warning: N variables changed to S
Number of D variables = 13
Number of D variables =          13
D variables are:
wage1
wage2
wage3
wage4
wage5
wage6
wage7
wage8
wage9
wage10
wage11
wage12
wage13
T variables are:
exper1
exper2
exper3
exper4
exper5
exper6
exper7
exper8
exper9
exper10
exper11
exper12
exper13
D variables can be grouped into segments to look for patterns
Input 1 for equal-sized groups, 2 for custom groups ([1:2], <cr>=1):
Input number of roughly equal-sized groups ([2:9], <cr>=3):
Input number of interpolating points for prediction ([10:100], <cr>=31):
Reading data file ...
Number of records in data file: 888
Length of longest entry in data file: 16
Checking for missing values ...
Total number of cases: 888
  Column  Categorical      No. of  No. of missing

```

```

number  variable      levels  observations
   68   race              3           0

Re-checking data ...
Allocating missing value information
Assigning codes to categorical and missing values
Data checks complete
Creating missing value indicators
Rereading data
#cases w/ miss. D = number of cases with all D values missing
      Total  #cases w/  #missing
      #cases  miss. D  ord. vals  #X-var  #N-var  #F-var  #S-var
        888         0         0       40       0       0       1
      #M-var  #B-var  #C-var
         0         0         1

No. cases used for training: 888
No. cases excluded due to 0 weight or missing D: 0
Finished reading data file
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
  This is where we choose the 0-SE pruning rule
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):
Default minimum node sample size is 44
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): wage.tex
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 terminal 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): 3
Input file name: wage.var
Input 2 to save node IDs of observations, 1 otherwise ([1:2], <cr>=2):
Input name of file to store terminal node ID of each case: wage.nid
Input 2 to save fitted values at each terminal node; 1 otherwise ([1:2], <cr>=1): 2
Input file name: wage.nid
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < wage.in
```

5.11.2 Results

```
Lowess smoothing
Longitudinal data with T variables
Pruning by cross-validation
Data description file: wagedsc.txt
Training sample file: wagedat.txt
Missing value code: NA
Records in data file start on line 1
Warning: N variables changed to S
Number of D variables = 13
Number of D variables: 13
D variables are:
wage1
wage2
wage3
wage4
wage5
wage6
wage7
wage8
wage9
wage10
wage11
wage12
wage13
T variables are:
exper1
exper2
```



```

exper3
exper4
exper5
exper6
exper7
exper8
exper9
exper10
exper11
exper12
exper13

```

Number of records in data file: 888

Length of longest entry in data file: 16

Model fitted to subset of observations with complete D values

Summary information (without x variables)

d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
2	hgc	s	6.000	12.00		
3	exper1	t	0.1000E-02	5.637		
4	exper2	t	0.000	7.584		38
5	exper3	t	0.000	9.777		77
6	exper4	t	0.000	10.81		124
7	exper5	t	0.000	11.78		159
8	exper6	t	0.000	10.59		233
9	exper7	t	0.000	11.28		325
10	exper8	t	0.000	10.58		428
11	exper9	t	0.000	11.62		551
12	exper10	t	0.000	12.26		678
13	exper11	t	0.000	11.98		791
14	exper12	t	0.000	12.56		856
15	exper13	t	0.000	12.70		882
29	wage1	d	2.030	68.65		
30	wage2	d	2.069	50.40		38
31	wage3	d	2.046	34.50		77
32	wage4	d	2.117	33.15		124
33	wage5	d	2.104	49.30		159
34	wage6	d	2.208	74.00		233
35	wage7	d	2.104	47.28		325
36	wage8	d	2.316	37.71		428
37	wage9	d	2.529	46.11		551

38	wage10	d	2.998	56.54		678
39	wage11	d	4.084	22.20		791
40	wage12	d	3.432	46.20		856
41	wage13	d	4.563	7.776		882
68	race	c			3	

Total #cases	w/ miss.	#missing D	ord. vals	#X-var	#N-var	#F-var	#S-var
888		0	0	40	0	0	1
#P-var	#M-var	#B-var	#C-var	#I-var			
0	0	0	1	0			

No. cases used for training: 888

No. cases excluded due to 0 weight or missing D: 0

No nodewise interaction tests

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

Min. node sample size: 44

Warning 2: something wrong with 2nd best split variable

Number of SE's for pruned tree: 0.000

Size and CV Loss and SE of subtrees:

Tree	#Tnodes	Mean Loss	SE(Mean)	BSE(Mean)	Median Loss	BSE(Median)
1	9	1.258E+02	1.044E+01	8.446E+00	1.206E+02	1.515E+01
2	7	1.258E+02	1.044E+01	8.446E+00	1.206E+02	1.515E+01
3	5	1.243E+02	1.052E+01	8.400E+00	1.181E+02	1.527E+01
4**	3	1.238E+02	1.053E+01	8.513E+00	1.175E+02	1.583E+01
5	2	1.239E+02	1.056E+01	8.631E+00	1.175E+02	1.562E+01
6++	1	1.244E+02	1.064E+01	8.700E+00	1.157E+02	1.577E+01

0-SE tree based on mean is marked with * and has 3 terminal nodes

0-SE tree based on median is marked with + and has 1 terminal nodes

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
1	888	888	1.222E+02	hgc
2T	577	577	1.040E+02	race
3	311	311	1.513E+02	race
6T	95	95	1.079E+02	-
7T	216	216	1.680E+02	hgc

Number of terminal nodes of final tree: 3

Total number of nodes of final tree: 5

Regression tree for longitudinal data:

At splits on categorical variables, values not in training data go to the right

Node 1: hgc <= 9.5000000

Node 2: Mean cost = 103.80991

Node 1: hgc > 9.5000000 or NA

Node 3: race = "black"

Node 6: Mean cost = 106.75431

Node 3: race /= "black"

Node 7: Mean cost = 167.22580

Node 1: Intermediate node

A case goes into Node 2 if hgc <= 9.5000000

hgc mean = 8.9166667

Node 2: Terminal node

Node 3: Intermediate node

A case goes into Node 6 if race = "black"

race mode = "white"

Node 6: Terminal node

Node 7: Terminal node

Case and node IDs are in file: wage.nid

Node fitted values are in file: wage.fit

LaTeX code for tree is in wage.tex

Split and fit variable names are stored in wage.var

Figure 20 shows the tree and Figure 21 plots lowess-smoothed curves of mean wage in the two terminal nodes. The figure is produced by the following R code.

```
z <- read.table("widewage.txt",header=FALSE)
names(z) <- c("id","hgc","exper1","exper2","exper3","exper4","exper5","exper6",
             "exper7","exper8","exper9","exper10","exper11","exper12","exper13",
             "postexp1","postexp2","postexp3","postexp4","postexp5","postexp6",
             "postexp7","postexp8","postexp9","postexp10","postexp11","postexp12",
             "postexp13","wage1","wage2","wage3","wage4","wage5","wage6","wage7",
             "wage8","wage9","wage10","wage11","wage12","wage13","ged1","ged2",
             "ged3","ged4","ged5","ged6","ged7","ged8","ged9","ged10","ged11",
             "ged12","ged13","uerate1","uerate2","uerate3","uerate4","uerate5",
             "uerate6","uerate7","uerate8","uerate9","uerate10","uerate11",
             "uerate12","uerate13","race")
exper <- c(z$exper1,z$exper2,z$exper3,z$exper4,z$exper5,z$exper6,z$exper7,
          z$exper8,z$exper9,z$exper10,z$exper11,z$exper12,z$exper13)
wage <- c(z$wage1,z$wage2,z$wage3,z$wage4,z$wage5,z$wage6,z$wage7,z$wage8,
          z$wage9,z$wage10,z$wage11,z$wage12,z$wage13)
xr <- range(exper,na.rm=TRUE)
yr <- range(wage,na.rm=TRUE)

guide.fit <- read.table("wage.fit",header=TRUE)
g.node <- guide.fit$node
g.start <- guide.fit$t.start
g.end <- guide.fit$t.end
n <- length(g.node)
m <- dim(guide.fit)[2]
npts <- m-3 # number of time points for plotting

xvals <- guide.fit[,2:3]
xvals <- as.numeric(unlist(xvals))
yvals <- guide.fit[,4:m]
yvals <- as.numeric(unlist(yvals))
plot(range(xvals),range(yvals),type="n",xlab="exper (years)",ylab="hourly wage ($)")
leg.col <- c("blue","red","black")
leg.lty <- c(1,2,3)
for(i in 1:n){
  node <- g.node[i]
  start <- g.start[i]
  end <- g.end[i]
  gap <- (end-start)/(npts-1)
  x <- start+(0:(npts-1))*gap
  y <- as.numeric(guide.fit[i,4:m])
  lines(x,y,col=leg.col[i],lty=leg.lty[i])
}
```

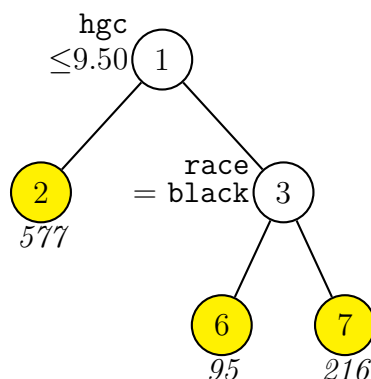


Figure 20: GUIDE v.29.6 0-SE pruned regression tree for predicting longitudinal variables `wage1`, `wage2`, etc. Maximum number of split levels is 10 and minimum node sample size is 44. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample sizes (*in italics*) printed below nodes.

```

}
leg.txt <- c(expression(paste("hgc" <= 9)),
             expression(paste("black, hgc" > 9)),
             expression(paste("not black, hgc" > 9))
             )
legend("topleft", legend=leg.txt, lty=leg.lty, col=leg.col, lwd=2)

```

The plotting values are obtained from the result file `wage.fit` whose contents are given below. The first column gives the node number and the next two columns the start and end of the times at which fitted values are computed. The other columns give the fitted values equally spaced between the start and end times.

node	t.start	t.end	fitted1	fitted2	fitted3	fitted4	fitted5	fitted6	fitted7	fitted8	fitted9	fitted10
2	0.10000E-02	0.12700E+02	0.48875E+01	0.51221E+01	0.53241E+01	0.54668E+01	0.55738E+01	0.56738E+01	0.57668E+01	0.58538E+01	0.59338E+01	0.60068E+01
6	0.80000E-02	0.12558E+02	0.61270E+01	0.58648E+01	0.57522E+01	0.57674E+01	0.57653E+01	0.57653E+01	0.57653E+01	0.57653E+01	0.57653E+01	0.57653E+01
7	0.20000E-02	0.12045E+02	0.56786E+01	0.58892E+01	0.60859E+01	0.62420E+01	0.63533E+01	0.64420E+01	0.65120E+01	0.65720E+01	0.66220E+01	0.66620E+01

The contents of the file `wage.var` are given below. The 1st column gives the node number. The 2nd column is a letter, with `t` indicating that the node is terminal and `c`, `s`, or `n` indicating an intermediate node split on a `c`, `n` or `s` variable. The 3rd column gives the name of the variable used to split the node; the name `NONE` is used if a terminal node cannot be split by any variable. The 4th column gives the name of the interacting variable if there is one; otherwise the name of the split variable is repeated. For a non-terminal node, the integer in the 5th column gives the number of split values to follow on the line.

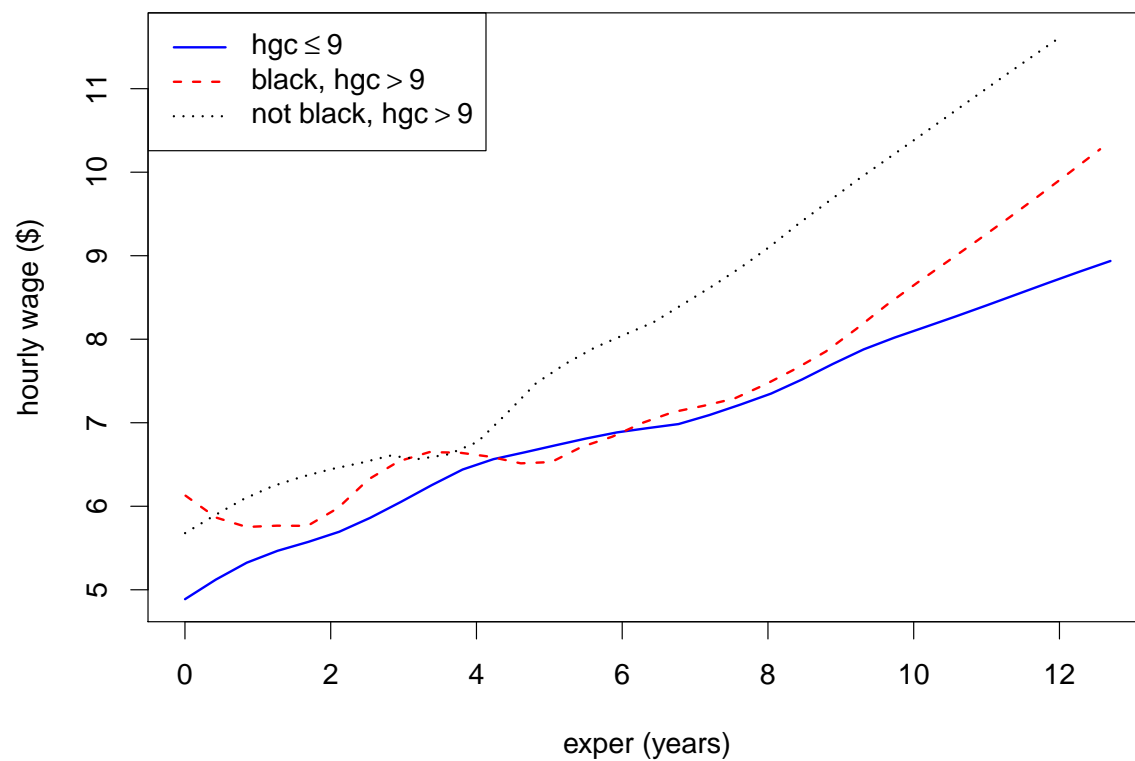


Figure 21: Lowess-smoothed mean wage curves in the terminal nodes of Figure 20.

```

1 s hgc hgc      1  0.9500000000E+01
2 t race race    0.0000000000E+00
3 c race race    1  "black"
6 t NONE NONE    0.0000000000E+00
3 c race race    1  "black"
7 t hgc hgc      0.0000000000E+00

```

5.12 Multiple longitudinal series: mother and child health

The above method may be extended to deal with more than one longitudinal response series by concatenating them together, making sure to set the group boundaries so that responses in different series are not grouped together. Because this technique is applicable only if the longitudinal observations take place at the same time points, no time variables are needed. We illustrate this with an example from [Diggle et al. \(2002\)](#) on maternal stress and child health which were previously analyzed in [Loh and Zheng \(2013\)](#). The data and description files are `mscm.txt` and `mscmboth.dsc`.

5.12.1 Input file creation

```

0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: both.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: both.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
    1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
    5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 5
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: mscmboth.dsc
Reading data description file ...
Training sample file: widedata.txt
Missing value code: NA
Records in data file start on line 1

```

Warning: N variables changed to S

Number of D variables = 56

D variables are:

stress1
stress2
stress3
stress4
stress5
stress6
stress7
stress8
stress9
stress10
stress11
stress12
stress13
stress14
stress15
stress16
stress17
stress18
stress19
stress20
stress21
stress22
stress23
stress24
stress25
stress26
stress27
stress28
illness1
illness2
illness3
illness4
illness5
illness6
illness7
illness8
illness9
illness10
illness11
illness12
illness13
illness14
illness15


```

illness16
illness17
illness18
illness19
illness20
illness21
illness22
illness23
illness24
illness25
illness26
illness27
illness28

```

Multivariate or univariate split variable selection:

Choose multivariate if there is an order among the D variables; otherwise choose univariate

Input 1 for multivariate, 2 for univariate ([1:2], <cr>=1):

The D vector can be grouped into segments to look for patterns

Input 1 for roughly equal groups, 2 otherwise

Input your selection ([1:2], <cr>=1):

Input number of groups ([2:30], <cr>=3): 8

Input 1 to normalize D variables, 2 for no normalization ([1:2], <cr>=2):

Input 1 for equal, 2 for unequal weighting of D variables ([1:2], <cr>=1):

Reading data file ...

Number of records in data file: 167

Length of longest entry in data file: 9

Checking for missing values ...

Total number of cases: 167

Column number	Categorical variable	No. of levels	No. of missing observations
58	married	2	0
59	educ	5	0
60	employ	2	0
63	race	2	0
64	csex	2	0

Re-checking data ...

Allocating missing value information

Assigning codes to categorical and missing values

Data checks complete

Creating missing value indicators

Rereading data

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

Total #cases	#cases w/ miss. D	#missing ord. vals	#X-var	#N-var	#F-var	#S-var
167	0	0	3	0	0	3
#M-var	#B-var	#C-var				

```

      0      0      5
No. cases used for training: 122
No. cases excluded due to 0 weight or missing D: 45
Finished reading data file
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): both.tex
Input 2 to save node IDs of observations, 1 otherwise ([1:2], <cr>=2):
Input name of file to store terminal node ID of each case: both.nid
Input 2 to save fitted values at each terminal node; 1 otherwise ([1:2], <cr>=1): 2
Input name of file to store node fitted values: both.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < both.in

```

5.12.2 Results

The tree has only two terminal nodes, splitting at `chlth` (Figure 22) and the text output is given below. Figure 23 plots the lowess-smoothed means of maternal stress and child illness in the terminal nodes. Note that the results here differ from those in Loh and Zheng (2013) partly because the latter used all the observations whereas the analysis here uses only observations with complete responses at all time points.

```

Multi-response or longitudinal data without T variables
Pruning by cross-validation
Data description file: mscmbboth.dsc
Training sample file: mscm.txt
Missing value code: NA
Records in data file start on line 1
Warning: N variables changed to S
Number of D variables = 56
Multivariate split variable selection method
Equal grouping of D variables with 8 groups
Segment boundaries are:
  7.500    14.50    21.50    28.50    35.50    42.50    49.50
Mean-squared errors (MSE) are calculated from unnormalized D variables
D variables equally weighted
Piecewise constant model
Number of records in data file: 167
Length of longest entry in data file: 9
Model fitted to subset of observations with complete D values
This says that subjects with incompletely observed responses are removed.
Summary information (without x variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),

```

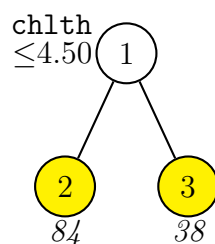


Figure 22: GUIDE v.29.6 0.50-SE pruned regression tree for predicting response variables `stress1`, `stress2`, `stress3`, `stress4`, `stress5`, `stress6`, `stress7`, `stress8`, `stress9`, `stress10`, `stress11`, `stress12`, `stress13`, `stress14`, `stress15`, `stress16`, `stress17`, `stress18`, `stress19`, `stress20`, `stress21`, `stress22`, `stress23`, `stress24`, `stress25`, `stress26`, `stress27`, `stress28`, `illness1`, `illness2`, `illness3`, `illness4`, `illness5`, `illness6`, `illness7`, `illness8`, `illness9`, `illness10`, `illness11`, `illness12`, `illness13`, `illness14`, `illness15`, `illness16`, `illness17`, `illness18`, `illness19`, `illness20`, `illness21`, `illness22`, `illness23`, `illness24`, `illness25`, `illness26`, `illness27`, and `illness28`. Maximum number of split levels is 10 and minimum node sample size is 10. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample sizes (*in italics*) printed below nodes. Second best split variable at root node is `csex`.

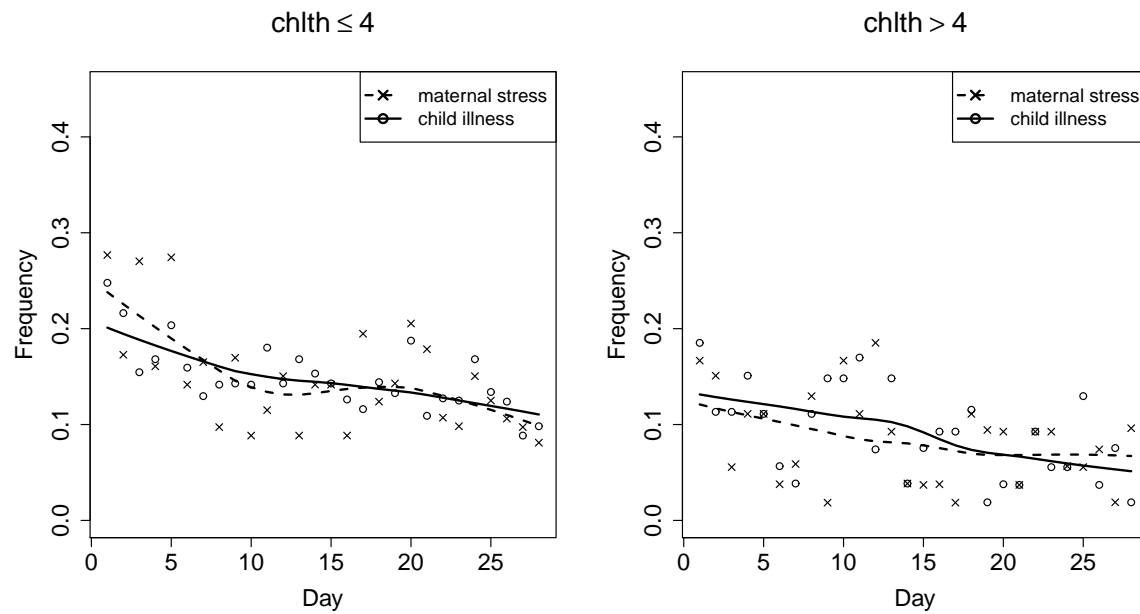


Figure 23: Lowess-smoothed mean curves in the terminal nodes of Figure 22.

s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
2	stress1	d	0.000	1.000		1
3	stress2	d	0.000	1.000		4
4	stress3	d	0.000	1.000		2
5	stress4	d	0.000	1.000		1
6	stress5	d	0.000	1.000		
7	stress6	d	0.000	1.000		1
8	stress7	d	0.000	1.000		7
9	stress8	d	0.000	1.000		
10	stress9	d	0.000	1.000		1
11	stress10	d	0.000	1.000		
12	stress11	d	0.000	1.000		
13	stress12	d	0.000	1.000		
14	stress13	d	0.000	1.000		
15	stress14	d	0.000	1.000		2
16	stress15	d	0.000	1.000		
17	stress16	d	0.000	1.000		1

18	stress17	d	0.000	1.000		
19	stress18	d	0.000	1.000		
20	stress19	d	0.000	1.000		2
21	stress20	d	0.000	1.000		1
22	stress21	d	0.000	1.000		1
23	stress22	d	0.000	1.000		1
24	stress23	d	0.000	1.000		1
25	stress24	d	0.000	1.000		1
26	stress25	d	0.000	1.000		1
27	stress26	d	0.000	1.000		
28	stress27	d	0.000	1.000		1
29	stress28	d	0.000	1.000		4
30	illness1	d	0.000	1.000		
31	illness2	d	0.000	1.000		3
32	illness3	d	0.000	1.000		4
33	illness4	d	0.000	1.000		1
34	illness5	d	0.000	1.000		
35	illness6	d	0.000	1.000		1
36	illness7	d	0.000	1.000		7
37	illness8	d	0.000	1.000		
38	illness9	d	0.000	1.000		1
39	illness10	d	0.000	1.000		
40	illness11	d	0.000	1.000		3
41	illness12	d	0.000	1.000		1
42	illness13	d	0.000	1.000		
43	illness14	d	0.000	1.000		4
44	illness15	d	0.000	1.000		2
45	illness16	d	0.000	1.000		2
46	illness17	d	0.000	1.000		1
47	illness18	d	0.000	1.000		4
48	illness19	d	0.000	1.000		1
49	illness20	d	0.000	1.000		2
50	illness21	d	0.000	1.000		3
51	illness22	d	0.000	1.000		3
52	illness23	d	0.000	1.000		1
53	illness24	d	0.000	1.000		
54	illness25	d	0.000	1.000		1
55	illness26	d	0.000	1.000		
56	illness27	d	0.000	1.000		1
57	illness28	d	0.000	1.000		2
58	married	c			2	
59	educ	c			5	
60	employ	c			2	
61	chlth	s	1.000	5.000		
62	mhlth	s	1.000	5.000		
63	race	c			2	

```

64 csex      c      2
65 housize   s      0.000      1.000

#cases w/ miss. D = number of cases with all D values missing
  Total #cases w/ #missing
  #cases miss. D ord. vals #X-var #N-var #F-var #S-var
    167      0      0      3      0      0      3
  #P-var #M-var #B-var #C-var #I-var
    0      0      0      5      0

No. cases used for training: 122
No. cases excluded due to 0 weight or missing D: 45

No nodewise interaction tests
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
Min. node sample size: 10
Number of SE's for pruned tree: 0.5000

Size and CV Loss and SE of subtrees:
Tree #Tnodes Mean Loss SE(Mean) BSE(Mean) Median Loss BSE(Median)
  1      7 1.240E-01 7.073E-03 9.041E-03 1.247E-01 8.921E-03
  2      6 1.235E-01 7.121E-03 8.995E-03 1.247E-01 8.873E-03
  3      5 1.224E-01 7.127E-03 8.878E-03 1.228E-01 8.790E-03
  4      4 1.225E-01 7.268E-03 8.904E-03 1.202E-01 1.120E-02
  5**     2 1.185E-01 7.195E-03 8.910E-03 1.202E-01 1.024E-02
  6++     1 1.278E-01 7.445E-03 9.045E-03 1.239E-01 8.634E-03

0-SE tree based on mean is marked with * and has 2 terminal nodes
0-SE tree based on median is marked with + and has 2 terminal nodes
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
* 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
1	122	122	1.099E-01	chlth
2T	84	84	1.236E-01	csex
3T	38	38	7.707E-02	married

Number of terminal nodes of final tree: 2

Total number of nodes of final tree: 3

Second best split variable (based on curvature test) at root node is csex

Regression tree for multi-response data:

Node 1: chlth <= 4.5000000

Node 2: Mean cost = 0.89217095E-001

Node 1: chlth > 4.5000000 or NA

Node 3: Mean cost = 0.54819328E-001

Node 1: Intermediate node

A case goes into Node 2 if chlth <= 4.5000000

chlth mean = 4.0737705

Means of stress1, stress2, stress3, stress4, stress5, stress6, stress7, stress8, stress9, stress10, stress11, stress12, stress13, stress14, stress15, stress16, stress17, stress18, stress19, stress20, stress21, stress22, stress23, stress24, stress25, stress26, stress27, stress28, illness1, illness2, illness3, illness4, illness5, illness6, illness7, illness8, illness9, illness10, illness11, illness12, illness13, illness14, illness15, illness16, illness17, illness18, illness19, illness20, illness21, illness22, illness23, illness24, illness25, illness26, illness27, and illness28

2.4590E-01	1.4754E-01	1.6393E-01	1.3934E-01	2.2131E-01
1.1475E-01	1.2295E-01	9.8361E-02	1.2295E-01	1.0656E-01
1.0656E-01	1.4754E-01	9.8361E-02	1.0656E-01	9.8361E-02
9.0164E-02	1.4754E-01	1.0656E-01	1.2295E-01	1.3115E-01
1.0656E-01	9.0164E-02	8.1967E-02	1.0656E-01	1.1475E-01
7.3770E-02	5.7377E-02	6.5574E-02	2.4590E-01	1.8033E-01
1.4754E-01	1.6393E-01	1.8033E-01	1.3934E-01	9.0164E-02
1.2295E-01	1.2295E-01	1.1475E-01	1.5574E-01	1.1475E-01
1.7213E-01	1.2295E-01	1.4754E-01	1.2295E-01	1.1475E-01
1.4754E-01	1.2295E-01	1.4754E-01	9.0164E-02	9.8361E-02
9.0164E-02	1.4754E-01	1.5574E-01	9.0164E-02	9.8361E-02
9.8361E-02				

Node 2: Terminal node

Means of stress1, stress2, stress3, stress4, stress5, stress6, stress7, stress8,

```

stress9, stress10, stress11, stress12, stress13, stress14, stress15, stress16,
stress17, stress18, stress19, stress20, stress21, stress22, stress23, stress24,
stress25, stress26, stress27, stress28, illness1, illness2, illness3, illness4,
illness5, illness6, illness7, illness8, illness9, illness10, illness11,
illness12, illness13, illness14, illness15, illness16, illness17, illness18,
illness19, illness20, illness21, illness22, illness23, illness24, illness25,
illness26, illness27, and illness28

```

```

2.6190E-01 1.5476E-01 2.1429E-01 1.4286E-01 2.6190E-01
1.4286E-01 1.6667E-01 9.5238E-02 1.6667E-01 9.5238E-02
1.1905E-01 1.1905E-01 8.3333E-02 1.3095E-01 1.3095E-01
1.0714E-01 2.0238E-01 1.1905E-01 1.4286E-01 1.5476E-01
1.3095E-01 9.5238E-02 8.3333E-02 1.3095E-01 1.3095E-01
8.3333E-02 7.1429E-02 5.9524E-02 2.7381E-01 2.0238E-01
1.6667E-01 1.7857E-01 2.1429E-01 1.9048E-01 1.1905E-01
1.3095E-01 1.0714E-01 1.0714E-01 1.6667E-01 1.4286E-01
1.9048E-01 1.6667E-01 1.7857E-01 1.3095E-01 1.1905E-01
1.5476E-01 1.6667E-01 2.0238E-01 1.1905E-01 1.1905E-01
1.0714E-01 1.7857E-01 1.5476E-01 1.1905E-01 1.0714E-01
1.3095E-01

```

```

-----
Node 3: Terminal node

```

```

Means of stress1, stress2, stress3, stress4, stress5, stress6, stress7, stress8,
stress9, stress10, stress11, stress12, stress13, stress14, stress15, stress16,
stress17, stress18, stress19, stress20, stress21, stress22, stress23, stress24,
stress25, stress26, stress27, stress28, illness1, illness2, illness3, illness4,
illness5, illness6, illness7, illness8, illness9, illness10, illness11,
illness12, illness13, illness14, illness15, illness16, illness17, illness18,
illness19, illness20, illness21, illness22, illness23, illness24, illness25,
illness26, illness27, and illness28

```

```

2.1053E-01 1.3158E-01 5.2632E-02 1.3158E-01 1.3158E-01
5.2632E-02 2.6316E-02 1.0526E-01 2.6316E-02 1.3158E-01
7.8947E-02 2.1053E-01 1.3158E-01 5.2632E-02 2.6316E-02
5.2632E-02 2.6316E-02 7.8947E-02 7.8947E-02 7.8947E-02
5.2632E-02 7.8947E-02 7.8947E-02 5.2632E-02 7.8947E-02
5.2632E-02 2.6316E-02 7.8947E-02 1.8421E-01 1.3158E-01
1.0526E-01 1.3158E-01 1.0526E-01 2.6316E-02 2.6316E-02
1.0526E-01 1.5789E-01 1.3158E-01 1.3158E-01 5.2632E-02
1.3158E-01 2.6316E-02 7.8947E-02 1.0526E-01 1.0526E-01
1.3158E-01 2.6316E-02 2.6316E-02 2.6316E-02 5.2632E-02
5.2632E-02 7.8947E-02 1.5789E-01 2.6316E-02 7.8947E-02
2.6316E-02

```

```

-----
Case and node IDs are in file: both.nid
Node fitted values are in file: both.fit
LaTeX code for tree is in both.tex

```


5.13 Subgroup identification: breast cancer

GUIDE has several methods to identify subgroups for differential treatment effects from randomized experiments. See [Loh et al. \(2015\)](#) and [Loh et al. \(2016\)](#) for details. The treatment variable is assumed to be categorical (i.e., it takes nominal values) and the response is an uncensored or censored event time (e.g., survival time). The key points are:

1. The treatment variable is designated as `R` (for “Rx”).
2. If there is no censoring in the response, it is designated as the dependent variable as `D` as usual.
3. If there is censoring in the response, the variable is designated as `T` (first letter of “Time”). In this case, the event indicator is designated as `D` (first letter of “Death”) and takes value 1 if the event (“death”) occurs and 0 if the event time is censored.

There are two types of covariate variables in subgroup identification. A *prognostic* variable is a clinical or biologic characteristic that provides information on the likely outcome of the disease in an untreated individual (e.g., patient age, family history, disease stage, and prior therapy). A *predictive* variable is a characteristic that provides information on the likely benefit from treatment. Predictive variables can be used to identify subgroups of patients who are most likely to benefit from a given therapy. Therefore prognostic variables define the effects of patient or tumor characteristics on the patient outcome, whereas predictive variables define the effect of treatment on the tumor ([Italiano, 2011](#)). Accordingly, GUIDE has two methods, called `Gi` and `Gs`. `Gi` is more sensitive to predictive variables and `Gs` tends to be equally sensitive to prognostic and predictive variables ([Loh et al., 2015](#)).

5.13.1 Without linear prognostic control

The simplest model only uses the covariates to split the intermediate nodes; terminal nodes are fitted with treatment means. We use a data set from a randomized controlled breast cancer trial ([Schmoor et al., 1996](#)) to show this. The data are in the file `cancerdata.txt`; it can also be obtained from the `TH.data` R package ([Hothorn, 2017](#)). In the description file `cancerdsc.txt` below, the treatment variable is hormone therapy, `horTh`. The variable `time` is (censored) time to recurrence of cancer and `event` = 1 if the cancer recurred and = 0 if it did not. Ordinal predictor variables

may be designated as “n” or “s” (with this option of no linear prognostic control, n variables will be automatically changed to s when the program is executed).

```
cancerdata.txt
NA
1
1 horTh r
2 age n
3 menostat c
4 tsize n
5 tgrade c
6 pnodes n
7 progrec n
8 estrec n
9 time t
10 event d
```

Input file generation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: nolin.in
Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):
Name of batch output file: nolin.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 4
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended if R variable is present,
  unless there are too many N, F or B variables)
Choose 2 for simple linear in one N or F variable + R (if present)
Choose 3 to fit a constant + R (if present)
1: multiple linear, 2: simple linear, 3: constant ([1:3], <cr>=3):
  Options 1 and 2 are for linear prognostic control
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
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 characters);
```

```

enclose with matching quotes if it has spaces: cancerdsc.txt
Reading data description file ...
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
Warning: N variables changed to S
Warning: model changed to linear in treatment
Warnings due to presence of R variable and choice of no linear prognostic effects
Dependent variable is death
Reading data file ...
Number of records in data file: 686
Length of longest entry in data file: 4
Checking for missing values ...
Total number of cases: 686

```

Column number	Categorical variable	No. of levels	No. of missing observations
1	horTh	2	0
3	menostat	2	0

```

Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to categorical and missing values
Data checks complete
Smallest uncensored T: 72.00
No. cases dropped due to missing D or T or censored T < smallest uncensored T: 14
No. complete cases excluding censored T < smallest uncensored T: 672
No. cases used to compute baseline hazard: 672
No. cases with D=1 and T >= smallest uncensored: 299
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 1
Choose a subgroup identification method:
1 = Sum of chi-squares (Gs)
2 = Treatment interactions (Gi)
Input your choice: ([1:2], <cr>=2):
Gi is the choice if splitting on predictive variables is preferred
Creating dummy variables
Rereading data

```

Total #cases	#cases w/ miss. D	#missing ord. vals	#X-var	#N-var	#F-var	#S-var
686	0	0	0	0	0	6
#P-var	#M-var	#B-var	#C-var	#I-var	#R-var	
0	0	0	1	0	1	

```

Survival time variable in column: 9
Event indicator variable in column: 10

```

```

Proportion uncensored among nonmissing T and D variables: .445
No. cases used for training: 672
Finished reading data file
Choose how you wish to deal with missing values in training or test data:
Option 1: Fit separate models to complete and incomplete cases
Option 2: Impute missing F and N values at each node with means for regression
Option 3: Fit a piecewise constant model
Input selection: ([1:3], <cr>=2):
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): 0.25
We choose 0.25 because the default value of 0.50 yields no splits
Choose fraction of cases for splitting
Larger values give more splits: 0 = median split and 1 = all possible splits
Default fraction is      1.0000
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):
Default min. cases per treatment in each node is 2
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 33
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input number of iterations ([1:100], <cr>=5):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): nolin.tex
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 color(s) for the terminal nodes:
(1) yellow-green
(2) red-green
(3) red-yellow
Input your choice ([1:3], <cr>=1):
Input 1 to print treatment effects in tree, 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 truncation limits and regression coefficients in a file, 1 otherwise ([1:2], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: nolin.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: nolin.r
Input file is created!

```

Run GUIDE with the command: `guide < nolin.in`

Results The contents of `nolin.out` follow.

```
Proportional hazards regression with relative risk estimates
Pruning by cross-validation
Data description file: cancerdsc.txt
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
Warning: N variables changed to S
Warning: model changed to linear in treatment
Dependent variable is death
Piecewise linear model
Number of records in data file: 686
Length of longest entry in data file: 4
Treatment (R) variable is horTh with values "no" and "yes"
Smallest uncensored T: 72.00
No. cases dropped due to missing D or T or censored T < smallest uncensored T: 14
No. complete cases excluding censored T < smallest uncensored T: 672
No. cases used to compute baseline hazard: 672
No. cases with D=1 and T >= smallest uncensored: 299
Number of dummy variables created: 1
```

Summary information (without x variables)

d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
t=survival time variable

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	horTh	r			2	
2	age	s	21.00	80.00		
3	menostat	c			2	
4	tsize	s	3.000	120.0		
5	tgrade	s	1.000	3.000		
6	pnodes	s	1.000	51.00		
7	progrec	s	0.000	2380.		
8	estrec	s	0.000	1144.		
9	time	t	72.00	2659.		
10	death	d	0.000	1.000		

===== Constructed variables =====

```

11 lnbasehaz0 z -6.510 0.5887E-01
12 horTh.yes f 0.000 1.000

```

```

Total #cases w/ #missing
#cases miss. D ord. vals #X-var #N-var #F-var #S-var
686 0 0 0 0 0 0 6
#P-var #M-var #B-var #C-var #I-var #R-var
0 0 0 1 0 1

```

Survival time variable in column: 9

Event indicator variable in column: 10

Proportion uncensored among nonmissing T and D variables: 0.445

No. cases used for training: 672

Missing values imputed with node means for regression

Treatment interactions (Gi)

No nodewise interaction tests

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: 1.0000

Max. number of split levels: 10

Min. number of cases per treatment at each node: 2

Min. node sample size: 33

Number of iterations: 5

Number of SE's for pruned tree: 0.2500

Size and CV Loss and SE of subtrees:

Tree	#Tnodes	Mean Loss	SE(Mean)	BSE(Mean)	Median Loss	BSE(Median)
1	14	1.463E+00	5.754E-02	4.290E-02	1.456E+00	4.773E-02
2	13	1.463E+00	5.754E-02	4.290E-02	1.456E+00	4.773E-02
3	12	1.466E+00	5.775E-02	4.345E-02	1.456E+00	4.774E-02
4	11	1.468E+00	5.760E-02	4.225E-02	1.458E+00	4.669E-02
5	9	1.472E+00	5.750E-02	4.188E-02	1.458E+00	4.328E-02
6	8	1.475E+00	5.757E-02	4.176E-02	1.468E+00	4.206E-02
7	7	1.424E+00	5.367E-02	3.621E-02	1.440E+00	4.073E-02
8	4	1.419E+00	5.246E-02	3.985E-02	1.451E+00	5.973E-02
9**	2	1.400E+00	5.025E-02	2.743E-02	1.393E+00	3.702E-02
10	1	1.418E+00	5.029E-02	1.618E-02	1.417E+00	1.961E-02

0-SE tree based on mean is marked with * and has 2 terminal nodes

0-SE tree based on median is marked with + and has 2 terminal nodes

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.

Rel. risk is mean risk relative to sample average ignoring covariates

Cases fit give the number of cases used to fit node

Deviance is mean residual deviance for all cases in node

Node	Total	Cases	Matrix	Node	Node	Split
label	cases	fit	rank	rel.risk	deviance	variable
1	672	672	1	1.000E+00	1.418E+00	progrec
2T	274	274	1	1.580E+00	1.584E+00	estrec
3T	398	398	1	7.112E-01	1.179E+00	menostat

Number of terminal nodes of final tree: 2

Total number of nodes of final tree: 3

Second best split variable (based on curvature test) at root node is estrec

Regression tree:

Node 1: progrec <= 21.500000

Node 2: Risk relative to sample average ignoring covariates = 1.5803895

Node 1: progrec > 21.500000 or NA

Node 3: Risk relative to sample average ignoring covariates = 0.71121830

Constant term for constant hazard model (ignoring covariates): -0.34747300E-002

WARNING: p-values below are not adjusted for split search. For a bootstrap solution, see Loh et al. (2016), "Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

Node 1: Intermediate node

A case goes into Node 2 if progrec <= 21.500000

progrec mean = 110.91518

Coefficients of log-relative risk function:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	0.12581					
horTh.yes	-0.36394	-2.9217	0.35980E-02	0.0000	0.36012	1.0000

Node 2: Terminal node

Coefficients of log-relative risk function:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	0.49659					

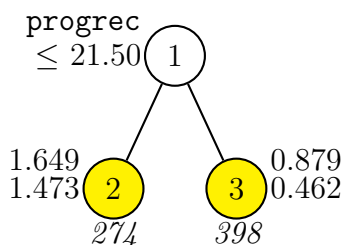


Figure 24: GUIDE v.29.6 0.25-SE pruned Gi proportional hazards regression tree for differential treatment effects for **time** without linear prognostic effects. Maximum number of split levels is 10, minimum node sample size is 33 and minimum number per treatment level is 2. At each split, an observation goes to the left branch if and only if the condition is satisfied. Numbers beside terminal nodes are estimated relative risks (relative to average for sample ignoring covariates) corresponding to **horTh** levels **no** and **yes**, respectively; numbers (*in italics*) below are sample sizes. Nodes with negative and positive effects for **horTh** level **yes** are colored yellow and green, respectively. Second best split variable at root node is **estrec**.

```

horTh.yes  -0.11268      -0.67893      0.49776      0.0000      0.36131      1.0000
-----
Node 3: Terminal node
Coefficients of log-relative risk function:
Regressor   Coefficient  t-stat      p-value      Minimum      Mean      Maximum
Constant    -0.13274
horTh.yes   -0.64394      -3.3682     0.83065E-03  0.0000      0.35930    1.0000
-----

Observed and fitted values are stored in nolin.fit
LaTeX code for tree is in nolin.tex
R code is stored in nolin.r

```

Let $\lambda(u, \mathbf{x})$ denote the hazard function at time u and predictor values \mathbf{x} and let $\lambda_0(u)$ denote the baseline hazard function. The results in **nolin.out** show that the fitted proportional hazards model is

$$\begin{aligned} \lambda(u, \mathbf{x}) = & \lambda_0(u) [\exp\{\hat{\beta}_1 + \hat{\gamma}_1 I(\text{horTh} = \text{yes})\} I(\text{progrec} \leq 21.5) \\ & + \exp\{\hat{\beta}_2 + \hat{\gamma}_2 I(\text{horTh} = \text{yes})\} I(\text{progrec} > 21.5)] \end{aligned}$$

with $\hat{\beta}_1 = 0.49659$, $\hat{\gamma}_1 = -0.11268$, $\hat{\beta}_2 = -0.13274$, and $\hat{\gamma}_2 = -0.64394$.

Figure 24 shows the L^AT_EX tree diagram. The numbers beside each terminal node are relative risks (relative to the average risk of the entire sample) defined

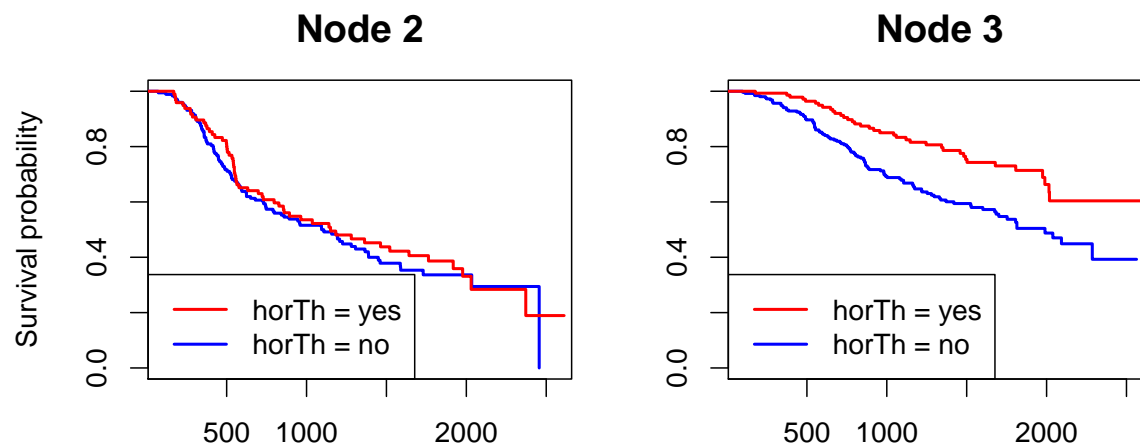


Figure 25: Estimated survival probability functions for breast cancer data

as $\exp\{\hat{\beta} + \hat{\gamma}I(\text{horTh} = \text{yes}) - \hat{\beta}_*\}$, where $\hat{\beta}_* = -0.00347473$ (printed in the line below the row of asterisks in the results) is the estimated regression coefficient for the constant model $\lambda(u, \mathbf{x}) = \lambda_0(u) \exp(\beta_*)$ fitted to the entire sample (see the text in Section 5.9 on page 193). For example, the relative risks for `horTh = no` and `yes` in the left terminal node of the tree are

$$\begin{aligned} \exp(0.49659 + 0.00347473) &= 1.648828 \\ \exp(0.49659 - 0.11268 + 0.00347473) &= 1.473123 \end{aligned}$$

respectively. The Kaplan-Meier survival functions estimated from the data in the terminal nodes of the tree are shown in Figure 25. The plots are produced by the following R code.

```
library(survival)
z <- read.table("cancer.dat",header=FALSE)
names(z) <- c("horTh","age","menostat","tsize","tgrade","pnodes","progrec",
              "estrec","time","death")
leg.txt <- c("horTh = yes","horTh = no")
leg.col <- c("red","blue")
leg.lty <- 1:2
xr <- range(z$time)
zg <- read.table("nolin.fit",header=TRUE)
nodes <- zg$node
uniq.gp <- unique(sort(nodes))
plotted <- FALSE
for(g in uniq.gp){
  gp <- nodes == g
```

```

y <- z$time[gp]
stat <- z$death[gp]
treat <- z$horTh[gp]
fit <- survfit(Surv(y,stat) ~ treat, conf.type="none")
if(plotted){
  plot(fit,xlim=xr,mark.time=FALSE,xlab="",ylab="",col=c("blue","red"),lwd=2)
} else {
  plot(fit,xlim=xr,mark.time=FALSE,xlab="",ylab="Survival probability",
       col=c("blue","red"),lwd=2)
  plotted <- TRUE
}
title(paste("Node",g))
legend("bottomleft",legend=leg.txt,lty=1,col=leg.col,lwd=2)
}

```

Estimated relative risks and survival probabilities The file `nolin.fit` gives the terminal node number, estimated survival time, log baseline cumulative hazard, relative risk (relative to the average for the data, ignoring covariates), survival probability, median survival time, and treatment effect (regression coefficient of treatment indicator) of each observation in the training sample (`cancerdata.txt`). The results for the first few observations are shown below. A plus (+) sign at the end of a value in the last column indicates that the observed survival time is censored. See Section 5.9 for definitions of the terms.

train	node	survivaltime	logbasecumhaz	relativerisk	survivalprob	mediansurvtime	horTh.yes
y	3	1.814000E+03	-3.375512E-01	8.787380E-01	5.353587E-01	2.014989E+03	-6.439363E-01
y	3	2.018000E+03	-2.141505E-01	4.615316E-01	6.898588E-01	2.659000E+03+	-6.439363E-01
y	3	7.120000E+02	-1.272907E+00	4.615316E-01	8.791602E-01	2.659000E+03+	-6.439363E-01
y	3	1.807000E+03	-3.603344E-01	4.615316E-01	7.255862E-01	2.659000E+03+	-6.439363E-01
y	3	7.720000E+02	-1.153244E+00	8.787380E-01	7.585265E-01	2.014989E+03	-6.439363E-01
y	2	4.480000E+02	-2.056938E+00	1.648824E+00	8.105337E-01	1.027322E+03	-1.126776E-01

5.13.2 Simple linear prognostic control

To reduce or eliminate confounding between treatment and covariate variables, it may be desirable to adjust for the effects of the latter by fitting a regression model that allows for the linear effects of one or more prognostic variables in each node (Loh et al., 2019b). This is done by choosing the “simple linear” or the “multiple linear” option and specifying each potential linear predictor as “n” in the description file (no change is needed in `cancerdsc.txt`). First we show how to choose the simple linear option, where a single prognostic variable is used in each node.

Input file generation

```

0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: lin.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: lin.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
    1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
    5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 4
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended if R variable is present,
    unless there are too many N, F or B variables)
Choose 2 for simple linear in one N or F variable + R (if present)
Choose 3 to fit a constant + R (if present)
1: multiple linear, 2: simple linear, 3: constant ([1:3], <cr>=3): 2
    Option 2 fits one prognostic variable in each node.
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: cancerdsc.txt
Reading data description file ...
Training sample file: cancer.txt
Missing value code: NA
Records in data file start on line 1
R variable present
Dependent variable is death
Reading data file ...
Number of records in data file: 686
Length of longest entry in data file: 4
Checking for missing values ...
Total number of cases: 686
Col. no. Categorical variable    #levels    #missing values
      1 horTh                    2                0
      3 menostat                 2                0
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to categorical and missing values
Data checks complete
Smallest uncensored T: 72.00
No. cases dropped due to missing D or T or censored T < smallest uncensored T: 14

```

```

No. complete cases excluding censored T < smallest uncensored T: 672
No. cases used to compute baseline hazard: 672
No. cases with D=1 and T >= smallest uncensored: 299
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 1
Choose a subgroup identification method:
1 = Sum of chi-squares (Gs)
2 = Treatment interactions (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables
Rereading data
      Total #cases w/   #missing
      #cases  miss. D  ord. vals  #X-var  #N-var  #F-var  #S-var  #B-var  #C-var  #R-var
        686         0         0        0        6        0        0        0        1        1
Survival time variable in column: 9
Event indicator variable in column: 10
Proportion uncensored among nonmissing T and D variables: .445
No. cases used for training: 672
Finished reading data file
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): lin.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: lin.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < lin.in

```

Results The results in the following output file `lin.out` show that there are no splits. The best linear predictor at the root node is the prognostic variable `pnodes`.

```

Proportional hazards regression with relative risk estimates
No truncation of predicted values
Pruning by cross-validation
Data description file: cancerdsc.txt
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
Dependent variable is death
Piecewise simple linear or constant model
Powers are dropped if they are not significant at level 1.0000
Number of records in data file: 686
Length of longest entry in data file: 4
Treatment (R) variable is horTh with values "no" and "yes"

```

Smallest uncensored T: 72.00
 No. cases dropped due to missing D or T or censored T < smallest uncensored T: 14
 No. complete cases excluding censored T < smallest uncensored T: 672
 No. cases used to compute baseline hazard: 672
 No. cases with D=1 and T >= smallest uncensored: 299
 Number of dummy variables created: 1

Summary information (without x variables)

d=dependent, b=split and fit cat variable using indicator variables,
 c=split-only categorical, i=fit-only categorical (via indicators),
 s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
 m=missing-value flag variable, p=periodic variable, w=weight
 t=survival time variable

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	horTh	r			2	
2	age	n	21.00	80.00		
3	menostat	c			2	
4	tsize	n	3.000	120.0		
5	tgrade	n	1.000	3.000		
6	pnodes	n	1.000	51.00		
7	progrec	n	0.000	2380.		
8	estrec	n	0.000	1144.		
9	time	t	72.00	2659.		
10	death	d	0.000	1.000		
===== Constructed variables =====						
11	lnbasehaz0	z	-6.510	0.5887E-01		
12	horTh.yes	f	0.000	1.000		

Total #cases	#cases w/ miss. D	#missing ord. vals	#X-var	#N-var	#F-var	#S-var
686	0	0	0	6	0	0
#P-var	#M-var	#B-var	#C-var	#I-var	#R-var	
0	0	0	1	0	1	

Survival time variable in column: 9

Event indicator variable in column: 10

Proportion uncensored among nonmissing T and D variables: 0.445

No. cases used for training: 672

Missing values imputed with node means for regression

Treatment interactions (Gi)

No nodewise interaction tests

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: 1.0000

Max. number of split levels: 10
 Min. number of cases per treatment at each node: 2
 Min. node sample size: 33
 Number of iterations: 5
 Number of SE's for pruned tree: 0.5000

Size and CV Loss and SE of subtrees:

Tree	#Tnodes	Mean Loss	SE(Mean)	BSE(Mean)	Median Loss	BSE(Median)
1	15	1.917E+00	1.911E-01	1.776E-01	1.697E+00	1.512E-01
2	14	1.918E+00	1.907E-01	1.774E-01	1.687E+00	1.510E-01
3	13	1.871E+00	1.564E-01	1.409E-01	1.687E+00	1.485E-01
4	12	1.863E+00	1.560E-01	1.404E-01	1.687E+00	1.390E-01
5	8	1.863E+00	1.560E-01	1.404E-01	1.687E+00	1.390E-01
6	6	1.843E+00	1.552E-01	1.444E-01	1.687E+00	1.622E-01
7	5	1.503E+00	7.236E-02	5.935E-02	1.475E+00	4.206E-02
8	2	1.459E+00	8.092E-02	8.464E-02	1.388E+00	4.183E-02
9**	1	1.393E+00	5.657E-02	4.279E-02	1.343E+00	3.223E-02

0-SE tree based on mean is marked with * and has 1 terminal node
 0-SE tree based on median is marked with + and has 1 terminal node
 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.

Rel. risk is mean risk relative to sample average ignoring covariates

Cases fit give the number of cases used to fit node

Deviance is mean residual deviance for all cases in node

Node	Total	Cases	Matrix	Node	Node	Split
label	cases	fit	rank	rel.risk	deviance	variable
1T	672	672	3	1.000E+00	1.381E+00	estrec

Best split at root node is on estrec at 4.5000

Number of terminal nodes of final tree: 1

Total number of nodes of final tree: 1

Second best split variable (based on curvature test) at root node is progrec

Regression tree:

Node 1: Risk relative to sample average ignoring covariates = 1.0000000

Constant term for constant hazard model (ignoring covariates): -0.46227816E-002

Node 1: Terminal node

Coefficients of log-relative risk function:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	-0.16815					
pnodes	0.57675E-01	8.8166	0.0000	1.0000	4.9866	51.000
horTh.yes	-0.35694	-2.8608	0.43577E-02	0.0000	0.36012	1.0000

Observed and fitted values are stored in lin.fit

LaTeX code for tree is in lin.tex

5.13.3 Multiple linear prognostic control

Now we show how to use all n variables as linear predictors in each node.

Input file generation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: mul.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: mul.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
    1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
    5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 4
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended if R variable is present,
    unless there are too many N, F or B variables)
Choose 2 for simple linear in one N or F variable + R (if present)
Choose 3 to fit a constant + R (if present)
1: multiple linear, 2: simple linear, 3: constant ([1:3], <cr>=3): 1
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
```

Option 1 fits all n and f variables in each node.

```

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: cancerdsc.txt
Reading data description file ...
Training sample file: cancer.txt
Missing value code: NA
Records in data file start on line 1
R variable present
Dependent variable is death
Reading data file ...
Number of records in data file: 686
Length of longest entry in data file: 4
Checking for missing values ...
Total number of cases: 686
Col. no. Categorical variable      #levels      #missing values
      1 horTh                      2              0
      3 menostat                  2              0
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to categorical and missing values
Data checks complete
Smallest uncensored T: 72.00
No. cases dropped due to missing D or T or censored T < smallest uncensored T: 14
No. complete cases excluding censored T < smallest uncensored T: 672
No. cases used to compute baseline hazard: 672
No. cases with D=1 and T >= smallest uncensored: 299
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 1
Choose a subgroup identification method:
1 = Sum of chi-squares (Gs)
2 = Treatment interactions (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables
Rereading data
      Total #cases w/      #missing
      #cases  miss. D  ord. vals  #X-var  #N-var  #F-var  #S-var  #B-var  #C-var  #R-var
      686      0      0      0      6      0      0      0      1      1

Survival time variable in column: 9
Event indicator variable in column: 10
Proportion uncensored among nonmissing T and D variables: .445
No. cases used for training: 672
Finished reading data file
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): mul.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):

```



```

Input name of file to store node ID and fitted value of each case: mul.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < mul.in

```

Results

```

Proportional hazards regression with relative risk estimates
No truncation of predicted values
Pruning by cross-validation
Data description file: cancerdsc.txt
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
Dependent variable is death
Piecewise linear model
Number of records in data file: 686
Length of longest entry in data file: 4
Treatment (R) variable is horTh with values "no" and "yes"
Smallest uncensored T: 72.00
No. cases dropped due to missing D or T or censored T < smallest uncensored T: 14
No. complete cases excluding censored T < smallest uncensored T: 672
No. cases used to compute baseline hazard: 672
No. cases with D=1 and T >= smallest uncensored: 299
Number of dummy variables created: 1

```

```

Summary information (without x variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
t=survival time variable

```

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	horTh	r			2	
2	age	n	21.00	80.00		
3	menostat	c			2	
4	tsize	n	3.000	120.0		
5	tgrade	n	1.000	3.000		
6	pnodes	n	1.000	51.00		
7	progrec	n	0.000	2380.		
8	estrec	n	0.000	1144.		
9	time	t	72.00	2659.		

```

10  death      d      0.000      1.000
===== Constructed variables =====
11  lnbasehaz0 z     -6.510      0.5887E-01
12  horTh.yes  f      0.000      1.000

Total  #cases w/  #missing
#cases  miss. D  ord. vals  #X-var  #N-var  #F-var  #S-var
   686      0      0      0      6      0      0
#P-var  #M-var  #B-var  #C-var  #I-var  #R-var
   0      0      0      1      0      1

Survival time variable in column: 9
Event indicator variable in column: 10
Proportion uncensored among nonmissing T and D variables: 0.445
No. cases used for training: 672

Missing values imputed with node means for regression
Treatment interactions (Gi)
No nodewise interaction tests
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: 1.0000
Max. number of split levels: 10
Min. number of cases per treatment at each node: 2
Min. node sample size: 33
Number of iterations: 5
Number of SE's for pruned tree: 0.000

Size and CV Loss and SE of subtrees:
Tree  #Tnodes  Mean Loss  SE(Mean)  BSE(Mean)  Median Loss  BSE(Median)
  1      13  5.613E+01  5.408E+01  4.980E+01  2.036E+00  1.215E+01
  2      12  5.609E+01  5.408E+01  4.981E+01  1.940E+00  1.215E+01
  3      11  5.609E+01  5.408E+01  4.981E+01  1.940E+00  1.215E+01
  4      10  5.607E+01  5.408E+01  4.981E+01  1.940E+00  1.215E+01
  5       9  5.606E+01  5.408E+01  4.981E+01  1.961E+00  1.215E+01
  6       7  5.599E+01  5.408E+01  4.983E+01  1.728E+00  1.216E+01
  7       6  5.581E+01  5.408E+01  4.984E+01  1.574E+00  1.216E+01
  8       5  1.584E+00  9.455E-02  1.310E-01  1.430E+00  9.644E-02
 9**      1  1.361E+00  5.511E-02  4.160E-02  1.313E+00  3.862E-02

0-SE tree based on mean is marked with * and has 1 terminal node
0-SE tree based on median is marked with + and has 1 terminal node
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.

Rel. risk is mean risk relative to sample average ignoring covariates

Cases fit give the number of cases used to fit node

Deviance is mean residual deviance for all cases in node

Node	Total	Cases	Matrix	Node	Node	Split
label	cases	fit	rank	rel.risk	deviance	variable
1T	672	672	7	1.000E+00	1.349E+00	estrec

Best split at root node is on estrec at 1.5000

Number of terminal nodes of final tree: 1

Total number of nodes of final tree: 1

Second best split variable (based on curvature test) at root node is progrec

Regression tree:

Node 1: Risk relative to sample average ignoring covariates = 1.0000000

Constant term for constant hazard model (ignoring covariates): -0.10204012E-001

Node 1: Terminal node

Coefficients of log-relative risk function:

Regressor	Coefficient	t-stat	p-value	Minimum	Mean	Maximum
Constant	-0.80565					
age	0.61317E-03	0.98057E-01	0.92192	21.000	53.077	80.000
tsize	0.74417E-02	1.8997	0.57904E-01	3.0000	29.317	120.00
tgrade	0.28302	2.6819	0.75031E-02	1.0000	2.1161	3.0000
pnodes	0.50016E-01	6.8377	0.0000	1.0000	4.9866	51.000
progrec	-0.23066E-02	-4.0009	0.70195E-04	0.0000	110.92	2380.0
estrec	0.17315E-03	0.39048	0.69631	0.0000	97.475	1144.0
horTh.yes	-0.32088	-2.5012	0.12618E-01	0.0000	0.36012	1.0000

Observed and fitted values are stored in mul.fit

LaTeX code for tree is in mul.tex

Table 4: Codes in M variables of CE data

A	valid nonresponse: a response is not anticipated
B	invalid nonresponse
C	“don’t know”, refusal, or other type of nonresponse
D	valid data value
T	topcoding applied to value

6 Multiple missing value codes: Consumer expenditure survey

The data in the examples thus far contain at most one missing value code, but GUIDE allows more than one missing value code. It does this using the format of the Bureau of Labor Statistics Consumer Expenditure (CE) Survey, where each variable with missing values has an associated “flag” variable that gives the reason for the missingness. Table 4 lists the missing value codes for the CE Survey; see [Loh et al. \(2019a\)](#) for more information about the data.

The example data and description files `cedata.txt` and `ceclass.dsc`, respectively, show a typical setup. Flag variables are indicated by the letter `m` or `M` in the description file. Further, each `M` variable should follow immediately behind the `N` or `S` variable with which it is associated. For example, the `M` variable `AGE2_` which is associated with the `N` variable `AGE2`, is listed immediately after the latter in the file `ceclass.dsc` whose top few lines are shown below. GUIDE exits with an error if it detects an `M` variable immediately following any variable that is not `N` or `S`.

```
cedata.txt
NA
2
1 DIRACC C
2 DIRACC_ X
3 AGE_REF N
4 AGE_REF_ X
5 AGE2 N
6 AGE2_ M
:
515 INTRDVX X
516 INTRDVX_ D
:
```

The top few lines of the data file `cedata.txt` (below) show that the first respondent has an `AGE2` = 87 and `AGE2_` = T, and the second respondent has `AGE2` = NA and `AGE2_` = A.

```
"DIRACC" "DIRACC_" "AGE_REF" "AGE_REF_" "AGE2" "AGE2_" ...
```

```
"1" "D" 82 "D" 87 "T" ...
"1" "D" 69 "D" NA "A" ...
"1" "D" 45 "D" 43 "D" ...
"1" "D" 53 "D" 59 "D" ...
"1" "D" 46 "D" NA "A" ...
```

There can be up to 31 different codes in each **M** variable, and the codes may be *different* from one **M** variable to another. Codes in **M** variables that refer to non-missing values in the associated **N** or **S** variables (such as **D** and **T** here) are ignored and do not count towards the 31 number limit. A missing value code (**NA** in this example) is always required in the second line of the description file to indicate which values in the **N** and **S** variables are missing.

M variables are not allowed for categorical (**C** or **B**) variables. Missing value codes in these variables should be included among their categorical values. An example is the variable **DIRACC_** which is a flag variable for the **C** variable **DIRACC** in the current data. **DIRACC_** is assigned **X** in the description file because **NA** values in **DIRACC** are replaced with the missing value codes in **DIRACC_**.

6.1 Classification

This section shows the construction of a classification tree for predicting the value of **INTRDVX_**, which takes value **C** or **D**, depending on whether **INTRDVX** is missing or non-missing. A similar procedure applies for regression trees.

6.1.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: class.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: class.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: cecclass.dsc
Reading data description file ...
Training sample file: cedata.txt
```

```

Missing value code: NA
Records in data file start on line 2
Warning: N variables changed to S
Dependent variable is INTRDVX_
Reading data file ...
Number of records in data file: 4609
Length of longest entry in data file: 11
Checking for missing values ...
Total number of cases: 4609
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Number of classes: 2

```

Column number	Categorical variable	No. of levels	No. of missing observations
1	DIRACC	2	154
22	BUILDING	10	0
24	CUTENURE	5	0
26	EARNCOMP	8	0
34	FAM_TYPE	9	0
37	FAMT_EDX	2	0
45	FINCAT_X	2	0
47	FINCBT_X	2	0
49	FIND_ETX	2	0
52	FJSS_EDX	2	0
54	FPRI_ENX	2	0
56	FRRDEX_	2	0
60	FSAL_RYX	2	0
62	FSLTAXX_	2	0
73	INCNONW1	6	2912
74	INCN_NW1	2	0
75	INCNONW2	6	3656
76	INCN_NW2	2	0
77	INCOMEY1	6	1697
78	INCO_EY1	2	0
79	INCOMEY2	6	2832
80	INCO_EY2	2	0
89	MARITAL1	5	0
94	NONI_CMV	2	0
97	OCCUCOD1	15	1697
98	OCCU_OD1	2	0
99	OCCUCOD2	15	2832
100	OCCU_OD2	2	0
108	PRINEARN	7	0
110	QINTRVMO	12	0
111	QINTRVYR	2	0
112	RACE2	6	1879

113	RACE2_	2	0
114	REF_RACE	6	0
116	REGION	4	33
119	RESPSTAT	2	0
123	SEX_REF	2	0
125	SEX2	2	1879
126	SEX2_	2	0
131	SMSASTAT	2	0
132	ST_HOUS	2	0
135	TOTT_PDX	2	0
304	HHID	46	4531
305	HHID_	2	0
306	POV_CY	2	378
307	POV_CY_	2	0
308	POV_PY	2	378
309	POV_PY_	2	0
310	SWIMPOOL	1	4045
311	SWIM_OOL	3	0
312	APTMENT	1	4535
313	APTMENT_	3	0
314	OFSTPARK	1	1160
315	OFST_ARK	3	0
316	WINDOWAC	1	3977
317	WIND_WAC	3	0
318	CNTRALAC	1	1459
319	CNTR_LAC	3	0
320	CHILDAGE	8	0
323	STATE	39	486
410	PORCH	1	997
411	PORCH_	3	0
455	WELF_EBX	2	0
467	HORREF1	6	4448
468	HORREF1_	2	0
469	HORREF2	5	4495
470	HORREF2_	2	0
474	FGOV_ETM	2	0
476	FPRI_ENM	2	0
478	FRRDEDM_	2	0
479	PSU	21	2579
480	HISP_REF	2	0
481	HISP2	2	1879
596	RETS_RVB	3	0
601	RETSURV	2	0
604	RETSURVI	3	0

Column Missing-value No. of No. of missing

number	flag variable	codes	observations
6	AGE2_	1	0
18	BATHRMQ_	2	0
20	BEDR_OMQ	2	0
39	FEDR_NDX	2	0
41	FEDTAXX_	2	0
66	HLFB_THQ	2	0
68	INC__RS1	1	0
70	INC__RS2	1	0
72	INC__ANK	1	0
84	INCW_EK2	1	0
86	MISC_AXX	3	0
88	LUMP_UMX	2	0
102	OTHR_NCX	2	0
118	RENT_QVX	1	0
122	ROOMSQ_	2	0
128	SLOC_AXX	2	0
130	SLRF_NDX	2	0
139	WELF_REX	2	0
325	ERANKH_	1	0
457	LUMP_UMB	2	0
459	LMPS_MBX	2	0
461	OTHR_NCB	2	0
484	BUILT_	2	0
486	CRED_INX	2	0
488	CREDITB_	2	0
490	CRED_TBX	2	0
492	CREDITX_	2	0
494	CRED_YRX	2	0
496	CREDYRB_	2	0
498	CRED_RBX	2	0
500	DEFB_NRP	2	0
502	EITC_	2	0
505	FMLP_YRX	2	0
507	FS_MTHI_	1	0
525	IRAB_	2	0
527	IRABX_	2	0
529	IRAX_	2	0
531	IRAYRB_	2	0
533	IRAYRBX_	2	0
535	IRAYRX_	2	0
539	LIQD_RBX	2	0
541	LIQU_DBX	2	0
543	LIQU_YRB	2	0
545	LIQU_YRX	2	0
547	LIQUIDB_	2	0

549	LIQUIDX_	2	0
551	MEAL_PAY	2	0
553	MLPA_WKX	2	0
555	MLPY_WKS	2	0
557	NETR_NTB	3	0
559	NETR_NTX	2	0
561	NETR_TBX	2	0
563	OTHA_TBX	2	0
565	OTHASTB_	3	0
567	OTHASTX_	2	0
569	OTHFINX_	2	0
571	OTHL_NBX	2	0
573	OTHL_RBX	2	0
575	OTHL_YRB	2	0
577	OTHL_YRX	2	0
579	OTHLOAN_	1	0
581	OTHLONB_	2	0
583	OTHLONX_	2	0
585	OTHR_GBX	2	0
587	OTHREGB_	2	0
589	OTHREGX_	2	0
591	OTHS_RBX	2	0
593	OTHS_YRB	2	0
595	OTHS_YRX	2	0
598	RETS_RVX	2	0
600	RETS_VBX	2	0
607	ROYE_TBX	2	0
609	ROYESTB_	2	0
611	ROYESTX_	2	0
613	STCK_RBX	2	0
615	STDN_YRB	2	0
617	STDN_YRX	2	0
619	STDT_RBX	2	0
621	STOC_YRB	2	0
623	STOC_YRX	2	0
625	STOCKB_	3	0
627	STOCKBX_	2	0
629	STOCKX_	2	0
631	STUD_INX	2	0
633	STUD_TBX	2	0
635	STUDNTB_	2	0
637	STUDNTX_	2	0
639	WHLF_RBX	2	0
641	WHLFYRB_	2	0
643	WHLFYRX_	3	0
645	WHOL_FBX	2	0

```

        647 WHOLIFB_          3          0
        649 WHOLIFX_          3          0
Re-checking data ...
Allocating missing value information
Assigning codes to categorical and missing values
Data checks complete
Creating missing value indicators
Rereading data
Class #Cases    Proportion
C      1771      0.38424821
D      2838      0.61575179
  Total #cases w/ #missing
#cases  miss. D ord. vals #X-var #N-var #F-var #S-var
  4609      0    4609     71      0      0     412
#M-var #B-var #C-var
   93      0      76
No. cases used for training: 4609
No. cases excluded due to 0 weight or missing D: 0
Finished reading data file
Warning: No interaction tests; too many predictor variables
Choose 1 for estimated priors, 2 for equal priors, 3 for 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):
Input 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):
Input file name to store LaTeX code (use .tex as suffix): class.tex
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2): 1
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < class.in

```

6.1.2 Results

The contents of the output file are given below and the \LaTeX tree in Figure 26. Instead of all missing values going to the left or to the right at each split, the splits on missing value flag variables (indicated in red) are on particular missing value codes.

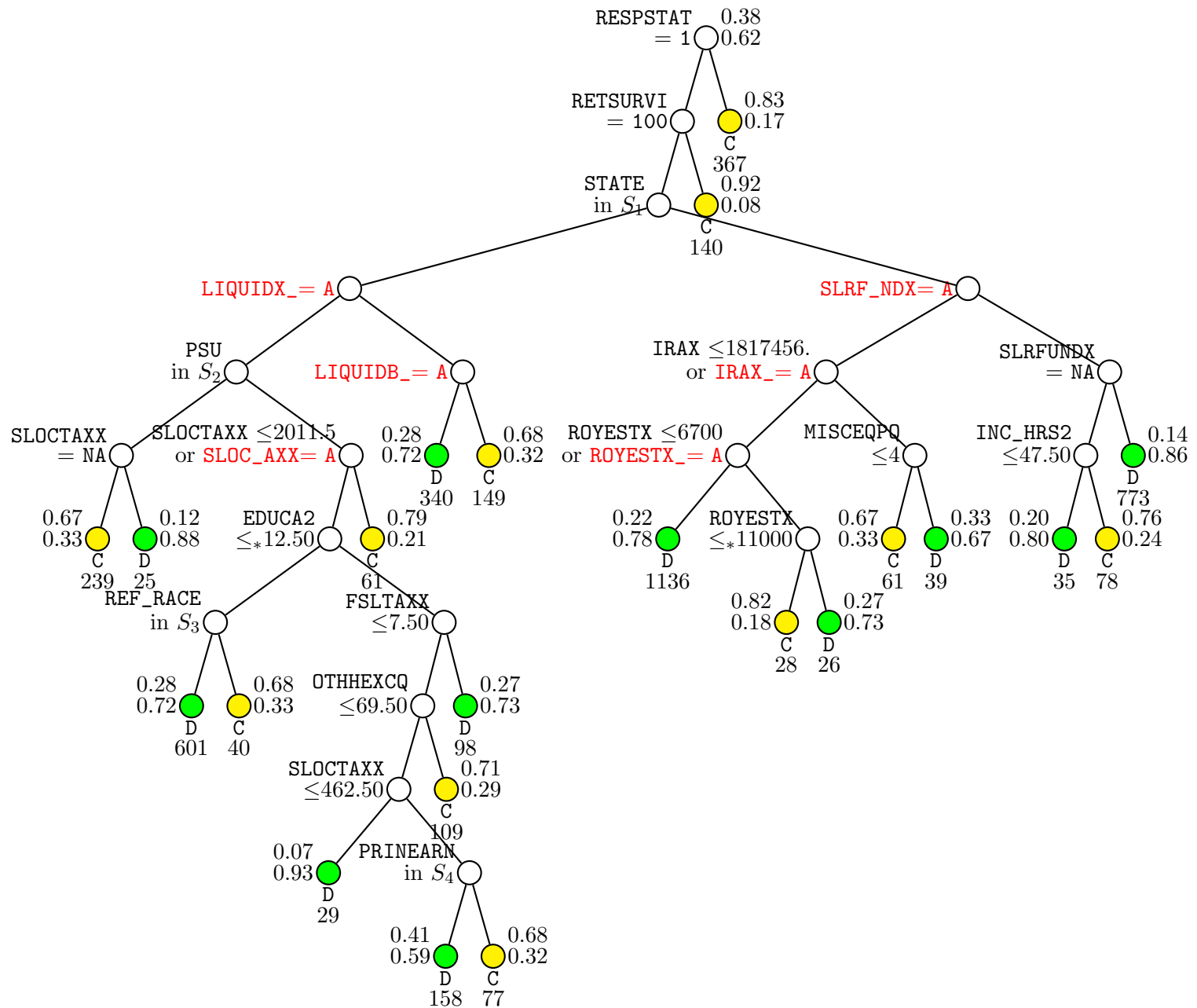


Figure 26: GUIDE v.29.6 0.50-SE pruned classification tree for predicting INTRDVX_.

```

Classification tree
Pruning by cross-validation
Data description file: cecclass.dsc
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Warning: N variables changed to S
Dependent variable is INTRDVX_
Number of records in data file: 4609
Length of longest entry in data file: 11
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Number of classes: 2
Training sample class proportions of D variable INTRDVX_:
Class  #Cases      Proportion
C       1771      0.38424821
D       2838      0.61575179

```

Summary information (without x variables)

d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	DIRACC	c			2	154
3	AGE_REF	s	18.00	87.00		
5	AGE2	s	22.00	87.00		1879
6	AGE2_	m			1	
7	AS_COMP1	s	0.000	4.000		
9	AS_COMP2	s	0.000	4.000		
11	AS_COMP3	s	0.000	4.000		
13	AS_COMP4	s	0.000	4.000		
15	AS_COMP5	s	0.000	2.000		
17	BATHRMQ	s	1.000	8.000		21
18	BATHRMQ_	m			2	
19	BEDROOMQ	s	0.000	9.000		25
20	BEDR_OMQ	m			2	
21	BLS_URBN	s	1.000	2.000		
22	BUILDING	c			10	
24	CUTENURE	c			5	
26	EARNCOMP	c			8	
28	EDUC_REF	s	10.00	16.00		
30	EDUCA2	s	10.00	16.00		1879
32	FAM_SIZE	s	1.000	9.000		

34	FAM_TYPE	c			9	
36	FAMTFEDX	s	0.000	0.9928E+05		
37	FAMT_EDX	c			2	
38	FEDRFNDX	s	4.000	0.1428E+05		2530
39	FEDR_NDX	m			2	
40	FEDTAXX	s	2.000	0.8223E+05		3752
41	FEDTAXX_	m			2	
42	FGOVRETX	s	0.000	0.2469E+05		
44	FINCATAX	s	-0.3380E+06	0.1410E+07		
45	FINCAT_X	c			2	
46	FINCBTAX	s	-0.3430E+06	0.1410E+07		
47	FINCBT_X	c			2	
48	FINDRETX	s	0.000	0.1272E+06		
49	FIND_ETX	c			2	
51	FJSSDEDX	s	0.000	0.3042E+05		
52	FJSS_EDX	c			2	
53	FPRIPENX	s	0.000	0.5902E+05		
54	FPRI_ENX	c			2	
55	FRREDEX	s	0.000	9980.		
56	FRREDEX_	c			2	
57	FRRETIRX	s	0.000	0.5807E+05		
59	FSALARYX	s	0.000	0.5301E+06		
60	FSAL_RYX	c			2	
61	FSLTAXX	s	0.000	0.3010E+05		
62	FSLTAXX_	c			2	
63	FSSIX	s	0.000	0.3048E+05		
65	HLFBATHQ	s	0.000	4.000		23
66	HLFB_THQ	m			2	
67	INC_HRS1	s	2.000	93.00		1697
68	INC__RS1	m			1	
69	INC_HRS2	s	1.000	99.00		2832
70	INC__RS2	m			1	
71	INC_RANK	s	0.1000E-03	1.000		367
72	INC__ANK	m			1	
73	INCNONW1	c			6	2912
74	INCN_NW1	c			2	
75	INCNONW2	c			6	3656
76	INCN_NW2	c			2	
77	INCOMEY1	c			6	1697
78	INCO_EY1	c			2	
79	INCOMEY2	c			6	2832
80	INCO_EY2	c			2	
81	INCWEEK1	s	0.000	52.00		
83	INCWEEK2	s	0.000	52.00		1879
84	INCW_EK2	m			1	
85	MISCTAXX	s	5.000	0.2524E+05		4520

86	MISC_AXX	m			3	
87	LUMPSUMX	s	4.000	0.5492E+06		4378
88	LUMP_UMX	m			2	
89	MARITAL1	c			5	
91	NO_EARNR	s	0.000	6.000		
93	NONINCMX	s	0.000	0.5492E+06		
94	NONI_CMX	c			2	
95	NUM_AUTO	s	0.000	7.000		
97	OCCUCOD1	c			15	1697
98	OCCU_OD1	c			2	
99	OCCUCOD2	c			15	2832
100	OCCU_OD2	c			2	
101	OTHRINCX	s	2.000	0.5788E+05		4483
102	OTHR_NCX	m			2	
103	PERSLT18	s	0.000	7.000		
105	PERSOT64	s	0.000	4.000		
107	POPSIZE	s	1.000	5.000		33
108	PRINEARN	c			7	
110	QINTRVMO	c			12	
111	QINTRVYR	c			2	
112	RACE2	c			6	1879
113	RACE2_	c			2	
114	REF_RACE	c			6	
116	REGION	c			4	33
117	RENTEQVX	s	1.000	4694.		660
118	RENT_QVX	m			1	
119	RESPSTAT	c			2	
121	ROOMSQ	s	1.000	19.00		30
122	ROOMSQ_	m			2	
123	SEX_REF	c			2	
125	SEX2	c			2	1879
126	SEX2_	c			2	
127	SLOCTAXX	s	1.000	0.2657E+05		3990
128	SLOC_AXX	m			2	
129	SLRFUNDX	s	1.000	4169.		3167
130	SLRF_NDX	m			2	
131	SMSASTAT	c			2	
132	ST_HOUS	c			2	
134	TOTTXPDX	s	-0.1845E+05	0.1467E+06		
135	TOTT_PDX	c			2	
136	VEHQ	s	0.000	12.00		
138	WELFAREX	s	300.0	4344.		4596
139	WELF_REX	m			2	
140	TOTEXPPQ	s	233.2	0.1249E+06		
141	TOTEXPCQ	s	-3759.	0.9669E+05		
142	FOODPQ	s	0.000	0.2358E+05		

143	FOODCQ	s	0.000	7363.
144	FDHOMEPQ	s	0.000	8450.
145	FDHOMECPQ	s	0.000	6067.
146	FDAWAYPQ	s	0.000	0.2098E+05
147	FDAWAYCQ	s	0.000	5660.
148	FDXMAPPQ	s	0.000	0.2098E+05
149	FDXMAPCQ	s	0.000	5660.
150	FDMAPPQ	s	0.000	900.0
151	FDMAPCQ	s	0.000	666.7
152	ALCBEPVQ	s	0.000	3152.
153	ALCBEVCQ	s	0.000	2550.
154	HOUSPQ	s	0.000	0.4191E+05
155	HOUSCQ	s	-2196.	0.3466E+05
156	SHELTPQ	s	0.000	0.3070E+05
157	SHELTCQ	s	0.000	0.3354E+05
158	OWNDWEPQ	s	0.000	0.3070E+05
159	OWNDWECQ	s	0.000	0.3321E+05
160	MRTINTPQ	s	0.000	0.2531E+05
161	MRTINTCQ	s	0.000	0.1112E+05
162	PROPTXPQ	s	0.000	5870.
163	PROPTXCQ	s	0.000	4247.
164	MRPINSPQ	s	0.000	0.2110E+05
165	MRPINSCQ	s	0.000	0.2373E+05
166	RENDWEPQ	s	0.000	8546.
167	RENDWECQ	s	0.000	6742.
168	RNTXRPPQ	s	0.000	8546.
169	RNTXRPCQ	s	0.000	6742.
170	RNTAPYPQ	s	0.000	2922.
171	RNTAPYCQ	s	0.000	3000.
172	OTHLQDPQ	s	0.000	0.1616E+05
173	OTHLQDCQ	s	0.000	0.1367E+05
174	UTILPQ	s	0.000	4297.
175	UTILCQ	s	0.000	3661.
176	NTLGASPQ	s	0.000	2306.
177	NTLGASCQ	s	0.000	885.0
178	ELCTRCQPQ	s	0.000	4000.
179	ELCTRCCQ	s	0.000	3261.
180	ALLFULPQ	s	0.000	2752.
181	ALLFULCQ	s	0.000	2628.
182	FULOILPQ	s	0.000	2752.
183	FULOILCQ	s	0.000	2628.
184	OTHFLSPQ	s	0.000	1981.
185	OTHFLSCQ	s	0.000	2269.
186	TELEPHPQ	s	0.000	1638.
187	TELEPHCQ	s	0.000	1907.
188	WATRPSPQ	s	0.000	1880.

189	WATRPSCQ	s	0.000	1035.
190	HOUSOPPQ	s	-37.00	0.2493E+05
191	HOUSOPCQ	s	-4868.	0.1815E+05
192	DOMSRVPQ	s	-37.00	0.2003E+05
193	DOMSRVCQ	s	-4960.	0.1805E+05
194	DMSXCCPQ	s	-37.00	0.2003E+05
195	DMSXCCCQ	s	-4960.	0.1000E+05
196	BBYDAYPQ	s	0.000	0.1500E+05
197	BBYDAYCQ	s	0.000	0.1740E+05
198	OTHHEXPQ	s	0.000	0.2493E+05
199	OTHHEXCQ	s	0.000	5653.
200	HOUSEQPQ	s	0.000	0.2282E+05
201	HOUSEQCQ	s	0.000	0.2268E+05
202	TEXTILPQ	s	0.000	1302.
203	TEXTILCQ	s	0.000	2946.
204	FURNTRPQ	s	0.000	0.1855E+05
205	FURNTRCQ	s	0.000	0.1811E+05
206	FLRCVRPQ	s	0.000	8000.
207	FLRCVRCQ	s	0.000	5500.
208	MAJAPPPQ	s	0.000	0.1802E+05
209	MAJAPPCQ	s	0.000	0.1200E+05
210	SMLAPPPQ	s	0.000	3000.
211	SMLAPPCQ	s	0.000	944.0
212	MISCEQPQ	s	0.000	8280.
213	MISCEQCQ	s	0.000	7155.
214	APPARPQ	s	0.000	0.2440E+05
215	APPARCQ	s	0.000	4604.
216	MENBOYPQ	s	0.000	4200.
217	MENBOYCQ	s	0.000	1797.
218	MENSIXPQ	s	0.000	4200.
219	MENSIXCQ	s	0.000	1797.
220	BOYFIFPQ	s	0.000	2150.
221	BOYFIFCQ	s	0.000	448.0
222	WOMGRLPQ	s	0.000	4540.
223	WOMGRLCQ	s	0.000	2958.
224	WOMSIXPQ	s	0.000	4474.
225	WOMSIXCQ	s	0.000	2958.
226	GRLFIFPQ	s	0.000	1799.
227	GRLFIFCQ	s	0.000	1624.
228	CHLDRNPQ	s	0.000	717.0
229	CHLDRNCQ	s	0.000	961.0
230	FOOTWRPQ	s	0.000	2162.
231	FOOTWRCQ	s	0.000	1148.
232	OTHAPLPQ	s	0.000	0.2048E+05
233	OTHAPLCQ	s	0.000	4076.
234	TRANSPQ	s	0.000	0.4937E+05

235	TRANSCQ	s	0.000	0.6490E+05
236	CARTKNPQ	s	0.000	0.4664E+05
237	CARTKNCQ	s	0.000	0.6480E+05
238	CARTKUPQ	s	0.000	0.4200E+05
239	CARTKUCQ	s	0.000	0.4163E+05
240	OTHVEHPQ	s	0.000	0.1417E+05
241	OTHVEHCQ	s	0.000	0.1800E+05
242	GASMOPQ	s	0.000	4832.
243	GASMOCQ	s	0.000	6400.
244	VEHFINPQ	s	0.000	1201.
245	VEHFINCQ	s	0.000	716.0
246	MAINRPPQ	s	0.000	0.1400E+05
247	MAINRPCQ	s	0.000	8060.
248	VEHINSPQ	s	0.000	4236.
249	VEHINSCQ	s	0.000	3800.
250	VRNTLOPQ	s	0.000	0.2200E+05
251	VRNTLOCQ	s	0.000	0.2223E+05
252	PUBTRAPQ	s	0.000	0.2287E+05
253	PUBTRACQ	s	0.000	0.1198E+05
254	TRNTRPPQ	s	0.000	0.2287E+05
255	TRNTRPCQ	s	0.000	0.1198E+05
256	TRNOTHPQ	s	0.000	1448.
257	TRNOTHCQ	s	0.000	1386.
258	HEALTHPQ	s	-2402.	0.1665E+05
259	HEALTHCQ	s	-0.1281E+05	0.2189E+05
260	HLTHINPQ	s	0.000	0.1426E+05
261	HLTHINCQ	s	0.000	8789.
262	MEDSRVPQ	s	-3290.	0.1543E+05
263	MEDSRVCQ	s	-0.1330E+05	0.1368E+05
264	PREDRGPQ	s	-940.0	6844.
265	PREDRGCQ	s	-260.0	2800.
266	MEDSUPPQ	s	-3600.	7000.
267	MEDSUPCQ	s	-449.0	7530.
268	ENTERTPQ	s	0.000	0.6318E+05
269	ENTERTCQ	s	0.000	0.4249E+05
270	FEEADMPQ	s	0.000	0.1958E+05
271	FEEADMCQ	s	0.000	0.1577E+05
272	TVRDIOPQ	s	0.000	7007.
273	TVRDIOCQ	s	0.000	5143.
274	OTHEQPPQ	s	0.000	0.6300E+05
275	OTHEQPCQ	s	0.000	0.4204E+05
276	PETTOYPQ	s	0.000	0.1165E+05
277	PETTOYCQ	s	0.000	5657.
278	OTHENTPQ	s	0.000	0.6300E+05
279	OTHENTCQ	s	0.000	0.4204E+05
280	PERSCAPQ	s	0.000	1550.

281	PERSCACQ	s	0.000	973.3		
282	READPQ	s	0.000	2066.		
283	READCQ	s	0.000	1100.		
284	EDUCAPQ	s	0.000	0.3850E+05		
285	EDUCACQ	s	0.000	0.3500E+05		
286	TOBACCPQ	s	0.000	2253.		
287	TOBACCCQ	s	0.000	2600.		
288	MISCPQ	s	0.000	0.2305E+05		
289	MISCCQ	s	0.000	0.1703E+05		
290	MISC1PQ	s	0.000	0.2305E+05		
291	MISC1CQ	s	0.000	0.1703E+05		
294	CASHCOPQ	s	0.000	0.8109E+05		
295	CASHCOCQ	s	0.000	0.2150E+05		
296	PERINSPQ	s	0.000	0.7000E+05		
297	PERINSCQ	s	0.000	0.3337E+05		
298	LIFINSPQ	s	0.000	0.7000E+05		
299	LIFINSCQ	s	0.000	0.3100E+05		
300	RETPENPQ	s	0.000	0.2584E+05		
301	RETPENCQ	s	0.000	0.2298E+05		
302	HH_CU_Q	s	1.000	5.000		
304	HHID	c			46	4531
305	HHID_	c			2	
306	POV_CY	c			2	378
307	POV_CY_	c			2	
308	POV_PY	c			2	378
309	POV_PY_	c			2	
310	SWIMPOOL	c			1	4045
311	SWIM_OOL	c			3	
312	APTMENT	c			1	4535
313	APTMENT_	c			3	
314	OFSTPARK	c			1	1160
315	OFST_ARK	c			3	
316	WINDOWAC	c			1	3977
317	WIND_WAC	c			3	
318	CNTRALAC	c			1	1459
319	CNTR_LAC	c			3	
320	CHILDAGE	c			8	
322	INCLASS	s	1.000	9.000		
323	STATE	c			39	486
324	ERANKH	s	0.4735E-02	1.000		367
325	ERANKH_	m			1	
326	TOTEX4PQ	s	233.2	0.1249E+06		
327	TOTEX4CQ	s	-3759.	0.9669E+05		
328	MISX4PQ	s	0.000	0.2305E+05		
329	MISX4CQ	s	0.000	0.1703E+05		
330	VEHQL	s	0.000	4.000		

332	NUM_TVAN	s	0.000	6.000
334	TTOTALP	s	0.000	0.3821E+05
335	TTOTALC	s	0.000	0.2215E+05
336	TFOODTOP	s	0.000	5600.
337	TFOODTOC	s	0.000	2991.
338	TFOODAWP	s	0.000	5500.
339	TFOODAWC	s	0.000	2450.
340	TFOODHOP	s	0.000	3300.
341	TFOODHOC	s	0.000	1050.
342	TALCBEVP	s	0.000	2252.
343	TALCBEVC	s	0.000	1220.
344	TOTHRLOP	s	0.000	9282.
345	TOTHRLOC	s	0.000	4089.
346	TTRANPRP	s	0.000	0.2296E+05
347	TTRANPRC	s	0.000	0.1198E+05
348	TGASMOTP	s	0.000	1750.
349	TGASMOTC	s	0.000	2200.
350	TVRENTLP	s	0.000	445.0
351	TVRENTLC	s	0.000	275.0
356	TOTHTREP	s	0.000	445.0
357	TOTHTREC	s	0.000	275.0
358	TTRNTRIP	s	0.000	0.2287E+05
359	TTRNTRIC	s	0.000	0.1198E+05
360	TFAREP	s	0.000	0.2202E+05
361	TFAREC	s	0.000	0.1126E+05
362	TAIRFARP	s	0.000	0.2086E+05
363	TAIRFARC	s	0.000	6996.
364	TOTHFARP	s	0.000	9800.
365	TOTHFARC	s	0.000	6238.
366	TLOCALTP	s	0.000	853.0
367	TLOCALTC	s	0.000	1000.
368	TENTRMNP	s	0.000	6296.
369	TENTRMNC	s	0.000	4131.
370	TFEESADP	s	0.000	6296.
371	TFEESADC	s	0.000	4131.
372	TOTHENTP	s	0.000	1400.
373	TOTHENTC	s	0.000	2400.
374	OWNVACP	s	0.000	0.1616E+05
375	OWNVACC	s	0.000	0.1367E+05
376	VOTHRLOP	s	0.000	0.1616E+05
377	VOTHRLOC	s	0.000	0.1367E+05
380	UTILOWNP	s	0.000	2077.
381	UTILOWNC	s	0.000	1523.
382	VFUELOIP	s	0.000	682.0
383	VFUELOIC	s	0.000	625.0
384	VOTHRFLP	s	0.000	547.0

385	VOTHRFLC	s	0.000	907.0		
386	VELECTRP	s	0.000	840.0		
387	VELECTRC	s	0.000	988.0		
388	VNATLGAP	s	0.000	2077.		
389	VNATLGAC	s	0.000	201.0		
390	VWATERPP	s	0.000	475.0		
391	VWATERPC	s	0.000	571.0		
392	MRTPRNOP	s	0.000	0.2643E+05		
393	MRTPRNOC	s	0.000	0.1322E+05		
394	UTILRNTF	s	0.000	1157.		
395	UTILRNTC	s	0.000	451.0		
397	RFUELOIC	s	0.000	334.0		
400	RELECTRP	s	0.000	558.0		
401	RELECTRC	s	0.000	209.0		
402	RNATLGAP	s	0.000	254.0		
403	RNATLGAC	s	0.000	89.00		
404	RWATERPP	s	0.000	552.0		
405	RWATERPC	s	0.000	242.0		
406	POVLEVCY	s	0.1145E+05	0.5184E+05		
408	POVLEVRY	s	0.1122E+05	0.5078E+05		
410	PORCH	c			1	997
411	PORCH_	c			3	
412	ETOTALP	s	233.2	0.1321E+06		
413	ETOTALC	s	-2683.	0.7288E+05		
414	ETOTAPX4	s	233.2	0.1321E+06		
415	ETOTACX4	s	-2683.	0.7288E+05		
416	EHOUSNGP	s	0.000	0.4913E+05		
417	EHOUSNGC	s	-2196.	0.3897E+05		
418	ESHELTRP	s	0.000	0.4456E+05		
419	ESHELTRC	s	0.000	0.3786E+05		
420	EOWNDWLP	s	0.000	0.4456E+05		
421	EOWNDWLC	s	0.000	0.3752E+05		
422	EOTHLODP	s	0.000	0.2798E+05		
423	EOTHLODC	s	0.000	0.1433E+05		
424	EMRTPNOP	s	0.000	0.3516E+05		
425	EMRTPNOC	s	0.000	0.2247E+05		
426	EMRTPNVP	s	0.000	0.2643E+05		
427	EMRTPNVC	s	0.000	0.1322E+05		
428	ETRANPTF	s	0.000	0.4132E+05		
429	ETRANPTC	s	0.000	0.5436E+05		
430	EVEHPURP	s	0.000	0.4010E+05		
431	EVEHPURC	s	0.000	0.5400E+05		
432	ECARTKNP	s	0.000	0.4010E+05		
433	ECARTKNC	s	0.000	0.5400E+05		
434	ECARTKUP	s	0.000	0.2643E+05		
435	ECARTKUC	s	0.000	0.2662E+05		

436	EOTHVEHP	s	0.000	0.1166E+05		
437	EOTHVEHC	s	0.000	6542.		
438	EENTRMTP	s	0.000	0.6318E+05		
439	EENTRMTC	s	0.000	0.1605E+05		
440	EOTHENTP	s	0.000	0.6300E+05		
441	EOTHENTC	s	0.000	7502.		
442	ENOMOTRP	s	0.000	7700.		
443	ENOMOTRC	s	0.000	1500.		
444	EMOTRVHP	s	0.000	0.6300E+05		
445	EMOTRVHC	s	0.000	6971.		
446	EENTMSCP	s	0.000	6000.		
447	EENTMSCC	s	0.000	5000.		
448	EMISCELP	s	0.000	0.2305E+05		
449	EMISCELC	s	0.000	0.1703E+05		
450	EMISCMTP	s	0.000	1096.		
451	EMISCMTC	s	0.000	2113.		
452	UNISTRQ	s	1.000	10.00		
455	WELF_EBX	c			2	
456	LUMPSUMB	s	2.000	12.00		4600
457	LUMP_UMB	m			2	
458	LMPSUMBX	s	1200.	0.8000E+05		4600
459	LMP5_MBX	m			2	
460	OTHRINCB	s	5.000	12.00		4603
461	OTHR_NCB	m			2	
464	INCLASS2	s	1.000	7.000		
467	HORREF1	c			6	4448
468	HORREF1_	c			2	
469	HORREF2	c			5	4495
470	HORREF2_	c			2	
471	ERANKHM	s	0.6205E-02	1.000		
473	FGOVRETM	s	0.000	0.2509E+05		
474	FGOV_ETM	c			2	
475	FPRIPENM	s	0.000	0.5826E+05		
476	FPRI_ENM	c			2	
477	FRRDEDM	s	0.000	0.1043E+05		
478	FRRDEDM_	c			2	
479	PSU	c			21	2579
480	HISP_REF	c			2	
481	HISP2	c			2	1879
482	HIGH_EDU	s	10.00	16.00		
483	BUILT	s	1915.	2013.		585
484	BUILT_	m			2	
485	CREFFINX	s	0.000	6629.		4282
486	CRED_INX	m			2	
487	CREDITB	s	1.000	5.000		4584
488	CREDITB_	m			2	

489	CREDITBX	s	250.0	0.2250E+05		4584
490	CRED_TBX	m			2	
491	CREDITX	s	1.000	0.5132E+05		4233
492	CREDITX_	m			2	
493	CREDTYRX	s	0.000	0.5092E+05		4248
494	CRED_YRX	m			2	
495	CREDYRB	s	1.000	6.000		4573
496	CREDYRB_	m			2	
497	CREDYRBX	s	250.0	0.3500E+05		4573
498	CRED_RBX	m			2	
499	DEFBENRP	s	1.000	2.000		3490
500	DEFB_NRP	m			2	
501	EITC	s	1.000	2.000		1032
502	EITC_	m			2	
504	FMLPYRX	s	4.000	4000.		4514
505	FMLP_YRX	m			2	
506	FS_MTHI	s	1.000	12.00		4560
507	FS_MTHI_	m			1	
508	FSMPFRMX	s	-0.4000E+06	0.1090E+07		
516	INTRDVX_	d			2	
524	IRAB	s	1.000	6.000		4432
525	IRAB_	m			2	
526	IRABX	s	1000.	0.7250E+06		4432
527	IRABX_	m			2	
528	IRAX	s	0.000	0.2635E+07		3853
529	IRAX_	m			2	
530	IRAYRB	s	1.000	6.000		4407
531	IRAYRB_	m			2	
532	IRAYRBX	s	1000.	0.7250E+06		4407
533	IRAYRBX_	m			2	
534	IRAYRX	s	0.000	0.2129E+07		3899
535	IRAYRX_	m			2	
536	JFS_AMT	s	0.000	4800.		
538	LIQDYRBX	s	250.0	0.3500E+05		4448
539	LIQD_RBX	m			2	
540	LIQUIDBX	s	250.0	0.3500E+05		4481
541	LIQU_DBX	m			2	
542	LIQUDYRB	s	1.000	6.000		4448
543	LIQU_YRB	m			2	
544	LIQUDYRX	s	0.000	0.5155E+06		3876
545	LIQU_YRX	m			2	
546	LIQUIDB	s	1.000	6.000		4481
547	LIQUIDB_	m			2	
548	LIQUIDX	s	0.000	0.4910E+06		3827
549	LIQUIDX_	m			2	
550	MEALSPAY	s	1.000	2.000		9

551	MEAL_PAY	m			2	
552	MLPAYWKX	s	2.000	300.0		4514
553	MLPA_WKX	m			2	
554	MLPYQWKS	s	1.000	52.00		4508
555	MLPY_WKS	m			2	
556	NETRENTB	s	0.000	12.00		4582
557	NETR_NTB	m			3	
558	NETRENTX	s	-0.5499E+05	0.1148E+06		4258
559	NETR_NTX	m			2	
560	NETRNTBX	s	-2400.	0.7130E+05		4582
561	NETR_TBX	m			2	
562	OTHAStBX	s	0.3000E+05	0.7250E+06		4589
563	OTHA_TBX	m			2	
564	OTHAStB	s	3.000	6.000		4589
565	OTHAStB_	m			3	
566	OTHAStX	s	2.000	0.2767E+07		4564
567	OTHAStX_	m			2	
568	OTHFInX	s	0.000	900.0		4571
569	OTHFInX_	m			2	
570	OTHLONBX	s	250.0	0.2250E+05		4606
571	OTHL_NBX	m			2	
572	OTHLYRBX	s	750.0	0.2250E+05		4605
573	OTHL_RBX	m			2	
574	OTHLNYRB	s	2.000	5.000		4605
575	OTHL_YRB	m			2	
576	OTHLNYRX	s	0.000	0.5500E+05		4559
577	OTHL_YRX	m			2	
578	OTHLON	s	1.000	2.000		3424
579	OTHLON_	m			1	
580	OTHLONB	s	1.000	5.000		4606
581	OTHLONB_	m			2	
582	OTHLONX	s	1.000	0.3106E+06		4555
583	OTHLONX_	m			2	
584	OTHREGBX	s	488.0	0.5000E+05		4594
585	OTHR_GBx	m			2	
586	OTHREGB	s	1.000	12.00		4594
587	OTHREGB_	m			2	
588	OTHREGX	s	36.00	0.6367E+05		4338
589	OTHREGX_	m			2	
590	OTHSYRBX	s	6000.	0.7250E+06		4585
591	OTHS_RBX	m			2	
592	OTHSYRB	s	2.000	6.000		4585
593	OTHS_YRB	m			2	
594	OTHSYRX	s	0.000	0.1533E+07		4572
595	OTHS_YRX	m			2	
596	RETS_RVB	c			3	

597	RETSURVX	s	30.00	0.1269E+06		3520
598	RETS_RVX	m			2	
599	RETSRVBX	s	480.0	0.6200E+05		4542
600	RETS_VBX	m			2	
601	RETSURV	c			2	
603	RETSURVB	s	1.000	12.00		4542
604	RETSURVI	c			3	
605	RETSURVM	s	30.00	0.9303E+05		3289
606	ROYESTBX	s	200.0	0.6000E+05		4570
607	ROYE_TBX	m			2	
608	ROYESTB	s	1.000	12.00		4570
609	ROYESTB_	m			2	
610	ROYESTX	s	1.000	0.1592E+06		4364
611	ROYESTX_	m			2	
612	STCKYRBX	s	1000.	0.7250E+06		4531
613	STCK_RBX	m			2	
614	STDNTYRB	s	3.000	6.000		4591
615	STDN_YRB	m			2	
616	STDNTYRX	s	0.000	0.4100E+06		4483
617	STDN_YRX	m			2	
618	STDYRBX	s	1750.	0.3500E+05		4591
619	STDY_RBX	m			2	
620	STOCKYRB	s	1.000	6.000		4531
621	STOC_YRB	m			2	
622	STOCKYRX	s	0.000	0.5784E+07		4347
623	STOC_YRX	m			2	
624	STOCKB	s	1.000	6.000		4550
625	STOCKB_	m			3	
626	STOCKBX	s	1000.	0.7250E+06		4550
627	STOCKBX_	m			2	
628	STOCKX	s	25.00	0.6587E+07		4319
629	STOCKX_	m			2	
630	STUDFINX	s	0.000	9000.		4511
631	STUD_INX	m			2	
632	STUDNTBX	s	6250.	0.3500E+05		4598
633	STUD_TBX	m			2	
634	STUDNTB	s	4.000	6.000		4598
635	STUDNTB_	m			2	
636	STUDNTX	s	250.0	0.4200E+06		4473
637	STUDNTX_	m			2	
638	WHLFYRBX	s	250.0	0.3500E+05		4564
639	WHLF_RBX	m			2	
640	WHLFYRB	s	1.000	6.000		4564
641	WHLFYRB_	m			2	
642	WHLFYRX	s	0.000	0.7674E+06		4444
643	WHLFYRX_	m			3	

644	WHOLIFBX	s	250.0	0.3500E+05		4571
645	WHOL_FBX	m			2	
646	WHOLIFB	s	1.000	6.000		4571
647	WHOLIFB_	m			3	
648	WHOLIFX	s	1.000	0.7892E+06		4428
649	WHOLIFX_	m			3	
650	TOTXEST	s	-8990.	0.2865E+06		
651	FFTAXOWE	s	-8943.	0.2351E+06		
652	FSTAXOWE	s	-2505.	0.5991E+05		
653	ETOTA	s	1199.	0.1321E+06		

Total	#cases w/	#missing					
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var	
4609	0	4609	71	0	0	412	
#P-var	#M-var	#B-var	#C-var	#I-var			
0	93	0	76	0			

No. cases used for training: 4609

No. cases excluded due to 0 weight or missing D: 0

Univariate split highest priority

Interaction and linear splits 2nd and 3rd priorities

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

Selected tree is based on mean of CV estimates

Warning: No interaction tests; too many predictor variables

Simple node models

Estimated priors

Unit misclassification costs

Split values for N and S variables based on exhaustive search

Max. number of split levels: 14

Min. node sample size: 23

Number of SE's for pruned tree: 0.5000

Size and CV mean cost and SE of subtrees:

Tree	#Tnodes	Mean Cost	SE(Mean)	BSE(Mean)	Median Cost	BSE(Median)
1	122	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
2	121	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
3	120	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
4	119	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
5	118	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
6	117	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
7	116	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
8	115	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
9	113	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
10	112	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
11	111	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
12	110	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03

13	109	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
14	108	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
15	107	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
16	106	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
17	105	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
18	104	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
19	102	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
20	101	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
21	100	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
22	99	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
23	98	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
24	97	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
25	96	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
26	95	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
27	93	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
28	92	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
29	91	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
30	90	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
31	89	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
32	88	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
33	87	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
34	86	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
35	84	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
36	83	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
37	82	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
38	81	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
39	80	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
40	79	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
41	78	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
42	77	2.636E-01	6.490E-03	3.587E-03	2.638E-01	5.224E-03
43	71	2.604E-01	6.464E-03	3.040E-03	2.636E-01	5.249E-03
44	70	2.630E-01	6.485E-03	3.522E-03	2.636E-01	5.584E-03
45	65	2.619E-01	6.476E-03	3.716E-03	2.636E-01	4.589E-03
46	62	2.619E-01	6.476E-03	3.284E-03	2.625E-01	4.811E-03
47	61	2.614E-01	6.473E-03	3.045E-03	2.625E-01	3.179E-03
48	59	2.614E-01	6.473E-03	3.045E-03	2.625E-01	3.179E-03
49	58	2.614E-01	6.473E-03	3.045E-03	2.625E-01	3.179E-03
50	57	2.614E-01	6.473E-03	3.045E-03	2.625E-01	3.179E-03
51	53	2.614E-01	6.473E-03	3.045E-03	2.625E-01	3.179E-03
52	50	2.610E-01	6.469E-03	3.345E-03	2.625E-01	2.788E-03
53	47	2.627E-01	6.483E-03	3.606E-03	2.657E-01	3.775E-03
54	45	2.612E-01	6.471E-03	3.363E-03	2.636E-01	3.334E-03
55	41	2.601E-01	6.462E-03	3.364E-03	2.614E-01	3.675E-03
56+	39	2.601E-01	6.462E-03	3.397E-03	2.614E-01	4.347E-03
57	36	2.591E-01	6.453E-03	3.495E-03	2.625E-01	4.376E-03
58	34	2.588E-01	6.452E-03	4.198E-03	2.625E-01	4.463E-03

59	32	2.604E-01	6.464E-03	4.766E-03	2.657E-01	4.791E-03
60	30	2.591E-01	6.453E-03	4.471E-03	2.625E-01	4.230E-03
61*	29	2.588E-01	6.452E-03	4.437E-03	2.625E-01	3.961E-03
62	25	2.597E-01	6.459E-03	4.208E-03	2.625E-01	3.341E-03
63**	22	2.604E-01	6.464E-03	4.635E-03	2.625E-01	3.540E-03
64	20	2.660E-01	6.509E-03	4.807E-03	2.679E-01	4.457E-03
65	18	2.669E-01	6.515E-03	5.197E-03	2.733E-01	6.604E-03
66	15	2.727E-01	6.560E-03	4.839E-03	2.722E-01	5.101E-03
67	13	2.738E-01	6.568E-03	4.965E-03	2.744E-01	5.949E-03
68	12	2.742E-01	6.571E-03	4.921E-03	2.747E-01	5.613E-03
69	9	2.766E-01	6.589E-03	4.322E-03	2.766E-01	4.971E-03
70	8	2.853E-01	6.651E-03	3.573E-03	2.820E-01	4.945E-03
71	3	3.105E-01	6.815E-03	5.349E-03	3.048E-01	2.731E-03
72	2	3.324E-01	6.939E-03	4.044E-03	3.297E-01	4.633E-03
73	1	3.842E-01	7.165E-03	2.939E-04	3.839E-01	2.091E-04

0-SE tree based on mean is marked with * and has 29 terminal nodes

0-SE tree based on median is marked with + and has 39 terminal nodes

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.

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

Node label	Total cases	Train cases	Predicted class	Node cost	Split variables	Interacting variable
1	4609	4609	D	3.842E-01	RESPSTAT	
2	4242	4242	D	3.461E-01	RETSURVI	
4	4102	4102	D	3.264E-01	STATE	
8	1926	1926	D	4.299E-01	LIQUIDX	
16	1437	1437	D	4.384E-01	PSU	
32	264	264	C	3.788E-01	SLOCTAXX	
64T	239	239	C	3.264E-01	VEHFINCQ	
65T	25	25	D	1.200E-01	-	
33	1173	1173	D	3.973E-01	SLOCTAXX	
66	1112	1112	D	3.759E-01	EDUCA2	
132	641	641	D	3.073E-01	REF_RACE	
264T	601	601	D	2.829E-01	FEDRFNDX	
265T	40	40	C	3.250E-01	-	
133	471	471	D	4.692E-01	FSLTAXX	

266	373	373	C	4.772E-01	OTHHEXCQ
532	264	264	D	4.470E-01	SLOCTAXX
1064T	29	29	D	6.897E-02	-
1065	235	235	D	4.936E-01	PRINEARN
2130T	158	158	D	4.051E-01	AS_COMP3
2131T	77	77	C	3.247E-01	FSTAXOWE
533T	109	109	C	2.936E-01	FSTAXOWE
267T	98	98	D	2.653E-01	EDUC_REF
67T	61	61	C	2.131E-01	HOUSOPPPQ
17	489	489	D	4.049E-01	LIQUIDB
34T	340	340	D	2.824E-01	FSLTAXX
35T	149	149	C	3.154E-01	FEDRFNDX
9	2176	2176	D	2.348E-01	SLRFUNDX
18	1290	1290	D	2.581E-01	IRAX
36	1190	1190	D	2.345E-01	ROYESTX
72T	1136	1136	D	2.192E-01	NETRENTX
73	54	54	C	4.444E-01	ROYESTX
146T	28	28	C	1.786E-01	-
147T	26	26	D	2.692E-01	-
37	100	100	C	4.600E-01	MISCEQPQ
74T	61	61	C	3.279E-01	STATE
75T	39	39	D	3.333E-01	-
19	886	886	D	2.009E-01	SLRFUNDX
38	113	113	C	4.159E-01	INC_HRS2
76T	35	35	D	2.000E-01	-
77T	78	78	C	2.436E-01	READPQ
39T	773	773	D	1.449E-01	NETRENTX
5T	140	140	C	7.857E-02	EITC
3T	367	367	C	1.744E-01	FINCBTAX

Warning: tree very large, omitting node numbers in LaTeX file

Number of terminal nodes of final tree: 22

Total number of nodes of final tree: 43

Second best split variable (based on curvature test) at root node is POV_PY_

Classification tree:

At splits on categorical variables, values not in training data go to the right

Node 1: RESPSTAT = "1"

Node 2: RETSURVI = "100"

Node 4: STATE = "10", "12", "15", "17", "22", "25", "26", "34", "36", "39",
"42", "45", "47", "53", "55", "8"

Node 8: LIQUIDX_ = A

Node 16: PSU = "1102", "1109", "1110", "1423"

Node 32: SLOCTAXX = NA

Node 64: C

```

Node 32: SLOCTAXX not NA
Node 65: D
Node 16: PSU /= "1102", "1109", "1110", "1423"
Node 33: SLOCTAXX <= 2011.5000 or SLOC_AXX = A
Node 66: EDUCA2 <= 12.500000 or NA
Node 132: REF_RACE = "1", "3", "5"
Node 264: D
Node 132: REF_RACE /= "1", "3", "5"
Node 265: C
Node 66: EDUCA2 > 12.500000
Node 133: FSLTAXX <= 7.5000000
Node 266: OTHHEXCQ <= 69.500000
Node 532: SLOCTAXX <= 462.50000
Node 1064: D
Node 532: SLOCTAXX > 462.50000 or NA
Node 1065: PRINEARN = "1", "4"
Node 2130: D
Node 1065: PRINEARN /= "1", "4"
Node 2131: C
Node 266: OTHHEXCQ > 69.500000 or NA
Node 533: C
Node 133: FSLTAXX > 7.5000000 or NA
Node 267: D
Node 33: SLOCTAXX > 2011.5000 or SLOC_AXX = C
Node 67: C
Node 8: LIQUIDX_ not A
Node 17: LIQUIDB_ = A
Node 34: D
Node 17: LIQUIDB_ not A
Node 35: C
Node 4: STATE /= "10", "12", "15", "17", "22", "25", "26", "34", "36", "39",
"42", "45", "47", "53", "55", "8"
Node 9: SLRF_NDX = A
Node 18: IRAX <= 1817455.5 or IRAX_ = A
Node 36: ROYESTX <= 6700.0000 or ROYESTX_ = A
Node 72: D
Node 36: ROYESTX > 6700.0000 or ROYESTX_ = C
Node 73: ROYESTX <= 11000.000 or NA
Node 146: C
Node 73: ROYESTX > 11000.000
Node 147: D
Node 18: IRAX > 1817455.5 or IRAX_ = C
Node 37: MISCEQPQ <= 4.0000000
Node 74: C
Node 37: MISCEQPQ > 4.0000000 or NA
Node 75: D

```

```

Node 9: SLRF_NDX not A
Node 19: SLRFUNDX = NA
Node 38: INC_HRS2 <= 47.500000
Node 76: D
Node 38: INC_HRS2 > 47.500000 or NA
Node 77: C
Node 19: SLRFUNDX not NA
Node 39: D
Node 2: RETSURVI /= "100"
Node 5: C
Node 1: RESPSTAT /= "1"
Node 3: C

```

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

Node 1: Intermediate node

A case goes into Node 2 if RESPSTAT = "1"
RESPSTAT mode = "1"

Class	Number	Posterior
C	1771	0.38425
D	2838	0.61575

Number of training cases misclassified = 1771
Predicted class is D

Node 2: Intermediate node

A case goes into Node 4 if RETSURVI = "100"
RETSURVI mode = "100"

Class	Number	Posterior
C	1468	0.34606
D	2774	0.65394

Number of training cases misclassified = 1468
Predicted class is D

Node 4: Intermediate node

A case goes into Node 8 if STATE =
"10", "12", "15", "17", "22", "25", "26", "34", "36", "39", "42", "45",
"47", "53", "55", "8"
STATE mode = "NA"

Class	Number	Posterior
C	1339	0.32643
D	2763	0.67357

Number of training cases misclassified = 1339
Predicted class is D

Node 8: Intermediate node

A case goes into Node 16 if LIQUIDX_ = A

LIQUIDX mean = 64347.618

Class	Number	Posterior
C	828	0.42991
D	1098	0.57009

Number of training cases misclassified = 828

Predicted class is D

Node 16: Intermediate node

A case goes into Node 32 if PSU =

"1102", "1109", "1110", "1423"

PSU mode = "NA"

Class	Number	Posterior
C	630	0.43841
D	807	0.56159

Number of training cases misclassified = 630

Predicted class is D

Node 32: Intermediate node

A case goes into Node 64 if SLOCTAXX = NA

SLOCTAXX mean = 3290.5200

Class	Number	Posterior
C	164	0.62121
D	100	0.37879

Number of training cases misclassified = 100

Predicted class is C

Node 64: Terminal node

Class	Number	Posterior
C	161	0.67364
D	78	0.32636

Number of training cases misclassified = 78

Predicted class is C

Node 65: Terminal node

Class	Number	Posterior
C	3	0.12000
D	22	0.88000

Number of training cases misclassified = 3

Predicted class is D

Node 33: Intermediate node

A case goes into Node 66 if SLOCTAXX <= 2011.5000 or SLOC_AX = A

SLOCTAXX mean = 841.54348

Class	Number	Posterior
-------	--------	-----------

```

C           466      0.39727
D           707      0.60273
Number of training cases misclassified = 466
Predicted class is D
-----

```

Node 66: Intermediate node

A case goes into Node 132 if EDUCA2 <= 12.500000 or NA
 EDUCA2 mean = 14.017628

```

Class      Number    Posterior
C           418      0.37590
D           694      0.62410

```

Number of training cases misclassified = 418
 Predicted class is D

Node 132: Intermediate node

A case goes into Node 264 if REF_RACE =
 "1", "3", "5"

REF_RACE mode = "1"

```

Class      Number    Posterior
C           197      0.30733
D           444      0.69267

```

Number of training cases misclassified = 197
 Predicted class is D

Node 264: Terminal node

```

Class      Number    Posterior
C           170      0.28286
D           431      0.71714

```

Number of training cases misclassified = 170
 Predicted class is D

Node 265: Terminal node

```

Class      Number    Posterior
C           27       0.67500
D           13       0.32500

```

Number of training cases misclassified = 13
 Predicted class is C

Node 133: Intermediate node

A case goes into Node 266 if FSLTAXX <= 7.5000000
 FSLTAXX mean = 1017.3652

```

Class      Number    Posterior
C           221      0.46921
D           250      0.53079

```

Number of training cases misclassified = 221
 Predicted class is D


```

-----
Node 266: Intermediate node
A case goes into Node 532 if OTHHEXCQ <= 69.500000
OTHHEXCQ mean = 65.928150
Class      Number  Posterior
C           195    0.52279
D           178    0.47721
Number of training cases misclassified = 178
Predicted class is C
-----

```

```

Node 532: Intermediate node
A case goes into Node 1064 if SLOCTAXX <= 462.50000
SLOCTAXX mean = 484.29787
Class      Number  Posterior
C           118    0.44697
D           146    0.55303
Number of training cases misclassified = 118
Predicted class is D
-----

```

```

Node 1064: Terminal node
Class      Number  Posterior
C           2     0.06897
D           27    0.93103
Number of training cases misclassified = 2
Predicted class is D
-----

```

```

Node 1065: Intermediate node
A case goes into Node 2130 if PRINEARN =
  "1", "4"
PRINEARN mode = "1"
Class      Number  Posterior
C           116    0.49362
D           119    0.50638
Number of training cases misclassified = 116
Predicted class is D
-----

```

```

Node 2130: Terminal node
Class      Number  Posterior
C           64     0.40506
D           94     0.59494
Number of training cases misclassified = 64
Predicted class is D
-----

```

```

Node 2131: Terminal node
Class      Number  Posterior
C           52     0.67532

```

```

D                25      0.32468
Number of training cases misclassified = 25
Predicted class is C
-----

```

Node 533: Terminal node

```

Class      Number  Posterior
C          77      0.70642
D          32      0.29358
Number of training cases misclassified = 32
Predicted class is C
-----

```

Node 267: Terminal node

```

Class      Number  Posterior
C          26      0.26531
D          72      0.73469
Number of training cases misclassified = 26
Predicted class is D
-----

```

Node 67: Terminal node

```

Class      Number  Posterior
C          48      0.78689
D          13      0.21311
Number of training cases misclassified = 13
Predicted class is C
-----

```

Node 17: Intermediate node

```

A case goes into Node 34 if LIQUIDB_ = A
LIQUIDB mean = 4.4852941
Class      Number  Posterior
C          198      0.40491
D          291      0.59509
Number of training cases misclassified = 198
Predicted class is D
-----

```

Node 34: Terminal node

```

Class      Number  Posterior
C          96      0.28235
D          244      0.71765
Number of training cases misclassified = 96
Predicted class is D
-----

```

Node 35: Terminal node

```

Class      Number  Posterior
C          102      0.68456
D          47      0.31544
Number of training cases misclassified = 47

```

```

Predicted class is C
-----
Node 9: Intermediate node
A case goes into Node 18 if SLRF_NDX = A
SLRFUNDX mean = 826.06468
Class      Number  Posterior
C           511    0.23483
D          1665    0.76517
Number of training cases misclassified = 511
Predicted class is D
-----
Node 18: Intermediate node
A case goes into Node 36 if IRAX <= 1817455.5 or IRAX_ = A
IRAX mean = 254002.55
Class      Number  Posterior
C           333    0.25814
D           957    0.74186
Number of training cases misclassified = 333
Predicted class is D
-----
Node 36: Intermediate node
A case goes into Node 72 if ROYESTX <= 6700.0000 or ROYESTX_ = A
ROYESTX mean = 19694.151
Class      Number  Posterior
C           279    0.23445
D           911    0.76555
Number of training cases misclassified = 279
Predicted class is D
-----
Node 72: Terminal node
Class      Number  Posterior
C           249    0.21919
D           887    0.78081
Number of training cases misclassified = 249
Predicted class is D
-----
Node 73: Intermediate node
A case goes into Node 146 if ROYESTX <= 11000.000 or NA
ROYESTX mean = 39704.171
Class      Number  Posterior
C           30     0.55556
D           24     0.44444
Number of training cases misclassified = 24
Predicted class is C
-----
Node 146: Terminal node

```

Class	Number	Posterior
C	23	0.82143
D	5	0.17857

Number of training cases misclassified = 5

Predicted class is C

Node 147: Terminal node

Class	Number	Posterior
C	7	0.26923
D	19	0.73077

Number of training cases misclassified = 7

Predicted class is D

Node 37: Intermediate node

A case goes into Node 74 if MISCEQPQ <= 4.0000000

MISCEQPQ mean = 113.70000

Class	Number	Posterior
C	54	0.54000
D	46	0.46000

Number of training cases misclassified = 46

Predicted class is C

Node 74: Terminal node

Class	Number	Posterior
C	41	0.67213
D	20	0.32787

Number of training cases misclassified = 20

Predicted class is C

Node 75: Terminal node

Class	Number	Posterior
C	13	0.33333
D	26	0.66667

Number of training cases misclassified = 13

Predicted class is D

Node 19: Intermediate node

A case goes into Node 38 if SLRFUNDX = NA

SLRFUNDX mean = 826.06468

Class	Number	Posterior
C	178	0.20090
D	708	0.79910

Number of training cases misclassified = 178

Predicted class is D

Node 38: Intermediate node

A case goes into Node 76 if INC_HRS2 <= 47.500000

INC_HRS2 mean = 41.745098

Class	Number	Posterior
C	66	0.58407
D	47	0.41593

Number of training cases misclassified = 47

Predicted class is C

Node 76: Terminal node

Class	Number	Posterior
C	7	0.20000
D	28	0.80000

Number of training cases misclassified = 7

Predicted class is D

Node 77: Terminal node

Class	Number	Posterior
C	59	0.75641
D	19	0.24359

Number of training cases misclassified = 19

Predicted class is C

Node 39: Terminal node

Class	Number	Posterior
C	112	0.14489
D	661	0.85511

Number of training cases misclassified = 112

Predicted class is D

Node 5: Terminal node

Class	Number	Posterior
C	129	0.92143
D	11	0.07857

Number of training cases misclassified = 11

Predicted class is C

Node 3: Terminal node

Class	Number	Posterior
C	303	0.82561
D	64	0.17439

Number of training cases misclassified = 64

Predicted class is C

Classification matrix for training sample:

Predicted	True class	
class	C	D
C	1022	327
D	749	2511
Total	1771	2838

Number of cases used for tree construction: 4609

Number misclassified: 1076

Resubstitution est. of mean misclassification cost: 0.23345628

Observed and fitted values are stored in `class.fit`

LaTeX code for tree is in `class.tex`

R code is stored in `classr.r`

6.2 Regression

The CE data contains a variable `FINLWT21`, giving the sampling weight of each observation, that can be used to fit a weighted least squares regression model to predict `INTRDVX`. This is done by giving `FINLWT21` the `w` specifier in the description file `cereg.dsc`. The resulting piecewise constant tree model is shown in Figure [27](#).

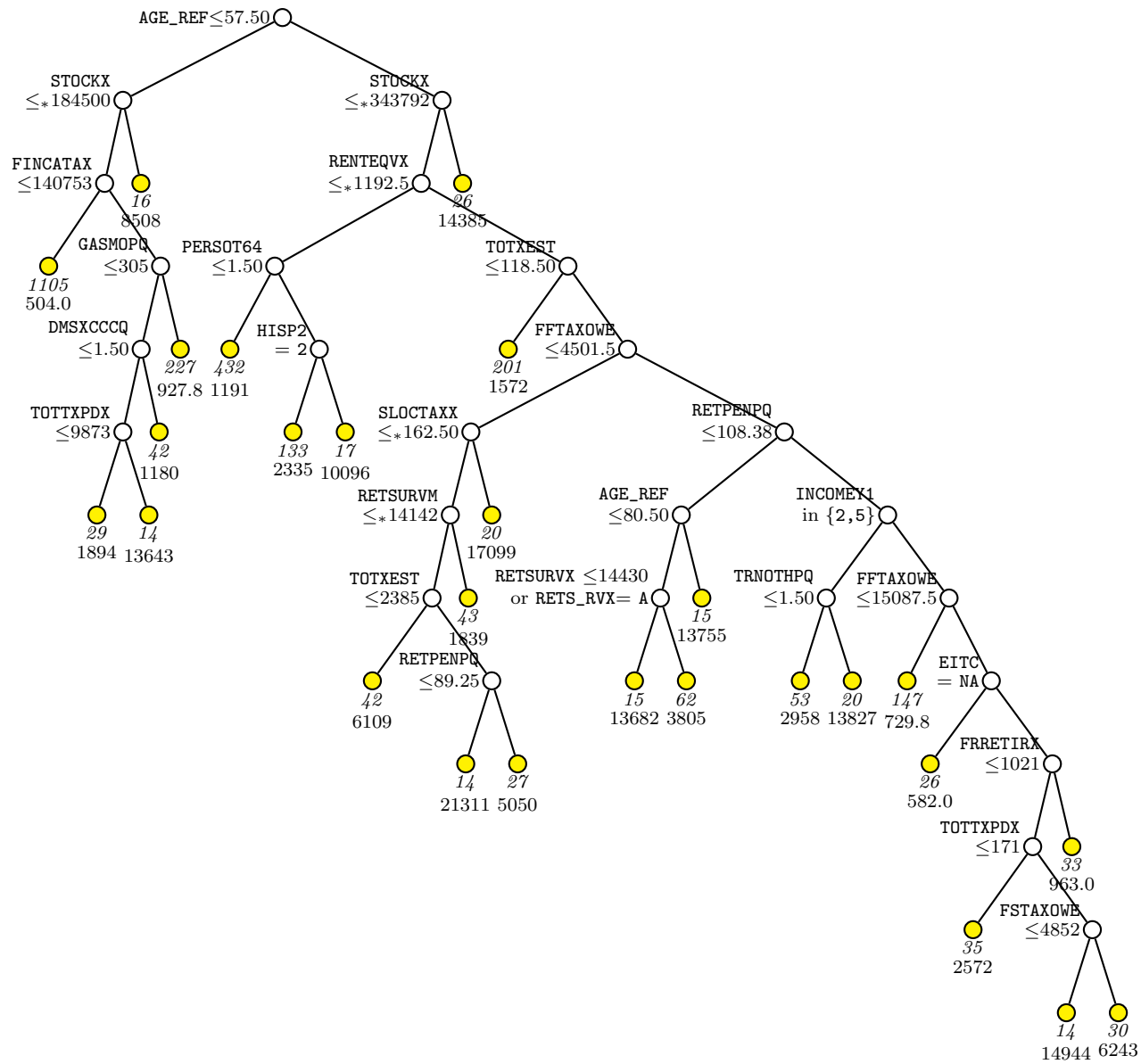


Figure 27: GUIDE v.29.6 0.50-SE pruned piecewise constant least-squares regression tree for predicting INTRDVX.

7 Periodic variables: NHTSA crash tests

Periodic variables that have a cyclic property, such as angular measurements, hour of day, day of week, and month of year, can be used by designating them as P in the description file. There can be multiple P variables in the same data set. Unlike the other types of variables, each line in the description file containing a P variable must have the value of its period (e.g., 360 for angular measurements, 24 for hour of day, 7 for day of week, and 12 for month of year) immediately after P on the same line.

We demonstrate this with the files `nhtsadata.csv` and `nhtsaclass.dsc`, which are obtained from vehicle crash test results from the National Highway Transportation Safety Administration (NHTSA) (www-nrd.nhtsa.dot.gov/database/veh/). The variable HIC is the head injury criterion, which measures the severity of head injury. For this illustration, we construct a classification tree with equal priors to predict the dichotomized variable HIC2, which equals 1 if $HIC > 999$, and equals 0 otherwise. Many experts believe that $HIC > 999$ is absolutely life threatening. The contents of `nhtsaclass.dsc` are reproduced here:

```
nhtsadata.csv
NA
2
1 TSTNO x
2 BARRIG c
3 BARSHP c
4 BARANG p 360
5 BARDIA n
6 OCCTYP c
7 OCCAGE n
8 OCCSEX c
9 OCCHT n
10 OCCWT n
11 MTHCAL x
12 DUMSIZ c
13 HH n
14 HW n
15 HR n
16 HS n
17 CD n
18 CS n
19 AD n
20 HD n
21 KD n
22 HB n
23 NB n
24 CB n
25 KB n
```


26 SEPOSN c
 27 CTRL2 c
 28 HIC x
 29 TSTCFN c
 30 TKSURF c
 31 TKCOND c
 32 TEMP n
 33 RECTYP x
 34 LINK x
 35 CLSSPD n
 36 IMPANG p 360
 37 OFFSET n
 38 IMPPNT s
 39 MAKED c
 40 MODEL D c
 41 YEAR n
 42 BODY c
 43 ENGINE c
 44 ENGDSP n
 45 TRANSM c
 46 VEHTWT n
 47 CURBWT n
 48 WHLBAS n
 49 VEHLEN n
 50 VEHWID n
 51 VEHCG n
 52 STRSEP c
 53 COLMEC c
 54 MODIND c
 55 BX1 n
 56 BX2 n
 57 BX3 n
 58 BX4 n
 59 BX5 n
 60 BX6 n
 61 BX7 n
 62 BX8 n
 63 BX9 n
 64 BX10 n
 65 BX11 n
 66 BX12 n
 67 BX13 n
 68 BX14 n
 69 BX15 n
 70 BX16 n
 71 BX17 n

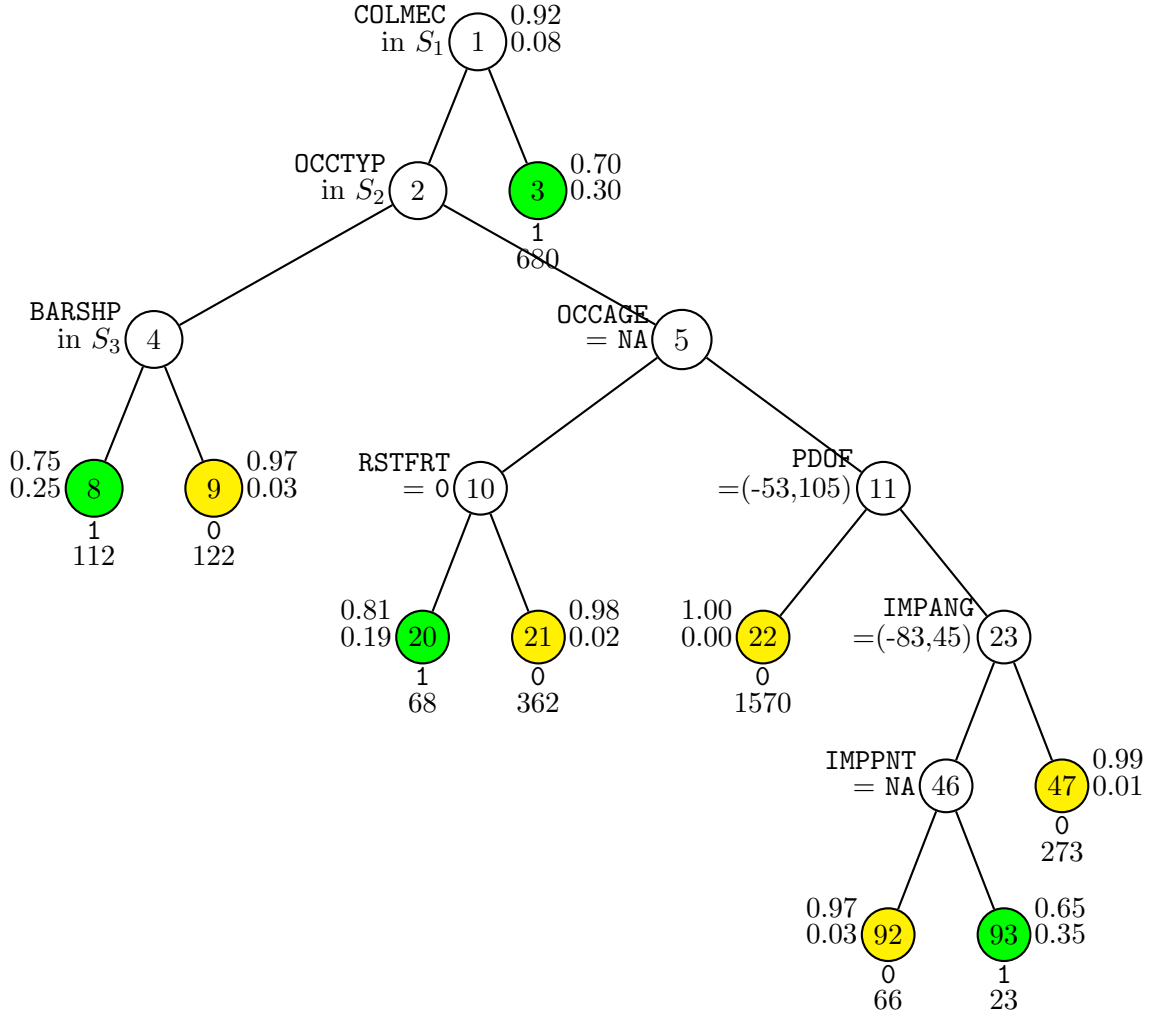
72 BX18 n
73 BX19 n
74 BX20 n
75 BX21 n
76 VEHSPD n
77 CRBANG p 360
78 PDOF p 360
79 BMPENG c
80 SILENG c
81 APLENG c
82 DPD1 x d
83 DPD2 x d
84 DPD3 x d
85 DPD4 x d
86 DPD5 x d
87 DPD6 x d
88 LENCNT x d
89 DAMDST x d
90 CRHDST x d
91 AX1 x d
92 AX2 x d
93 AX3 x d
94 AX4 x d
95 AX5 x d
96 AX6 x d
97 AX7 x d
98 AX8 x d
99 AX9 x d
100 AX10 x d
101 AX11 x d
102 AX12 x d
103 AX13 x d
104 AX14 x d
105 AX15 x d
106 AX16 x d
107 AX17 x d
108 AX18 x d
109 AX19 x d
110 AX20 x d
111 AX21 x d
112 CARANG p 360
113 VEHOR p 360
114 RST3PT c
115 RST5PT c
116 RSTABG c
117 RSTABT c

```
118 RSTBSS c
119 RSTCSF c
120 RSTCSR c
121 RSTCUR c
122 RSTDPL c
123 RSTFCA c
124 RSTFRT c
125 RSTFSS c
126 RSTHDT c
127 RSTISS c
128 RSTKNE c
129 RSTLAP c
130 RSTNAP c
131 RSTNON c
132 RSTOT c
133 RSTOTH c
134 RSTPEL c
135 RSTPS2 c
136 RSTPS3 c
137 RSTSBK c
138 RSTSCE c
139 RSTSHE c
140 RSTSPA c
141 RSTSWE c
142 RSTSWN c
143 RSTTAP c
144 RSTTOR c
145 RSTUNK c
146 RSTVES c
147 HIC2 d
148 HIC3 x
```

The resulting tree is shown in Figure 28. It splits on two angular variables, PDODF (principle direction of force) and IMPANG (impact angle), and that the splits are based on degree intervals.

8 Importance scoring

When there are numerous predictor variables, it may be useful to rank them in order of their “importance”. GUIDE has a facility to do this. In addition, it provides a threshold for distinguishing the important variables from the unimportant ones—see [Loh et al. \(2015\)](#) and [Loh \(2012\)](#); the latter also shows that using GUIDE to find a subset of variables can increase the prediction accuracy of a model.



8.1 Classification: glaucoma data

8.1.1 Input file creation

```

0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: imp.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1): 2
Name of batch output file: imp.out
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: glaucoma.dsc
Reading data description file ...
Training sample file: glaucomadata.txt
Missing value code: NA
Records in data file start on line 2
Warning: N variables changed to S
Dependent variable is Class
Reading data file ...
Number of records in data file: 170
Length of longest entry in data file: 8
Checking for missing values ...
Total number of cases: 170
Number of classes: 2
Re-checking data ...
Allocating missing value information
Assigning codes to categorical and missing values
Data checks complete
Creating missing value indicators
Rereading data
  Class  #Cases   Proportion
glaucoma    85   0.50000000
normal      85   0.50000000
  Total #cases w/ #missing
#cases  miss. D  ord. vals  #X-var  #N-var  #F-var  #S-var
   170      0      17      0      0      0      66
#M-var  #B-var  #C-var
   0      0      0
No. cases used for training: 170
No. cases excluded due to 0 weight or missing D: 0

```

```

Finished reading data file
Input fraction of noise variables erroneously identified as important ([0.00:0.99], <cr>=0.01):
Choose 1 for estimated priors, 2 for equal priors, 3 for priors from a file
Input 1, 2, or 3 ([1:3], <cr>=2):
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([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):
You can also output the importance scores and variable names to a file
Input 1 to create such a file, 2 otherwise ([1:2], <cr>=1):
Input file name: imp.scr
Input file is created!
Run GUIDE with the command: guide < imp.in

```

8.1.2 Contents of imp.out

The most interesting part of the output file is at the end which, for this data set, is given below. The variables are sorted according to their importance scores, with a cut-off value of 1.0 separating the potentially important variables from the unimportant ones—see [Loh \(2012\)](#) and [Loh et al. \(2015\)](#) for details.

```

Predictor variables sorted by importance scores
Importance Scores
Scaled      Unscaled      Rank  Variable
100.0      1.25396E+01      1.00  clv
 74.8      9.38457E+00      2.00  lora
 73.6      9.23261E+00      3.00  vars
 70.8      8.88400E+00      4.00  vari
 67.7      8.48988E+00      5.00  varg
 60.0      7.52936E+00      6.00  tmi
 58.4      7.32422E+00      7.00  rnf
 53.2      6.66948E+00      8.00  tmg
 51.4      6.44574E+00      9.00  vbri
 48.9      6.13271E+00     10.00  cs
 48.0      6.01301E+00     11.00  varn
 46.9      5.87828E+00     12.00  abri
 45.3      5.68463E+00     13.00  phcn
 42.3      5.30802E+00     14.00  hic
 41.9      5.25677E+00     15.00  tms
 39.9      5.00834E+00     16.00  abrs
 39.1      4.89881E+00     17.00  vbrg
 37.2      4.67098E+00     18.00  abrg
 34.9      4.37705E+00     19.00  mhcn
 34.4      4.31498E+00     20.00  phcg
 34.1      4.27212E+00     21.00  vart

```

33.4	4.19392E+00	22.00	mdic
33.2	4.16517E+00	23.00	phci
32.0	4.00955E+00	24.00	abrn
28.7	3.60422E+00	25.00	ean
28.0	3.51577E+00	26.00	vbrs
27.6	3.45686E+00	27.00	mhci
25.6	3.20825E+00	28.00	vbsi
25.2	3.15486E+00	29.00	tmt
23.2	2.90732E+00	30.00	mhcg
23.1	2.89210E+00	31.00	eai
23.1	2.89077E+00	32.00	mhcs
23.0	2.88102E+00	33.00	vbrn
22.9	2.86609E+00	34.00	hvc
22.5	2.82017E+00	35.00	vbsn
21.7	2.71793E+00	36.00	vbsg
21.5	2.69026E+00	37.00	abrt
20.7	2.59410E+00	38.00	vbss
19.8	2.47915E+00	39.00	phcs
19.3	2.42484E+00	40.00	vbrt
18.0	2.25568E+00	41.00	vasi
17.6	2.20710E+00	42.00	eag
17.4	2.17870E+00	43.00	vass
16.7	2.09427E+00	44.00	emd
15.7	1.97208E+00	45.00	vasg
13.9	1.74688E+00	46.00	eas
11.1	1.39356E+00	47.00	tmn
10.5	1.31680E+00	48.00	eat
10.2	1.27765E+00	49.00	vasn
9.7	1.21854E+00	50.00	vbst
8.7	1.08558E+00	51.00	at
----- cut-off -----			
7.9	9.84729E-01	52.00	vast
7.8	9.79693E-01	53.00	ai
7.1	8.91321E-01	54.00	mhct
6.8	8.54121E-01	55.00	mdg
6.1	7.62363E-01	56.00	mdn
4.6	5.71816E-01	57.00	mds
4.1	5.11765E-01	58.00	an
3.1	3.84075E-01	59.00	mdi
3.1	3.82881E-01	60.00	mdt
2.6	3.31943E-01	61.00	ag
2.6	3.28491E-01	62.00	mr
2.5	3.09370E-01	63.00	as
2.1	2.66461E-01	64.00	phct
0.6	8.08811E-02	65.00	tension
0.4	5.50378E-02	66.00	mv

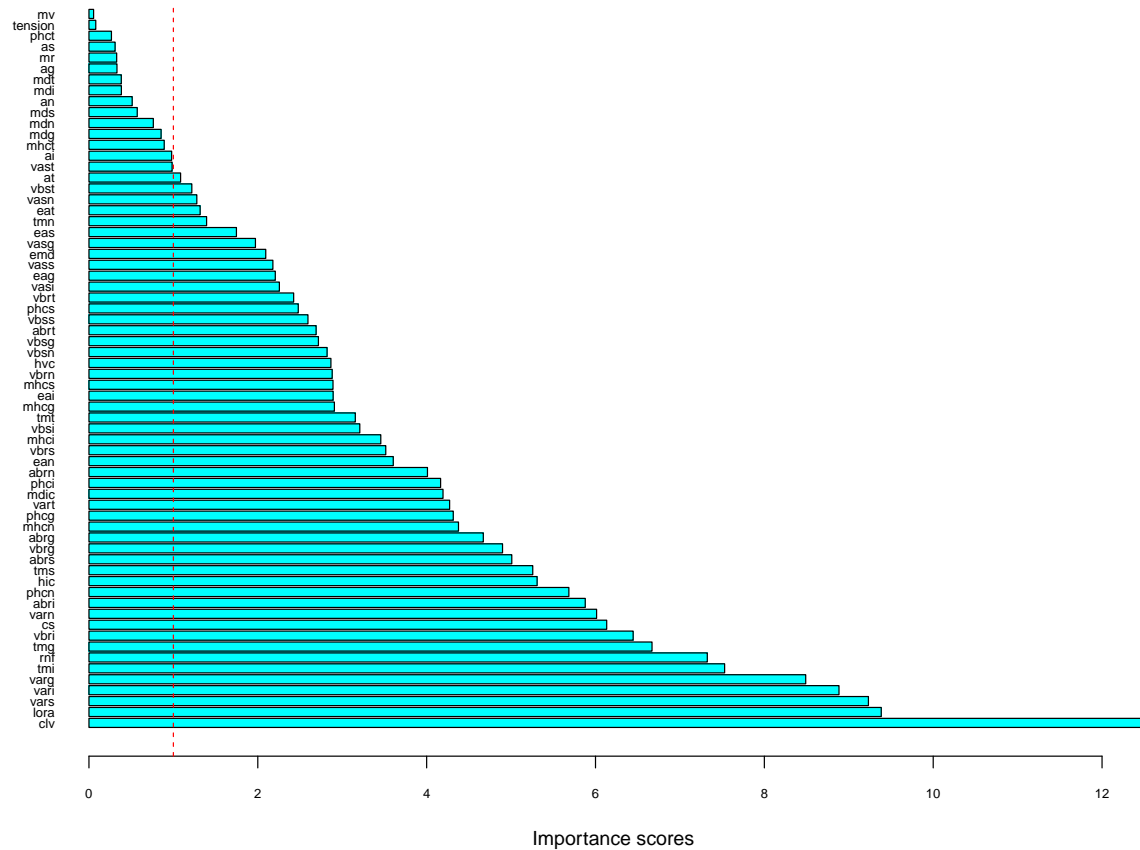


Figure 29: Importance scores for glaucoma data; variables with bars shorter than indicated by the red dashed line are considered unimportant.

Variables with unscaled scores above 1 are important

Number of important and unimportant split variables: 51, 15
Importance scores are stored in imp.scr

The scores are also printed in the file `imp.scr`. Following is the R code for making the graph in Figure 29.

```
z0 <- read.table("imp.scr",header=TRUE)
par(mar=c(5,6,2,1),las=1)
barplot(z0$Score,names.arg=z0$Variable,col="cyan",horiz=TRUE,xlab="Importance scores")
abline(v=1,col="red",lty=2)
```


8.2 Regression with censoring: heart attack data

We now show how to obtain the importance scores for the Worcester Heart Attack Study data analyzed in Section 5.9. We also show how to make GUIDE produce a description file with the unimportant variables designated as 'x'.

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: whas500imp.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1): 2
Name of batch output file: whas500imp.out
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
    1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
    5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 4
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: whas500.dsc
Reading data description file ...
Training sample file: whas500.csv
Missing value code: NA
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is fstat
Reading data file ...
Number of records in data file: 500
Length of longest entry in data file: 10
Checking for missing values ...
Total number of cases: 500
```

Column number	Categorical variable	No. of levels	No. of missing observations
3	gender	2	0
8	cvd	2	0
9	afb	2	0
10	sho	2	0
11	chf	2	0
12	av3	2	0
13	miord	2	0
14	mitype	2	0
15	year	3	0

```

Re-checking data ...
Assigning codes to categorical and missing values
Data checks complete
Smallest uncensored T: 1.0000
No. complete cases excluding censored T < smallest uncensored T: 500
No. cases used to compute baseline hazard: 500
No. cases with D=1 and T >= smallest uncensored: 215
Rereading data
      Total #cases w/ #missing
      #cases miss. D ord. vals #X-var #N-var #F-var #S-var
      500      0      0      5      0      0      6
      #M-var #B-var #C-var
      0      0      9
Survival time variable in column: 21
Event indicator variable in column: 22
Proportion uncensored among nonmissing T and D variables: .430
No. cases used for training: 500
Finished reading data file
Input fraction of noise variables erroneously identified as important ([0.00:0.99], <cr>=0.01):
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): 2
Input 1 to keep only selected variables, 2 to exclude selected variables ([1:2], <cr>=1):
This option produces a description file with unimportant variables marked as 'x'.
Input file name: whas500new.dsc
You can also output the importance scores and variable names to a file
Input 1 to create such a file, 2 otherwise ([1:2], <cr>=1):
Input file name: whas500imp.scr
Input file is created!
Run GUIDE with the command: guide < whas500imp.in

```

Results The importance scores are given at the end of the output file `imp.out` as show below. Variables with scores less that 1.0 (i.e., below the cut-off line) are considered unimportant.

```

Predictor variables sorted by importance scores
Importance Scores
Scaled      Unscaled      Rank  Variable
100.0      1.23262E+01      1.00  age
 83.7      1.03158E+01      2.00  chf
 52.7      6.49611E+00      3.00  year
 38.5      4.74837E+00      4.00  bmi
 35.7      4.39464E+00      5.00  hr
 23.7      2.92171E+00      6.00  diasbp
 20.0      2.46714E+00      7.00  mitype

```

```
15.4    1.89435E+00      8.00  miord
14.3    1.75810E+00      9.00  sho
13.9    1.70820E+00     10.00  gender
13.5    1.65875E+00     11.00  afb
11.1    1.36378E+00     12.00  los
 9.6    1.18736E+00     13.00  sysbp
----- cut-off -----
 6.9    8.49764E-01     14.00  cvd
 2.4    2.89877E-01     15.00  av3
Variables with unscaled scores above 1 are important

Number of important and unimportant split variables: 13, 2
Score scale for categorical variables with 2 and 3 levels = 1.100
Importance scores are stored in whas500imp.scr
Description file with selected variables in whas500new.dsc
```

The scores are also contained in the file `whas500imp.scr` for input into another computer program:

Rank	Score	Variable
1.00	1.23262E+01	age
2.00	1.03158E+01	chf
3.00	6.49611E+00	year
4.00	4.74837E+00	bmi
5.00	4.39464E+00	hr
6.00	2.92171E+00	diasbp
7.00	2.46714E+00	mitype
8.00	1.89435E+00	miord
9.00	1.75810E+00	sho
10.00	1.70820E+00	gender
11.00	1.65875E+00	afb
12.00	1.36378E+00	los
13.00	1.18736E+00	sysbp
14.00	8.49764E-01	cvd
15.00	2.89877E-01	av3

Finally, here are the contents of the file `whas500new.dsc`. It puts an “x” against the variables (`cvd` and `av3` here) that are not important.

```
"whas500.csv"
"NA"
1
1 id x
2 age n
3 gender c
```

```
4 hr n
5 sysbp n
6 diasbp n
7 bmi n
8 cvd x
9 afb c
10 sho c
11 chf c
12 av3 x
13 miord c
14 mitype c
15 year c
16 admitdate x
17 disdate x
18 fdate x
19 los n
20 dstat x
21 lenfol t
22 fstat d
```

9 Differential item functioning: GDS data

GUIDE has an experimental option to identify important predictor variables and items with differential item functioning (DIF) in a data set with two or more item (dependent variable) scores. We illustrate it with a data set from [Broekman et al. \(2011, 2008\)](#) and [Marc et al. \(2008\)](#). It consists of responses from 1978 subjects on 15 items. There are 3 predictor variables (age, education, and gender). The data and description files are `GDS.dat` and `GDS.dsc`. Although the item responses in this example are 0-1, GUIDE allows them to be in any ordinal (e.g., Likert) scale. The contents of `GDS.dsc` are:

```
GDS.dat
NA
1
1 rid x
2 satis d
3 drop d
4 empty d
5 bored d
6 spirit d
7 afraid d
8 happy d
9 help d
10 home d
```

```
11 memory d
12 alive d
13 worth d
14 energy d
15 hope d
16 better d
17 total x
18 gender c
19 education n
20 age n
21 dxcurrent x
22 sumscore x
```

Here is the session log to create an input file for identifying DIF items and the important predictor variables:

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: dif.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1): 2
Name of batch output file: dif.out
Input 1 for classification, 2 for regression, 3 for propensity score grouping
(propensity score grouping is an experimental option)
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
    1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
    5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 5
Choose option 5 for item response data.
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: GDS.dsc
Reading data description file ...
Training sample file: GDS.dat
Missing value code: NA
Records in data file start on line 1
Warning: N variables changed to S
Number of D variables = 15
D variables are:
satis
drop
empty
bored
```

spirit
afraid
happy
help
home
memory
alive
worth
energy
hope
better

Multivariate or univariate split variable selection:

Choose multivariate if there is an order among the D variables; otherwise choose univariate

Input 1 for multivariate, 2 for univariate ([1:2], <cr>=1): 2

Input 1 to normalize D variables, 2 for no normalization ([1:2], <cr>=1): 2

Input 1 for equal, 2 for unequal weighting of D variables ([1:2], <cr>=1):

Reading data file ...

Number of records in data file: 1978

Length of longest entry in data file: 4

Checking for missing values ...

Total number of cases: 1978

Column number	Categorical variable	No. of levels	No. of missing observations
18	gender	2	0

Re-checking data ...

Allocating missing value information

Assigning codes to categorical and missing values

Data checks complete

Creating missing value indicators

Some D variables have missing values

You can use all the data or only those with complete D values

Using only cases with complete D values will reduce the sample size

but allows the option of using PCA for split selection

Input 1 to use all obs, 2 to use obs with complete D values ([1:2], <cr>=2):

Choose option 2 because DIF scoring is desired.

Rereading data

PCA can be used for variable selection

Do not use PCA if differential item functioning (DIF) scores are wanted

Input 1 to use PCA, 2 otherwise ([1:2], <cr>=2):

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

Total #cases	#cases w/ miss. D	#missing ord. vals	#X-var	#N-var	#F-var	#S-var
1978	0	0	4	0	0	2
#B-var	#C-var					
0	1					

```

No. cases used for training: 1977
No. cases excluded due to 0 weight or missing D: 1
Finished reading data file
Input fraction of noise variables erroneously identified as important ([0.00:0.99], <cr>=0.01):
Input 1 to save p-value matrix for differential item functioning (DIF), 2 otherwise ([1:2], <cr>=1):
Input file name to store DIF p-values: dif.pv
  This file is useful for finding the items with DIF.
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):
You can also output the importance scores and variable names to a file
Input 1 to create such a file, 2 otherwise ([1:2], <cr>=1):
Input file name: dif.scr
Input file is created!
Run GUIDE with the command: guide < dif.in

```

The importance scores are in the file `dif.scr`. They show that **gender** is most important, followed closely by **age**.

Rank	Score	Variable
1.00	4.61587E+00	age
2.00	3.76411E+00	gender
3.00	2.33832E+00	education

The word ‘yes’ in the last column of `dif.pv` below shows that item #10 (memory) has DIF.

Item	Itemname	education	age	gender	DIF
1	satis	0.794E-01	0.332E-01	0.924E-01	no
2	drop	0.154E-01	0.143E+00	0.904E+00	no
3	empty	0.499E-03	0.365E-01	0.241E+00	no
4	bored	0.202E-06	0.296E+00	0.360E+00	no
5	spirit	0.972E+00	0.813E+00	0.267E-01	no
6	afraid	0.482E-01	0.154E-02	0.295E-02	no
7	happy	0.825E+00	0.584E+00	0.337E-01	no
8	help	0.216E-01	0.807E+00	0.404E-02	no
9	home	0.221E+00	0.155E+00	0.172E-03	no
10	memory	0.469E+00	0.000E+00	0.641E-02	yes
11	alive	0.259E+00	0.289E+00	0.414E+00	no
12	worth	0.965E-01	0.928E+00	0.648E+00	no
13	energy	0.477E+00	0.759E+00	0.233E-04	no
14	hope	0.509E+00	0.418E+00	0.224E+00	no
15	better	0.409E+00	0.620E+00	0.438E+00	no

10 Tree ensembles

A tree ensemble is a collection of trees. GUIDE has two methods of constructing an ensemble. The preferred one is called “GUIDE forest.” Similar to Random Forest (Breiman, 2001), it fits *unpruned* trees to bootstrap samples and randomly selects a small subset of variables to search for splits at each node. There are, however, two important differences:

1. GUIDE forest uses the unbiased GUIDE method for split selection and Random Forest uses the biased CART method. As a result, GUIDE forest is very much faster than Random Forest if the dependent variable is a class variable having more than two distinct values and there are categorical predictor variables with large numbers of categories. In addition, GUIDE forest is applicable to data with missing values
2. Random Forest (Liaw and Wiener, 2002) requires apriori imputation of missing values in the predictor variables but GUIDE forest does not need imputation.

A second GUIDE ensemble option is called “bagged GUIDE”. It fits *pruned* GUIDE trees to bootstrap samples of the training data (Breiman, 1996). There is some empirical evidence that, if there are many variables of which only a few are useful for prediction, bagged GUIDE is slightly more accurate than GUIDE forest (Loh, 2009, 2012). But GUIDE forest is much faster.

10.1 GUIDE forest: hepatitis data

Recall that in Section 4.4, the hepatitis data gave a null pruned tree due to 80% of the observations belonging to one class.

10.2 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: hepforest.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: hepforest.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1): 2
Input 1 for bagging, 2 for rforest: ([1:2], <cr>=2):
Input 1 for random splits of missing values, 2 for nonrandom: ([1:2], <cr>=2):
```


Random splits is not recommended; it is an experimental option.

Input 1 for classification, 2 for regression

Input your choice ([1:2], <cr>=1):

Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);

enclose with matching quotes if it has spaces: hepdsc.txt

Reading data description file ...

Training sample file: hepdsc.txt

Missing value code: ?

Records in data file start on line 1

Warning: N variables changed to S

Dependent variable is CLASS

Reading data file ...

Number of records in data file: 155

Length of longest entry in data file: 6

Checking for missing values ...

Total number of cases: 155

Missing values found among categorical variables

Separate categories will be created for missing categorical variables

Number of classes: 2

Column number	Categorical variable	No. of levels	No. of missing observations
3	SEX	2	0
4	STEROID	2	1
5	ANTIVIRALS	2	0
6	FATIGUE	2	1
7	MALAISE	2	1
8	ANOREXIA	2	1
9	BIGLIVER	2	10
10	FIRMLIVER	2	11
11	SPLEEN	2	5
12	SPIDERS	2	5
13	ASCITES	2	5
14	VARICES	2	5
20	HISTOLOGY	2	0

Re-checking data ...

Allocating missing value information

Assigning codes to categorical and missing values

Data checks complete

Creating missing value indicators

Rereading data

Class	#Cases	Proportion
die	32	0.20645161
live	123	0.79354839

```

      Total  #cases w/  #missing
#cases  miss. D  ord. vals  #X-var  #N-var  #F-var  #S-var
    155      0      72      0      0      0      6
#M-var  #B-var  #C-var
     0      0      13
No. cases used for training: 155
No. cases excluded due to 0 weight or missing D: 0
Finished reading data file
Choose 1 for estimated priors, 2 for equal priors, 3 for 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):
Input name of file to store predicted class and probability: hepforest.fit
Input file is created!
Run GUIDE with the command: guide < hepforest.in

```

10.3 Results

Warning: Owing to the intrinsic randomness in forests, your results may differ from those shown below.

```

Random forest of classification trees
No pruning
Data description file: hepdsc.txt
Training sample file: hepdat.txt
Missing value code: ?
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is CLASS
Number of records in data file: 155
Length of longest entry in data file: 6
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Number of classes: 2
Training sample class proportions of D variable CLASS:
Class  #Cases    Proportion
die      32    0.20645161
live    123    0.79354839

Summary information (without x variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
Codes/

```

Column	Name		Minimum	Maximum	Levels/ Periods	#Missing
1	CLASS	d			2	
2	AGE	s	7.000	78.00		
3	SEX	c			2	
4	STEROID	c			2	1
5	ANTIVIRALS	c			2	
6	FATIGUE	c			2	1
7	MALAISE	c			2	1
8	ANOREXIA	c			2	1
9	BIGLIVER	c			2	10
10	FIRMLIVER	c			2	11
11	SPLEEN	c			2	5
12	SPIDERS	c			2	5
13	ASCITES	c			2	5
14	VARICES	c			2	5
15	BILIRUBIN	s	0.3000	8.000		6
16	ALKPHOSPHATE	s	26.00	295.0		29
17	SGOT	s	14.00	648.0		4
18	ALBUMIN	s	2.100	6.400		16
19	PROTIME	s	0.000	100.0		67
20	HISTOLOGY	c			2	

Total	#cases w/	#missing					
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var	
155	0	72	0	0	0	6	
#P-var	#M-var	#B-var	#C-var	#I-var			
0	0	0	13	0			

No. cases used for training: 155

No. cases excluded due to 0 weight or missing D: 0

Univariate split highest priority

No interaction and linear splits

Number of trees in ensemble: 500

Number of variables used for splitting: 7

Simple node models

Estimated priors

Unit misclassification costs

Fraction of cases used for splitting each node: 0.64516

Max. number of split levels: 10

Min. node sample size: 5

Mean number of terminal nodes: 10.52

Classification matrix for training sample:

Predicted	True class	
class	die	live

die	14	8
live	18	115
Total	32	123

Number of cases used for tree construction: 155

Number misclassified: 26

Resubstitution est. of mean misclassification cost: 0.16774194

Predicted class probabilities are stored in `hepforest.fit`

Except for the number of observations misclassified, the above results are not particularly useful; they mostly provide a record of the parameter values chosen to construct the forest. The predicted class probabilities in the file `hepforest.fit` are more useful, the top few lines of which are shown below. The first column indicates whether or not the observation is used for training (labeled “y” vs. “n”), followed by its predicted class probabilities. The last two columns give the predicted and observed class labels. For example, observation 7 below has predicted probabilities of 0.3215 and 0.6785 for being in class `die` and `live`, respectively, and its predicted class is `live`.

train	"die"	"live"	predicted	observed
y	0.16179E-01	0.98382E+00	"live"	"live"
y	0.79266E-01	0.92073E+00	"live"	"live"
y	0.22853E-01	0.97715E+00	"live"	"live"
y	0.10441E-01	0.98956E+00	"live"	"live"
y	0.14944E-01	0.98506E+00	"live"	"live"
y	0.65932E-02	0.99341E+00	"live"	"live"
y	0.29461E+00	0.70539E+00	"live"	"die"

10.4 Bagged GUIDE

We now apply bagged GUIDE to the same data.

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: hepbag.in
Input 1 for model fitting, 2 for importance or DIF scoring,
    3 for data conversion ([1:3], <cr>=1):
Name of batch output file: hepbag.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1): 2
Input 1 for bagging, 2 for rforest: ([1:2], <cr>=2): 1
Input 1 for classification, 2 for regression
Input your choice ([1:2], <cr>=1):
```

Input 1 for default options, 2 otherwise ([1:2], <cr>=1):

Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: hepdsc.txt
Reading data description file ...

Training sample file: hepdsc.txt

Missing value code: ?

Records in data file start on line 1

Warning: N variables changed to S

Dependent variable is CLASS

Reading data file ...

Number of records in data file: 155

Length of longest entry in data file: 6

Checking for missing values ...

Total number of cases: 155

Missing values found among categorical variables

Separate categories will be created for missing categorical variables

Number of classes: 2

Column number	Categorical variable	No. of levels	No. of missing observations
3	SEX	2	0
4	STEROID	2	1
5	ANTIVIRALS	2	0
6	FATIGUE	2	1
7	MALAISE	2	1
8	ANOREXIA	2	1
9	BIGLIVER	2	10
10	FIRMLIVER	2	11
11	SPLEEN	2	5
12	SPIDERS	2	5
13	ASCITES	2	5
14	VARICES	2	5
20	HISTOLOGY	2	0

Re-checking data ...

Allocating missing value information

Assigning codes to categorical and missing values

Data checks complete

Creating missing value indicators

Rereading data

Class	#Cases	Proportion
-------	--------	------------

die	32	0.20645161
-----	----	------------

live	123	0.79354839
------	-----	------------

Total #cases	#cases w/ miss. D	#missing ord. vals	#X-var	#N-var	#F-var	#S-var
155	0	72	0	0	0	6

```

#M-var  #B-var  #C-var
      0      0      13
No. cases used for training: 155
No. cases excluded due to 0 weight or missing D: 0
Finished reading data file
Choose 1 for estimated priors, 2 for equal priors, 3 for 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):
Input name of file to store predicted class and probability: hepbag.fit
Input file is created!
Run GUIDE with the command: guide < hepbag.in

```

Results

```

Ensemble of bagged classification trees
Pruning by cross-validation
Data description file: hepdsc.txt
Training sample file: hepdsc.txt
Missing value code: ?
Records in data file start on line 1
Warning: N variables changed to S
Dependent variable is CLASS
Number of records in data file: 155
Length of longest entry in data file: 6
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Number of classes: 2
Training sample class proportions of D variable CLASS:
Class  #Cases    Proportion
die      32      0.20645161
live     123      0.79354839

```

```

Summary information (without x variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight

```

Column	Name		Minimum	Maximum	Codes/ Levels/ Periods	#Missing
1	CLASS	d			2	
2	AGE	s	7.000	78.00		
3	SEX	c			2	
4	STEROID	c			2	1

5	ANTIVIRALS	c				2	
6	FATIGUE	c				2	1
7	MALAISE	c				2	1
8	ANOREXIA	c				2	1
9	BIGLIVER	c				2	10
10	FIRMLIVER	c				2	11
11	SPLEEN	c				2	5
12	SPIDERS	c				2	5
13	ASCITES	c				2	5
14	VARICES	c				2	5
15	BILIRUBIN	s	0.3000	8.000			6
16	ALKPHOSPHATE	s	26.00	295.0			29
17	SGOT	s	14.00	648.0			4
18	ALBUMIN	s	2.100	6.400			16
19	PROTIME	s	0.000	100.0			67
20	HISTOLOGY	c				2	

Total	#cases w/	#missing					
#cases	miss. D	ord. vals	#X-var	#N-var	#F-var	#S-var	
155	0	72	0	0	0	6	
#P-var	#M-var	#B-var	#C-var	#I-var			
0	0	0	13	0			

No. cases used for training: 155

No. cases excluded due to 0 weight or missing D: 0

Univariate split highest priority

Interaction splits 2nd priority; no linear splits

Number of trees in ensemble: 100

Pruning by v-fold cross-validation, with v = 5

Selected tree is based on mean of CV estimates

Simple node models

Estimated priors

Unit misclassification costs

Fraction of cases used for splitting each node: 0.64516

Max. number of split levels: 7

Min. node sample size: 10

Number of SE's for pruned tree: 0.5000

Mean number of terminal nodes: 1.730

Classification matrix for training sample:

Predicted	True class	
class	die	live
die	0	0
live	32	123
Total	32	123

```
Number of cases used for tree construction: 155
Number misclassified: 32
Resubstitution est. of mean misclassification cost: 0.20645161
```

Predicted class probabilities are stored in `hepbag.fit`

The top few lines of `hepbag.fit` follow.

train	"die"	"live"	predicted	observed
y	0.15576E+00	0.84424E+00	"live"	"live"
y	0.18353E+00	0.81647E+00	"live"	"live"
y	0.15566E+00	0.84434E+00	"live"	"live"
y	0.17501E+00	0.82499E+00	"live"	"live"
y	0.16002E+00	0.83998E+00	"live"	"live"
y	0.15634E+00	0.84366E+00	"live"	"live"
y	0.19084E+00	0.80916E+00	"live"	"die"

11 Other features

11.1 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. Pruning with test-samples or no pruning are non-default options.

11.2 Prediction of test samples

GUIDE can produce R code to predict future observations from all except kernel and nearest neighbor classification and ensemble models. This is also a non-default option.

Predictions of the training data for all models can be obtained, however, at the time of tree construction. This feature can be used to obtain predictions on “test samples” (i.e., observations that are not used in tree construction) by adding them to the training sample file. There are two ways to distinguish the test observations from the training observations:

1. Use a *weight* variable (designated as **W** in the description file) that takes value 1 for each training observation and 0 for each test observation.
2. Replace the **D** values of the test observations with the missing value code.

For tree construction, GUIDE does not use observations in the training sample file that have zero weight.

11.3 GUIDE in R and in simulations

GUIDE can be used in simulations or used repeatedly on bootstrap samples to produce an ensemble of tree models. For the latter,

1. Create a file (with name **data.txt**, say) containing one set of bootstrapped data.
2. Create a data description file (with name **desc.txt**, say) that refers to **data.txt**.
3. Create an input file (with name **input.txt**, say) that refers to **desc.txt**.
4. Write a 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**; and
 - (c) reads and processes the results from each GUIDE run.

In R, the command in step 4b depends on the operating system. If the GUIDE program and the files **data.txt** and **input.txt** are in the same folder as the working R directory, the command is:

Linux/Macintosh: `system("guide < input.txt > log.txt")`

Windows: `shell("guide < input.txt > log.txt")`

If the files are not all in the same folder, full path names must be given. Here **log.txt** is a text file that stores messages during execution. If GUIDE does not run successfully, errors are also written to **log.txt**.

11.4 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 demonstrate, suppose we wish to fit a piecewise quadratic model in the variable `wtgain` in the birthweight data. This is easily done by adding one line to the file `birthwt.dsc`. First we assign the **s** (for splitting only) designator to every numerical predictor except `wtgain`. This will prevent all variables other than `wtgain` from acting as regressors in the piecewise quadratic models. To create the variable `wtgain2`, add the line

```
0 8 2 8 0 f
```

to the end of `birthwt.dsc`. The 8's in the above line refer to the column number of the variables `wtgain` in the data file, and the **f** tells the program to use the variable `wtgain2` for fitting terminal node models only. Note: The line defines `wtgain2` as `wtgain2 × wtgain0`. Since we can equivalently define the variable by `wtgain2 = wtgain1 × wtgain1`, we could also have used the line: "0 8 1 8 1 f".

The resulting description file now looks like this:

```
birthwt.dat
NA
1
1 weight d
2 black c
3 married c
4 boy c
5 age s
6 smoke c
7 cigsper s
8 wtgain n
9 visit c
10 ed c
11 lowbwt x
0 8 2 8 0 f
```

When the program is given this description file, the output will show the regression coefficients of `wtgain` and `wtgain2` in each terminal node of the tree.

11.5 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 hepatitis data are reformatted for R or Splus.

```
0. Read the warranty disclaimer
1. Create an input file for model fitting or importance scoring (recommended)
2. Convert data to other formats without creating input file
Input your choice: 3
Input name of log file: log.txt
```

```
Input 1 if D variable is categorical, 2 if real ([1:2], <cr>=1):
```

```
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: hepdsc.txt
```

```
Reading data description file ...
```

```
Training sample file: hepdat.txt
```

```
Missing value code: ?
```

```
Records in data file start on line 1
```

```
Warning: N variables changed to S
```

```
Dependent variable is CLASS
```

```
Reading data file ...
```

```
Number of records in data file: 155
```

```
Length of longest data entry: 6
```

```
Checking for missing values ...
```

```
Total number of cases: 155
```

```
Number of classes = 2
```

Col. no.	Categorical variable	#levels	#missing values
3	SEX	2	0
4	STEROID	2	1
5	ANTIVIRALS	2	0
6	FATIGUE	2	1
7	MALAISE	2	1
8	ANOREXIA	2	1
9	BIGLIVER	2	10
10	FIRMLIVER	2	11
11	SPLEEN	2	5
12	SPIDERS	2	5
13	ASCITES	2	5
14	VARICES	2	5
20	HISTOLOGY	2	0

```
Choose one of the following data formats:
```

```
Field Miss.val.codes
```

No.	Name	Separ	char.	numer.	Remarks
1	R/Splus	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>=1):					
Input name of new data file: hep.rdata					
Follow the commented lines in "hep.rdata" to read the data into R or Splus					

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.
- Broekman, B. F. P., Niti, M., Nyunt, M. S. Z., Ko, S. M., Kumar, R., and Ng, T. P. (2011). Validation of a brief seven-item response bias-free Geriatric Depression Scale. *American Journal of Geriatric Psychiatry*, 19:589–596.
- Broekman, B. F. P., Nyunt, S. Z., Niti, M., Jin, A. Z., Ko, S. M., Kumar, R., Fones, C. S. L., and Ng, T. P. (2008). Differential item functioning of the Geriatric Depression Scale in an Asian population. *Journal of Affective Disorders*, 108:285–290.

- 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.
<http://www.stat.wisc.edu/~loh/treeprogs/lotus/lotus.pdf>.
- Chaudhuri, P., Huang, M.-C., Loh, W.-Y., and Yao, R. (1994). Piecewise-polynomial regression trees. *Statistica Sinica*, 4:143–167.
<http://www3.stat.sinica.edu.tw/statistica/j4n1/j4n18/j4n18.htm>.
- Chaudhuri, P., Lo, W.-D., Loh, W.-Y., and Yang, C.-C. (1995). Generalized regression trees. *Statistica Sinica*, 5:641–666.
<http://www3.stat.sinica.edu.tw/statistica/j5n2/j5n217/j5n217.htm>.
- 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>.
- Choi, Y., Ahn, H., and Chen, J. J. (2005). Regression trees for analysis of count data with extra poisson variation. *Computational Statistics & Data Analysis*, 49(3):893–915.
- Diggle, P. J., Heagerty, P., Liang, K.-Y., and Zeger, S. L. (2002). *Analysis of Longitudinal Data*. Oxford University Press, Oxford, 2nd edition.
- Hosmer, D. W., Lemeshow, S., and May, S. (2008). *Applied Survival Analysis*. Wiley, 2nd edition.
- Hothorn, T. (2017). *TH.data: TH's Data Archive*. R package version 1.0-8.
- Ilter, N. and Guvenir, H. A. (1998). UCI machine learning repository.
- Italiano, A. (2011). Prognostic or predictive? It's time to get back to definitions! *Journal of Clinical Oncology*, 29:4718.
- 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. (2005). *Quantile Regression*. Cambridge University Press.
- Koenker, R. W. and Bassett, G. (1978). Regression quantiles. *Econometrica*, 46:33–50.
- Koenker, R. W. and Hallock, K. (2001). Quantile regression: an introduction. *Journal of Economic Perspectives*, 15:143–156.
- Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.
- 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. <http://www.stat.wisc.edu/~loh/treeprogs/lotus/springer.pdf>.
- 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. 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. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1:14–23.
<http://www.stat.wisc.edu/~loh/treeprogs/guide/wires11.pdf>.
- Loh, W.-Y. (2012). Variable selection for classification and regression in large p , small n problems. In Barbour, A., Chan, H. P., and Siegmund, D., editors, *Probability Approximations and Beyond*, volume 205 of *Lecture Notes in Statistics—Proceedings*, pages 133–157, New York. Springer.
<http://www.stat.wisc.edu/~loh/treeprogs/guide/lchen.pdf>.
- Loh, W.-Y. (2014). Fifty years of classification and regression trees (with discussion). *International Statistical Review*, 34:329–370.
<http://www.stat.wisc.edu/~loh/treeprogs/guide/LohISI14.pdf>.
- 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., Eltinge, J., Cho, M. J., and Li, Y. (2019a). Classification and regression trees and forests for incomplete data from sample surveys. *Statistica Sinica*. In press.
<http://www.stat.wisc.edu/~loh/treeprogs/guide/SS19preprint.pdf>.
- Loh, W.-Y., Fu, H., Man, M., Champion, V., and Yu, M. (2016). Identification of subgroups with differential treatment effects for longitudinal and multiresponse variables. *Statistics in Medicine*, 35:4837–4855.
<http://www.stat.wisc.edu/~loh/treeprogs/guide/LFMCY16.pdf>.
- Loh, W.-Y., He, X., and Man, M. (2015). A regression tree approach to identifying subgroups with differential treatment effects. *Statistics in Medicine*, 34:1818–1833.
<http://www.stat.wisc.edu/~loh/treeprogs/guide/LohHeMan15.pdf>.
- Loh, W.-Y., Man, M., and Wang, S. (2019b). Subgroups from regression trees with adjustment for prognostic effects and post-selection inference. *Statistics in Medicine*. In press.
<http://www.stat.wisc.edu/~loh/treeprogs/guide/sm18.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>.

- Loh, W.-Y. and Vanichsetakul, N. (1988). Tree-structured classification via generalized discriminant analysis (with discussion). *Journal of the American Statistical Association*, 83:715–728.
<http://www.stat.wisc.edu/~loh/treeprogs/fact/LV88.pdf>.
- Loh, W.-Y. and Zheng, W. (2013). Regression trees for longitudinal and multiresponse data. *Annals of Applied Statistics*, 7:495–522.
<http://www.stat.wisc.edu/~loh/treeprogs/guide/A0AS596.pdf>.
- Marc, L. G., Raue, P. J., and Bruce, M. L. (2008). Screening performance of the 15-item Geriatric Depression Scale in a diverse elderly home care population. *American Journal of Geriatric Psychiatry*, 16:914–921.
- Murnane, R. J., Boudett, K. P., and Willett, J. B. (1999). Do male dropouts benefit from obtaining a GED, postsecondary education, and training? *Evaluation Reviews*, 23:475–502.
- Peters, A. and Hothorn, T. (2015). *ipred: Improved Predictors*. R package version 0.9-5.
- 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.
- Schmoor, C., Olschewski, M., and Schumacher, M. (1996). Randomized and non-randomized patients in clinical trials: experiences with comprehensive cohort studies. *Statistics in Medicine*, 15:263–271.
- Segal, M. R. (1988). Regression trees for censored data. *Biometrics*, 44:35–47.
- Singer, J. D. and Willett, J. B. (2003). *Applied Longitudinal Data Analysis*. Oxford University Press, New York, NY.
- Therneau, T., Atkinson, B., and Ripley, B. (2017). *rpart: Recursive Partitioning and Regression Trees*. CRAN.R-project.org/package=rpart.
- 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>.