User Manual for GUIDE ver. 38.0[*]

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

July 16, 2021

Contents

[*] 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 Research and Pfizer.

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. Unbiased importance scoring and thresholding of predictor variables.

3. Automatic handling of missing values without requiring prior imputation.
4. One or more missing value codes and missing-value flag variables.
5. Periodic or cyclic variables, such as angular direction, hour of day, day of week, month of year, and seasons.
6. Subgroup identification for differential treatment effects.
7. Linear splits and kernel and nearest-neighbor node models for classification trees.
8. Weighted least squares, least median of squares, logistic, quantile, Poisson, and relative risk (proportional hazards) regression models.
9. Univariate, multivariate, censored, and longitudinal response variables.
10. Pairwise interaction detection at each node.
11. Categorical variables for splitting only, fitting only (via 0-1 dummy variables), or both in regression tree models.
12. Tree ensembles (bagging and forests).

Tables 1 and 2 compare the features of GUIDE with QUEST, CRUISE, C4.5 (Quinlan, 1993), CTREE (Hothorn et al., 2006), MOB (Hothorn and Zeileis, 2015), 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, 2011, 2014). Advanced features of the algorithm are reported in Chaudhuri and Loh (2002), Loh (2006b, 2008b), Kim et al. (2007), Loh et al. (2007, 2019b, 2016, 2015, 2019c), and Loh and Zhou (2021). For third-party applications of GUIDE and predecessors, see `http://www.stat.wisc.edu/~loh/apps.html`. This manual demonstrates use of the GUIDE software and interpretation of the results.

[1] RPART 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, C4.5, and CTREE classification tree algorithms. Node models: S = simple, K = kernel, L = linear discriminant, N = nearest-neighbor.

	GUIDE	QUEST	CRUISE	RPART	C4.5	CTREE
Unbiased splits	Yes	Yes	Yes	No	No	Yes
Splits per node	2	2	≥ 2	2	2	2
Linear splits	Yes	Yes	Yes	Yes	No	No
Categorical variable splits	Subsets	Subsets	Subsets	Subsets	Atoms	Subsets
Periodic variable splits	Yes	No	No	No	No	No
Interaction tests	Yes	No	Yes	No	No	No
Class priors	Yes	Yes	Yes	Yes	No	No
Misclassification costs	Yes	Yes	Yes	Yes	No	No[a]
Case weights	No[b]	No	No	Yes	Yes	Yes[c]
Node models	S, K, N	S	S, L	S	S	S
Splits on missing values	Separate class	Node mean/mode impute	Surrogate splits	Surrogate splits	Weights	Random splits[d]
Missing-value flag variables	Yes	No	No	No	No	No
Pruning	Yes	Yes	Yes	Yes	No	No
Tree diagrams	Text and LaTeX			R	Text	R
Bagging	Yes	No	No	No	No	No
Forests	Yes	No	No	No	No	cforest
Importance scores	Yes	No	No	Yes	No	Yes

[a] user defined
[b] positive weights treated as 1
[c] non-negative integer counts
[d] surrogate splits is a non-default option

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

	GUIDE	RPART	M5'	MOB
Unbiased splits	Yes	No	No	Yes
Interaction tests	Yes	No	No	No
Loss functions	Weighted least squares, least median of squares, logistic, quantile, Poisson, proportional hazards	Least squares, least absolute deviations	Least squares	Generalized linear models
Censored response	Yes	Yes	No	Yes
Longitudinal and multi-response	Yes	No	No	Yes
Node models	Constant, multiple, stepwise linear, polynomial, ANCOVA	Constant	Constant, stepwise	Constant, multiple linear
Variable roles	Split only, fit only, both, neither, weight, offset	Split only	Split and fit	Similar to GUIDE
Categorical variable splits	Subsets	Subsets	Atomic	Subsets
Periodic variables	Yes	No	No	No
Tree diagrams	Text and LaTeX	R	PostScript	R
Sampling weights	Yes	Yes	No	No[a]
Transformations	Powers and products	No	No	Yes
Missing values in split variables	Separate category	Surrogate splits	Mean/mode imputation	Random splits
Missing values in linear predictors	Node mean imputation	N/A	Imputation	Omitted
Missing-value flag variables	Yes	No	No	No
Bagging & forests	Yes & yes	No & no	No & no	cforest
Importance scores	Yes	Yes	No	Yes[b]

[a] replicate weights only
[b] from cforest or ctree

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 two 64-bit executables to choose from: **Intel ifort** or **Gfortran**. Both versions are compiled in Ubuntu 20.0. Unzip the file with "`gunzip guide.gz`" and, if necessary, make it executable by typing "`chmod a+x guide`" in a `Terminal` window. To execute, type "`./guide`".

macOS: There are four versions to choose from. Double-click the file `guide.gz` to gunzip it and make it executable by typing the command "`chmod a+x guide`" in a `Terminal` application in the folder where the file is located. If this still does not allow you to run the app, carry out these steps:

1. In the Finder on your Mac, locate the file `guide`.
2. Control-click the `guide` icon, then choose **Open** from the shortcut menu.
3. Click **Open**.

Now you can start the program by typing "`./guide`" in the Terminal window where the file `guide` resides.

NAG Fortran. This is version is recommended as it does not require installation of other software. It works for macOS 10.14–11.4 (Mojave, Catalina and Big Sur).

gfortran on Big Sur. This version requires **Xcode 12.4** (or higher) and **gfortran 10.2** (or higher). Follow these steps to ensure that the gfortran libraries are placed in the right place:

1. Install **Xcode** from https://developer.apple.com/xcode/downloads/.
2. Go to http://hpc.sourceforge.net and download file `gcc-10.2-bin.tar.gz` to your Downloads folder. The direct link to the file is http://prdownloads.sourceforge.net/hpc/gcc-10.2-bin.tar.gz?download
3. Open a **Terminal** window and type (or copy and paste):
   (a) `cd ~/Downloads`
   (b) `gunzip gcc-10.2-bin.tar.gz`
   (c) `sudo tar -xvf gcc-10.2-bin.tar -C /`

gfortran on Mojave. This version requires **Xcode 11.3** (or higher) and **gfortran 8.2** (or higher). Follow these steps to ensure that the gfortran libraries are placed in the right place:

1. Install **Xcode** from `https://developer.apple.com/xcode/downloads/`.
2. Go to `https://github.com/fxcoudert/gfortran-for-macOS/releases/tag/8.2` and download the disk image `gfortran-8.2-Mojave.dmg`.
3. Double-click the disk image to install `gfortran 8.2`.

gfortran on High Sierra. This version requires **Xcode 10.1** (or higher) and **gfortran 5.1** (or higher). Follow these steps to ensure that the gfortran libraries are placed in the right place:

1. Install **Xcode** from `https://developer.apple.com/xcode/downloads/`.
2. Go to `http://hpc.sourceforge.net` and download file `gcc-5.1-bin.tar.gz` to your Downloads folder. The direct link to the file is `http://prdownloads.sourceforge.net/hpc/gcc-5.1-bin.tar.gz?download`
3. Open a **Terminal** window and type (or copy and paste):
   (a) `cd ~/Downloads`
   (b) `gunzip gcc-5.1-bin.tar.gz`
   (c) `sudo tar -xvf gcc-5.1-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. Type "`C:\guide`" in a **Command Prompt** window to execute—see Section 3.1.
2. A folder that contains your data files. Type "`guide`" to execute.
3. A folder on your search path. Type "`guide`" to execute.

2.2 LATEX

GUIDE uses the public-domain software LATEX (`http://www.ctan.org`) to produce tree diagrams. The LATEX 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. Both include the **TeXShop** GUI frontend.

Windows: MikTex https://miktex.org/howto/install-miktex or proTeXt http://www.tug.org/protext/. The former includes the **TeXShop** GUI frontend and latter includes **TeXStudio**.

After LaTeX is installed and GUIDE has produced the LaTeX file (say, `diagram.tex`) there are two ways to generate pdf figures of the tree diagrams:

In TeXShop, TeXworks, or TeXStudio. Double-click `diagram.tex` to load it into one of these apps. Select **XeLaTeX** to typeset it to pdf.

In Terminal window. Type the following three commands in a **Terminal** (Linux or Mac) or **Command Prompt** (Win) window.

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

The first command produces a file called `diagram.dvi`. The second command converts the latter to postscript file called `diagram.ps`. The third command turns it into a pdf file with name `diagram.pdf`.

In Mac OSX, the **Preview** app can open postscript and pdf files for conversion to jpg, png, and other formats. In Windows, the same can be done with **ImageMagick** (https://www.imagemagick.org/). For inclusion of the pdf figures in MS PowerPoint or Word documents, convert them to jpg for Mac OSX and png for Windows.

The LaTeX files can be edited to change colors, node sizes, etc. (see the *pstricks manual* at http://tug.org/PSTricks/main.cgi/).

3 Program operation

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

Linux. Any terminal program will do.

Mac OSX. 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**

Do not double-click the GUIDE icon on the desktop!

After the terminal window is opened, change to the folder where the data and program files are stored. Mac and Windows users may consult `http://www.digitalcitizen.life/command-prompt-how-use-basic-commands` and `https://wiredpen.com/resources/basic-unix-commands-for-osx/`, respectively.

3.1 Required files

GUIDE requires two text files to begin.

Data file: This file contains the data from the training sample. Each data record consists of observations on the dependent variable, the predictor (i.e., X or independent) variables, and optional weight, missing value flag, time, offset, periodic, and event indicator (for censored responses) 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 output.

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`.

Note to R users: GUIDE can optionally generate R code for the prediction function of the tree model. But because GUIDE treats `"NA"` (with quotes) the same as `NA` (without quotes), the two are treated as missing values in the R function.

Description file: This provides information about the name and location of the data file, column locations and names of the variables, and their roles in the analysis. Different models may be fitted by changing the roles of the variables. An example description file is `rhcdsc1.txt` whose contents follow.

```
rhcdata.txt
NA
2
1 X x
2 cat1 c
3 cat2 c
4 ca c
5 sadmdte x
6 dschdte x
7 dthdte x
8 lstctdte x
9 death x
10 cardiohx c
11 chfhx c
12 dementhx c
13 psychhx c
14 chrpulhx c
15 renalhx c
16 liverhx c
17 gibledhx c
18 malighx c
19 immunhx c
20 transhx c
21 amihx c
22 age n
23 sex c
24 edu n
25 surv2md1 n
26 das2d3pc n
27 t3d30 x
28 dth30 x
29 aps1 n
30 scoma1 n
```

```
31 meanbp1 n
32 wblc1 n
33 hrt1 n
34 resp1 n
35 temp1 n
36 pafi1 n
37 alb1 n
38 hema1 n
39 bili1 n
40 crea1 n
41 sod1 n
42 pot1 n
43 paco21 n
44 ph1 n
45 swang1 d
46 wtkilo1 n
47 dnr1 c
48 ninsclas c
49 resp c
50 card c
51 neuro c
52 gastr c
53 renal c
54 meta c
55 hema c
56 seps c
57 trauma c
58 ortho c
59 adld3p n
60 urin1 n
61 race c
62 income c
63 ptid x
64 survtime x
```

The 1st line gives the name of the data file. If the file is not in the current folder, its full path must be given (e.g., `"c:\data\rhcdata.txt"`) surrounded by matching quotes (because it contains non-alphanumeric characters). 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 matching 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. A "2" is shown here because the variable names appear in the first line of `rhcdata.txt`. If the 1st line of the data file contains the 1st record, this entry would be "1" . Blank lines in the data and description files are ignored. The column location, 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 special characters, `#`, `%`, `{`, and `}`, in a variable name are replaced by dots (periods) in the outputs. Variable names are truncated to 10 characters in tabular output. Leading and trailing spaces in variable names are dropped.

The letters (lower or upper case) below are the permissible roles.

b Categorical variable used both for splitting and for node modeling in regression. Such variables are converted to 0-1 dummy variables when fitting models within nodes for regression. They are 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. 13), there can only be one **d** variable. For censored responses in proportional hazards models, it is the 0-1 event (death) indicator. For all other models, it is the response variable. It can take character string values for classification.

e Estimated probability variable, for logistic regression without **r** variable; see Section 15 for an example.

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 internally converted to 0-1 indicator variables for fitting regression models within nodes.

m Missing value flag variable. Each such variable should follow immediately after a **c**, **n** or **s** variable in the description file. Missing value flag variables

Table 3: Predictor variable role descriptors

Type of	Role of variable		
variable	Split nodes	Fit node models	Both
Categorical	`c`	`i`	`b`
Numerical	`s`	`f`	`n`

associated with any other variable type (including **p**) should be specified as **c**.

n Numerical variable used both for splitting the nodes and for fitting the node regression models. It is converted to type **s** in classification.

p Periodic (cyclic) variable, such as an angle, hour of day, day of week, or month of year. See Sec. 6 for an example.

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.

s Numerical-valued variable only used for splitting the nodes. It is not used as a linear predictor in in regression models. It is suitable for ordinal categorical variables if they take numerical values that reflect the orderings.

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 Sec. 21.2 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. Models may be fitted to different subsets of variables by indicating excluded variables in the description file without editing the data file.

z Offset variable used only in Poisson regression.

Table 3 summarizes the possible roles for predictor variables.

3.2 Input file creation

GUIDE is started by typing its (lowercase) name in a terminal and then typing “`1`” to answer some questions and save the answers into a file. In the following, the sign (`>`) is the computer prompt (not to be typed!).

```
> guide
GUIDE Classification and Regression Trees and Forests
Version 38.0 (Build date: July 16, 2021)
Compiled with GFortran 10.2.0 on macOS Big Sur 11.4
Copyright (c) 1997-2021 Wei-Yin Loh. All rights reserved.
This software is based upon work partially supported by the U.S. Army Research Office,
National Science Foundation and National Institutes of Health.

Choose one of the following options:
0. Read the warranty disclaimer
1. Create a GUIDE input file
```

4 Classification: RHC data

Doctors believe that direct measurement of cardiac function by right heart catheterization (RHC) is beneficial for some critically ill patients. The file `rhcdata.txt` contains observations on more than 60 variables for 5735 patients from 5 medical centers over 5 years (Connors et al., 1996). The variable `swang1` takes values "`RHC`" and "`NoRHC`", indicating whether or not a patient received RHC. Variable `dth30` is 1 if death occurs within 30 days of hospital admission and 0 otherwise; `death` is 1 if the subject eventually dies and 0 if death is unknown. Other variables are given in Tables 4–7.

To construct a classification tree for predicting `swang1`, we need to generate an input file from the description file `rhcdsc1.txt`, which specifies `swang1` as a `d` variable and `dth30` and `death` both as `x`. When GUIDE prompts 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.

4.1 Univariate splits

The default classification tree employs only one variable to split each node. We demonstrate this first.

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: classin.txt
```

Table 4: RHC demographic & outcome variables [#missing values in brackets]

swang1	Right heart catheterization (RHC) [0]
age	Age in years [0]
sex	Sex (female/male) [0]
wtkilo1	Weight in kilograms [515]
edu	Years of Education [0]
race	Race [0]
income	Income bracket (<11k, 11–25k, 25–50k, >50k) [0]
ninsclas	Medical insurance (Medicaid, Medicare, Medicare & Medicaid, no insurance, private, private & Medicare) [0]
t3d30	Days from admission to death within 30 days [0]
dth30	Death indicator for t3d30 (0=no, 1=yes) [0]
survtime	Days from admission to death or last contact day [0]
death	Death indicator for survtime (0=no, 1=yes) [0]
transhx	Transfer (> 24 hours) from another hospital (no/yes) [0]

Table 5: RHC disease variables [#missing values in brackets]

cat1	Primary disease category (9 levels) [0]
cat2	Secondary disease category (6 levels) [2798]
ca	Cancer (3 levels) [0]
card	Cardiovascular diagnosis [0]
gastr	Gastrointestinal diagnosis [0]
hema	Hematologic diagnosis [0]
meta	Metabolic diagnosis [0]
neuro	Neurological diagnosis [0]
ortho	Orthopedic diagnosis [0]
renal	Renal diagnosis [0]
resp	Respiratory diagnosis [0]
seps	Sepsis diagnosis [0]
trauma	Trauma diagnosis [0]

Table 6: RHC medical history variables [#missing values in brackets]

amihx	Definite myocardial infarction (no/yes) [0]
cardiohx	Acute MI, peripheral vascular disease, severe cardiovascular symptoms [0]
chfhx	Congestive heart failure (no/yes) [0]
chrpulhx	Chronic or severe pulmonary disease (no/yes) [0]
dementhx	Dementia, stroke or cerebral infarction, Parkinson's disease (no/yes) [0]
gibledhx	Upper GI bleeding (no/yes) [0]
liverhx	Cirrhosis, hepatic failure (no/yes) [0]
malighx	Solid tumor, metastatic disease, chronic leukemia/myeloma, acute leukemia, lymphoma (no/yes) [0]
immunhx	Immunosuppression, organ transplant, HIV positivity, diabetes mellitus, connective tissue disease(no/yes) [0]
psychhx	Psychiatric history, active psychosis or severe depression (no/yes) [0]
renalhx	Chronic renal disease, chronic hemodialysis or peritoneal dialysis (no/yes) [0]

```
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: classout.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
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: rhcdsc1.txt
Reading data description file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
20 N variables changed to S
D variable is swang1
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 2
Assigning integer codes to values of 30 categorical variables
```

Table 7: RHC admission variables [#missing values in brackets]; PaO2 is partial pressure of arterial oxygen, FiO2 is fraction of inspired oxygen

alb1	Albumin [0]
bili1	Bilirubin [0]
crea1	Serum creatinine [0]
hema1	Hematocrit [0]
hrt1	Heart rate [159]
meanbp1	Mean blood pressure [80]
pot1	Serum potassium [0]
pafi1	PaO2/(0.01*FiO2) [0]
paco21	Partial pressure of arterial carbon dioxide [0]
ph1	Serum ph [0]
resp1	Respiration rate [136]
scoma1	Glasgow coma score [0]
sod1	Serum sodium [0]
temp1	Temperature (Celsius) [0]
urin1	Urine output [3028]
wblc1	White blood cell count [0]
aps1	APACHE III score ignoring coma [0]
adld3p	Katz Activities of Daily Living Scale [3016]
das2d3pc	DASI (Duke Activity Status Index) [0]
dnr1	DNR (do-not-resuscitate) status [0]
surv2md1	Estimated probability of 2-month survival [0]

```
 Finished assigning codes to 10 categorical variables
 Finished assigning codes to 20 categorical variables
 Finished assigning codes to 30 categorical variables
 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
 NoRHC    3551     0.61918047
 RHC      2184     0.38081953
      Total  #cases w/   #missing
     #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       5735          0       5157      10        0        0       23
     #P-var   #M-var   #B-var   #C-var   #I-var
          0        0        0       30        0
 Number of cases used for training: 5735
 Number of split variables: 53
 Number of cases excluded due to 0 weight or missing D: 0
 Finished reading data file
 Choose 1 for estimated priors, 2 for equal priors, 3 to input 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 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
 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):
 Input name of file to store node ID and fitted value of each case: classfit.txt
 Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
 Input file name: classpred.r
 Input rank of top variable to split root node ([1:50], <cr>=1):
 Input file is created!
 Run GUIDE with the command: guide < classin.txt
```

4.1.2 Contents of classin.txt

The resulting input file is given below. Each line contains a value followed by all the permissible values in parentheses. GUIDE reads only the first value in each row.

```
GUIDE       (do not edit this file unless you know what you are doing)
  38.0      (version of GUIDE that generated this file)
 1          (1=model fitting, 2=importance or DIF scoring, 3=data conversion)
"classout.txt"  (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          (0=tree with fixed no. of nodes, 1=prune by CV, 2=by test sample, 3=no pruning)
"rhcdsc1.txt"  (name of data description file)
        10  (number of cross-validations)
 1          (1=mean-based CV tree, 2=median-based CV tree)
     0.250  (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)
 2          (0=no LaTeX code, 1=tree without node numbers, 2=tree with node numbers)
"class.tex" (latex file name)
 1          (1=color terminal nodes, 2=no colors)
 2          (0=#errors, 1=sample sizes, 2=sample proportions, 3=posterior probs, 4=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)
"classfit.txt" (file name for fitted values and node IDs)
 2          (1=do not write R function, 2=write R function)
"classpred.r" (R code file)
 1          (rank of top variable to split root node)
```

4.1.3 Contents of `classout.txt`

The classification tree model is obtained by executing the command "`guide < classin.txt`" in the terminal window. The output file `classout.txt`, with annotations in blue, follow.

```
 Classification tree
 Pruning by cross-validation
 Data description file: rhcdsc1.txt    name of description file
 Training sample file: rhcdata.txt     name of data file
 Missing value code: NA
 Records in data file start on line 2
 23 N variables changed to S
 D variable is swang1
 Number of records in data file: 5735
 Length of longest entry in data file: 19
 Missing values found among categorical variables
 Separate categories will be created for missing categorical variables
 Missing values found among non-categorical variables
```

```
Number of classes: 2
Training sample class proportions of D variable swang1:
Class  #Cases     Proportion
NoRHC    3551     0.61918047
RHC      2184     0.38081953

Summary information for training sample of size 5735
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
      2  cat1       c                                  9
      3  cat2       c                                  6     4535
      4  ca         c                                  3
     10  cardiohx   c                                  2
     11  chfhx      c                                  2
     12  dementhx   c                                  2
     13  psychhx    c                                  2
     14  chrpulhx   c                                  2
     15  renalhx    c                                  2
     16  liverhx    c                                  2
     17  gibledhx   c                                  2
     18  malighx    c                                  2
     19  immunhx    c                                  2
     20  transhx    c                                  2
     21  amihx      c                                  2
     22  age        s    18.04         101.8
     23  sex        c                                  2
     24  edu        s    0.000         30.00
     25  surv2md1   s    0.000        0.9620
     26  das2d3pc   s    11.00         33.00
     29  aps1       s    3.000         147.0
     30  scoma1     s    0.000         100.0
     31  meanbp1    s    10.00         259.0                    80
     32  wblc1      s    0.000         192.0
     33  hrt1       s    8.000         250.0                   159
     34  resp1      s    2.000         100.0                   136
     35  temp1      s    27.00         43.00
     36  pafi1      s    11.60         937.5
     37  alb1       s   0.3000         29.00
     38  hema1      s    2.000         66.19
     39  bili1      s   0.9999E-01     58.20
     40  crea1      s   0.9999E-01     25.10
     41  sod1       s    101.0        178.0
     42  pot1       s    1.100        11.90
     43  paco21     s    1.000        156.0
     44  ph1        s    6.579        7.770
     45  swang1     d                                  2
     46  wtkilo1    s    19.50        244.0                   515
     47  dnr1       c                                  2
     48  ninsclas   c                                  6
     49  resp       c                                  2
     50  card       c                                  2
     51  neuro      c                                  2
     52  gastr      c                                  2
     53  renal      c                                  2
     54  meta       c                                  2
     55  hema       c                                  2
     56  seps       c                                  2
     57  trauma     c                                  2
     58  ortho      c                                  2
     59  adld3p     s    0.000        7.000                  4296
     60  urin1      s    0.000        9000.                  3028
     61  race       c                                  3
     62  income     c                                  4
The above lists the active variables and their summary statistics.
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157       10        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       30        0
Number of cases used for training: 5735
Number of split variables: 53
Number of cases excluded due to 0 weight or missing D: 0

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Simple node models   node predictions are made by majority rule.
Estimated priors     class priors estimated by sample proportions.
Unit misclassification costs
Univariate split highest priority
Interaction and linear splits 2nd and 3rd priorities
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 15
Minimum node sample size: 57   smallest sample size in a node is 57.
Top-ranked variables and chi-squared values at root node
     1  0.3346E+03   cat1
     2  0.2728E+03   aps1
     3  0.2430E+03   crea1
     4  0.2402E+03   meanbp1
     5  0.2023E+03   pafi1
     :
    50  0.1052E+01   meta
    51  0.6357E+00   race

Size and CV mean cost and SE of subtrees:
 Tree    #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   1       68   3.236E-01   6.178E-03   3.960E-03   3.284E-01   6.780E-03
   2       67   3.236E-01   6.178E-03   3.960E-03   3.284E-01   6.780E-03
   3       66   3.236E-01   6.178E-03   3.960E-03   3.284E-01   6.780E-03
   :
  37       18   3.180E-01   6.150E-03   2.945E-03   3.217E-01   3.907E-03
  38+      12   3.198E-01   6.159E-03   3.064E-03   3.182E-01   3.105E-03
  39**     10   3.180E-01   6.150E-03   2.127E-03   3.188E-01   3.098E-03
  40        8   3.219E-01   6.169E-03   3.105E-03   3.217E-01   5.293E-03
  41        6   3.240E-01   6.180E-03   3.474E-03   3.249E-01   6.673E-03
  42        5   3.228E-01   6.174E-03   3.471E-03   3.249E-01   5.539E-03
  43        3   3.325E-01   6.221E-03   3.956E-03   3.365E-01   6.220E-03
  44        2   3.751E-01   6.393E-03   4.248E-03   3.801E-01   3.186E-03
  45        1   3.808E-01   6.412E-03   2.782E-04   3.805E-01   4.832E-04
Above shows that the largest tree has 68 terminal nodes.
0-SE tree based on mean is marked with * and has 10 terminal nodes
0-SE tree based on median is marked with + and has 12 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
* tree same as -- tree
Pruned tree has 10 terminal nodes and is marked by two asterisks.
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    Total    Train    Predicted        Node    Split           Interacting
      label    cases    cases    class            cost    variables       variable
          1     5735     5735    NoRHC          3.808E-01 cat1
          2     1683     1683    RHC            4.599E-01 meanbp1
          4     1117     1117    RHC            3.796E-01 pafi1
          8T     655      655    RHC            3.038E-01 resp1
          9      462      462    RHC            4.870E-01 ninsclas
         18T     244      244    RHC            3.730E-01 bili1
         19T     218      218    NoRHC          3.853E-01 card
          5T     566      566    NoRHC          3.816E-01 alb1
          3     4052     4052    NoRHC          3.147E-01 pafi1
          6     1292     1292    NoRHC          4.837E-01 resp
         12      581      581    RHC            4.200E-01 dnr1
         24      515      515    RHC            3.903E-01 cat1
         48T     438      438    RHC            3.447E-01 meanbp1
         49T      77       77    NoRHC          3.506E-01 -
         25T      66       66    NoRHC          3.485E-01 -
         13      711      711    NoRHC          4.051E-01 seps
         26T     110      110    RHC            3.636E-01 -
         27T     601      601    NoRHC          3.627E-01 adld3p
          7T    2760     2760    NoRHC          2.355E-01 aps1
Above gives the number of observations in each node (terminal
 node marked with a T), its predicted class, and the split variable.

Number of terminal nodes of final tree: 10
Total number of nodes of final tree: 19
Second best split variable (based on curvature test) at root node is aps1
If cat1 is omitted, aps1 will be chosen to split the root node.

Classification tree:
For categorical variable splits, values not in training data go to the right

Node 1: cat1 = "CHF", "MOSF w/Sepsis"
  Node 2: meanbp1 <= 68.500000 or NA
    Node 4: pafi1 <= 266.15625
      Node 8: RHC
    Node 4: pafi1 > 266.15625 or NA
      Node 9: ninsclas = "No insurance", "Private", "Private & Medicare"
        Node 18: RHC
      Node 9: ninsclas /= "No insurance", "Private", "Private & Medicare"
        Node 19: NoRHC
  Node 2: meanbp1 > 68.500000
    Node 5: NoRHC
Node 1: cat1 /= "CHF", "MOSF w/Sepsis"
  Node 3: pafi1 <= 142.35938
    Node 6: resp = "No"
      Node 12: dnr1 = "No"
        Node 24: cat1 = "ARF", "Lung Cancer", "MOSF w/Malignancy"
          Node 48: RHC
        Node 24: cat1 /= "ARF", "Lung Cancer", "MOSF w/Malignancy"
          Node 49: NoRHC
```

```
       Node 12: dnr1 /= "No"
         Node 25: NoRHC
     Node 6: resp /= "No"
       Node 13: seps = "Yes"
         Node 26: RHC
       Node 13: seps /= "Yes"
         Node 27: NoRHC
   Node 3: pafi1 > 142.35938 or NA
     Node 7: NoRHC

 ***************************************************************

 Predictor means below are means of cases with no missing values.

Node 1: Intermediate node
 A case goes into Node 2 if cat1 = "CHF", "MOSF w/Sepsis"
 cat1 mode = "ARF"
 Class      Number   Posterior
 NoRHC        3551  0.6192E+00
 RHC          2184  0.3808E+00
 Number of training cases misclassified = 2184
 Predicted class is NoRHC
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if meanbp1 <= 68.500000 or NA
 meanbp1 mean = 72.674985
 Class      Number   Posterior
 NoRHC         774  0.4599E+00
 RHC           909  0.5401E+00
 Number of training cases misclassified = 774
 Predicted class is RHC
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if pafi1 <= 266.15625
 pafi1 mean = 241.37331
 Class      Number   Posterior
 NoRHC         424  0.3796E+00
 RHC           693  0.6204E+00
 Number of training cases misclassified = 424
 Predicted class is RHC
 ----------------------------
Node 8: Terminal node
 Class      Number   Posterior
 NoRHC         199  0.3038E+00
 RHC           456  0.6962E+00
 Number of training cases misclassified = 199
 Predicted class is RHC
 ----------------------------
:
:
Node 27: Terminal node
 Class      Number   Posterior
 NoRHC         383  0.6373E+00
 RHC           218  0.3627E+00
 Number of training cases misclassified = 218
 Predicted class is NoRHC
 ----------------------------
Node 7: Terminal node
 Class      Number   Posterior
 NoRHC        2110  0.7645E+00
 RHC           650  0.2355E+00
 Number of training cases misclassified = 650
 Predicted class is NoRHC
 ----------------------------

 Classification matrix for training sample:
 Predicted      True class
 class          NoRHC       RHC
 NoRHC           3070      1218
 RHC              481       966
 Total           3551      2184

 Number of cases used for tree construction: 5735
 Number misclassified: 1699
 Resubstitution estimate of mean misclassification cost: 0.29625109
 Resubstitution estimate = (number misclassified)/(number of cases).

 Observed and fitted values are stored in classfit.txt
 LaTeX code for tree is in class.tex
 R code is stored in classpred.r
```

Figure 1 shows the LaTeX tree. Symbol "$\leq_*$" in the split at node 2, "`meanbp1` $\leq_*$ 68.50", means that observations with missing values in the variable go left. If missing values go right, as in node 3, there is no asterisk beside the inequality sign. The tree diagram can be viewed and saved as pdf by following the directions on page 11.

4.1.4 Contents of `classfit.txt`

Below are the first few lines of the file `classfit.txt`.

```
 train      node   observed     predicted   "P(NoRHC)"    "P(RHC)"
```

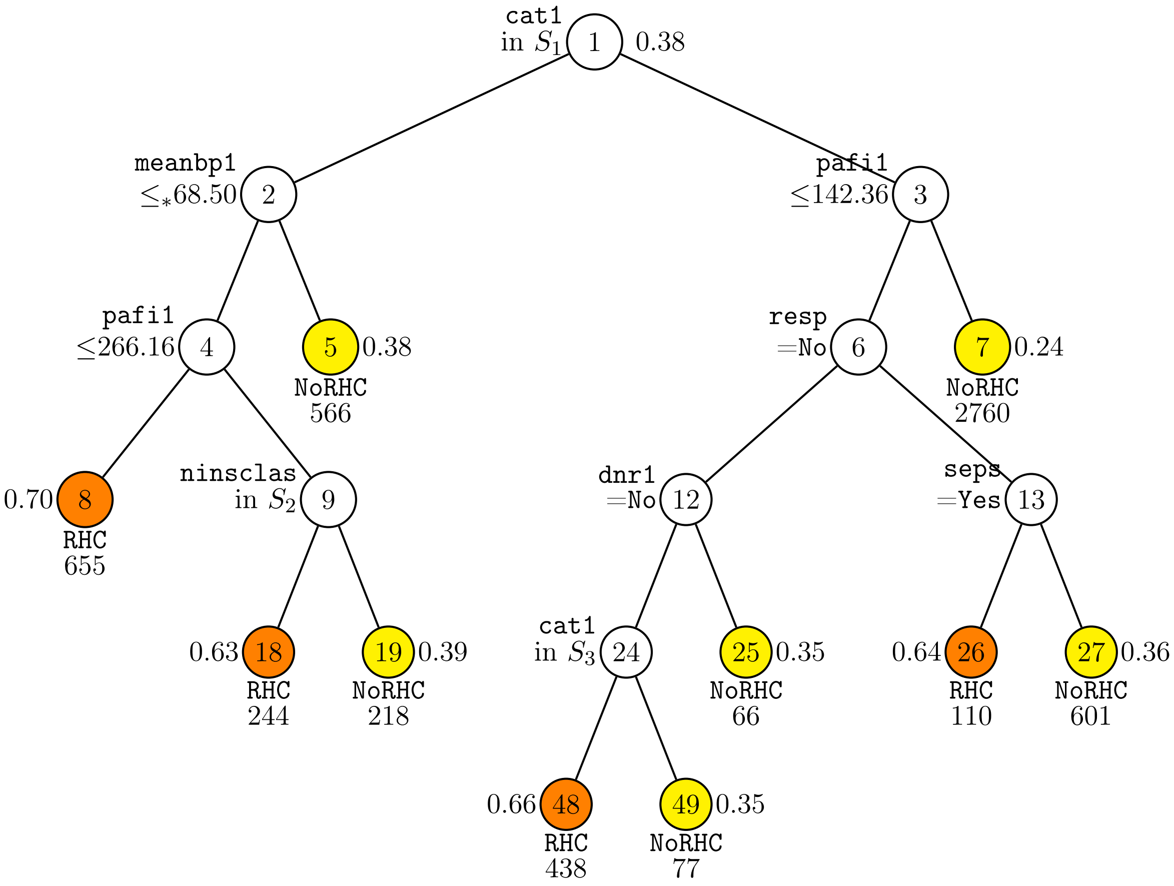

Figure 1: GUIDE v.38.0 0.25-SE classification tree for predicting swang1 using estimated priors and unit misclassification costs. Tree constructed with 5735 observations. Maximum number of split levels is 15 and minimum node sample size is 57. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. Set S_1 = {CHF, MOSF w/Sepsis}. Set S_2 = {No insurance, Private, Private & Medicare}. Set S_3 = {ARF, Lung Cancer, MOSF w/Malignancy}. Predicted classes and sample sizes printed below terminal nodes; class sample proportion for swang1 = RHC beside nodes. Second best split variable at root node is aps1.

```
   y      27    "NoRHC"     "NoRHC"    0.63727E+00  0.36273E+00
   y       8    "RHC"       "RHC"      0.30382E+00  0.69618E+00
   y       7    "RHC"       "NoRHC"    0.76449E+00  0.23551E+00
   y       7    "NoRHC"     "NoRHC"    0.76449E+00  0.23551E+00
   y      19    "RHC"       "NoRHC"    0.61468E+00  0.38532E+00
```

The row in this file match those in the data file. The meanings of the columns are:

train: equals "y" (for "yes") if the observation was used in model construction; otherwise "n" (for "no"). All the values in this example are "y" because every observation is used. Two typical situations where this value is n are (i) if its d variable value is missing and (ii) if there is a weight variable in the data that takes value 0 for the observation.

node: label of the terminal node the observation belongs to. For example, the first observation landed in node 27.

observed: value of the d variable for this observation in the data file.

predicted: predicted value of the d variable for this observation.

P(NoRHC): estimated posterior probability that the observation is in class "NoRHC".

P(RHC): estimated posterior probability that the observation is in class "RHC".

The posterior probabilities are calculated as follows. Let J be the number of classes, N_j be the number of class j observations in the whole sample and $N = \sum_j N_j$. Let π_j be the (estimated or specified) prior probability of class j. Let $n_j(t)$ be the number of class j training samples in node t. The posterior probability of class j in t is $p_j(t) = \pi_j n_j(t) N_j^{-1} / \sum_i \pi_i n_i(t) N_i^{-1}$. If $\min_j p_j(t) = 0$, the posterior probability is redefined to be $(Np_j(t) + \pi_j)/(N + 1)$; this ensures that no probability is zero if all π_j are positive.

4.1.5 Contents of classpred.r

The file classpred.r gives an R function for computing the predicted class and posterior probabilities.

```
predicted <- function(){
 catvalues <- c("CHF","MOSF w/Sepsis")
 if(cat1 %in% catvalues){
   if(is.na(meanbp1) | meanbp1 <= 68.5000000000 ){
     if(!is.na(pafi1) & pafi1 <= 266.156250000 ){
       nodeid <- 8
       predclass <- "RHC"
       posterior <- c( 0.30382E+00, 0.69618E+00)
     } else {
       catvalues <- c("No insurance","Private","Private & Medicare")
       if(ninsclas %in% catvalues){
         nodeid <- 18
         predclass <- "RHC"
         posterior <- c( 0.37295E+00, 0.62705E+00)
       } else {
         nodeid <- 19
         predclass <- "NoRHC"
         posterior <- c( 0.61468E+00, 0.38532E+00)
       }
     }
   } else {
     nodeid <- 5
     predclass <- "NoRHC"
     posterior <- c( 0.61837E+00, 0.38163E+00)
   }
 } else {
   if(!is.na(pafi1) & pafi1 <= 142.359375000 ){
     catvalues <- c("No")
     if(resp %in% catvalues){
       catvalues <- c("No")
       if(dnr1 %in% catvalues){
         catvalues <- c("ARF","Lung Cancer","MOSF w/Malignancy")
         if(cat1 %in% catvalues){
           nodeid <- 48
           predclass <- "RHC"
           posterior <- c( 0.34475E+00, 0.65525E+00)
         } else {
           nodeid <- 49
           predclass <- "NoRHC"
           posterior <- c( 0.64935E+00, 0.35065E+00)
         }
       } else {
         nodeid <- 25
         predclass <- "NoRHC"
         posterior <- c( 0.65152E+00, 0.34848E+00)
       }
     } else {
       catvalues <- c("Yes")
       if(seps %in% catvalues){
         nodeid <- 26
         predclass <- "RHC"
         posterior <- c( 0.36364E+00, 0.63636E+00)
       } else {
         nodeid <- 27
         predclass <- "NoRHC"
         posterior <- c( 0.63727E+00, 0.36273E+00)
       }
     }
   } else {
     nodeid <- 7
     predclass <- "NoRHC"
     posterior <- c( 0.76449E+00, 0.23551E+00)
   }
 }
 return(c(nodeid,predclass,posterior))
}
## end of function
##
##
## newdata.txt is the file containing the data to be predicted
## Missing value code is NA
newdata <- read.table("newdata.txt",header=TRUE,colClasses="character")
## node contains terminal node ID of each case
## pred.class contains predicted class
## pred contains predicted posterior probabilities
node <- NULL
pred <- NULL
pred.class <- NULL
for(i in 1:nrow(newdata)){
    cat1 <- as.character(newdata$cat1[i])
    meanbp1 <- as.numeric(newdata$meanbp1[i])
    pafi1 <- as.numeric(newdata$pafi1[i])
    dnr1 <- as.character(newdata$dnr1[i])
    ninsclas <- as.character(newdata$ninsclas[i])
    resp <- as.character(newdata$resp[i])
    seps <- as.character(newdata$seps[i])
    tmp <- predicted()
    node <- c(node,as.numeric(tmp[1]))
    pred.class <- rbind(pred.class,tmp[2])
    pred <- rbind(pred,as.numeric(tmp[-c(1,2)]))
}
```

4.2 Linear splits

The classification tree in Figure 1 can sometimes be reduced in size if we employ two ordinal variables to split each node. This can be done by selecting a non-default option.

4.2.1 Input file generation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: linearin.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: linearout.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
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):
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
Input 0 to specify tree with fixed no. of nodes, 1 to prune by CV, 2 by test sample,
      3 for no pruning ([0:3], <cr>=1):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: rhcdsc1.txt
Reading data description file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
20 N variables changed to S
D variable is swang1
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 2
Assigning integer codes to values of 30 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Class  #Cases     Proportion
NoRHC    3551     0.61918047
RHC      2184     0.38081953
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157       10        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       30        0
Number of cases used for training: 5735
Number of split variables: 53
Number of 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>=0.25):
Choose 1 for estimated priors, 2 for equal priors, 3 to input 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: 15
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 57
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): linear.tex
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 sample sizes, 2 for sample proportions,
      3 for posterior probs, 4 for nothing
Input your choice ([0:4], <cr>=2):
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):
 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: linearfit.txt
 Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
 Input file name: linearpred.r
 Input rank of top variable to split root node ([1:50], <cr>=1):
 Input file is created!
 Run GUIDE with the command: guide < linearin.txt
```

4.2.2 Contents of linearin.txt

```
GUIDE       (do not edit this file unless you know what you are doing)
  38.0      (version of GUIDE that generated this file)
 1          (1=model fitting, 2=importance or DIF scoring, 3=data conversion)
"linearout.txt"  (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)
 0          (0=linear 1st, 1=univariate 1st, 2=skip linear, 3=skip linear and interaction)
 1          (0=tree with fixed no. of nodes, 1=prune by CV, 2=by test sample, 3=no pruning)
"rhcdsc1.txt"  (name of data description file)
        10  (number of cross-validations)
 1          (1=mean-based CV tree, 2=median-based CV tree)
     0.250  (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)
 2          (0=no LaTeX code, 1=tree without node numbers, 2=tree with node numbers)
"linear.tex" (latex file name)
 1          (1=color terminal nodes, 2=no colors)
 2          (0=#errors, 1=sample sizes, 2=sample proportions, 3=posterior probs, 4=nothing)
 1          (1=no storage, 2=store split variables and values)
 2          (1=do not save fitted values and node IDs, 2=save in a file)
"linearfit.txt" (file name for fitted values and node IDs)
 2          (1=do not write R function, 2=write R function)
"linearpred.r" (R code file)
 1          (rank of top variable to split root node)
```

4.2.3 Contents of linearout.txt

```
 Classification tree
```

```
Pruning by cross-validation
Data description file: rhcdsc1.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
23 N variables changed to S
D variable is swang1
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 2
Training sample class proportions of D variable swang1:
Class  #Cases     Proportion
NoRHC    3551     0.61918047
RHC      2184     0.38081953

Summary information for training sample of size 5735
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
      2  cat1       c                                  9
      3  cat2       c                                  6     4535
      4  ca         c                                  3
      :
     44  ph1        s    6.579        7.770
     45  swang1     d                                  2
     46  wtkilo1    s    19.50        244.0                   515
     47  dnr1       c                                  2
     48  ninsclas   c                                  6
     49  resp       c                                  2
     50  card       c                                  2
     51  neuro      c                                  2
     52  gastr      c                                  2
     53  renal      c                                  2
     54  meta       c                                  2
     55  hema       c                                  2
     56  seps       c                                  2
     57  trauma     c                                  2
     58  ortho      c                                  2
     59  adld3p     s    0.000        7.000                  4296
     60  urin1      s    0.000        9000.                  3028
     61  race       c                                  3
     62  income     c                                  4

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals    #X-var   #N-var   #F-var   #S-var
      5735          0       5157       10        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       30        0
Number of cases used for training: 5735
Number of split variables: 53
Number of cases excluded due to 0 weight or missing D: 0

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Simple node models
Estimated priors
Unit misclassification costs
Linear split highest priority
Interaction and linear splits 2nd and 3rd priorities
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 15
Minimum node sample size: 57
Top-ranked variables and chi-squared values at root node
     1  0.3346E+03   cat1
     2  0.2728E+03   aps1
     3  0.2430E+03   crea1
     :
    50  0.1052E+01   meta
    51  0.6357E+00   race

Size and CV mean cost and SE of subtrees:
 Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   1       59   3.085E-01   6.099E-03   7.419E-03   3.139E-01   8.732E-03
   2       58   3.085E-01   6.099E-03   7.419E-03   3.139E-01   8.732E-03
   :
  29       17   3.060E-01   6.085E-03   7.366E-03   3.078E-01   8.293E-03
  30**     16   3.050E-01   6.079E-03   7.354E-03   3.025E-01   8.394E-03
  31       12   3.085E-01   6.099E-03   7.055E-03   3.072E-01   7.716E-03
  32        9   3.083E-01   6.098E-03   6.862E-03   3.069E-01   7.082E-03
  33        6   3.158E-01   6.138E-03   6.474E-03   3.191E-01   1.028E-02
  34        3   3.425E-01   6.266E-03   7.205E-03   3.479E-01   1.195E-02
  35        1   3.808E-01   6.412E-03   2.782E-04   3.805E-01   4.832E-04
```

```
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 16 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    Total    Train    Predicted        Node    Split          Interacting
      label    cases    cases    class            cost    variables      variable
          1     5735     5735    NoRHC       3.808E-01 cat1
          2     1683     1683    RHC         4.599E-01 meanbp1 +pafi1
          4     1174     1174    RHC         3.705E-01 resp1 +surv2md1
         8T      229      229    RHC         1.790E-01 sod1 :wtkilo1
          9      945      945    RHC         4.169E-01 ninsclas
        18T      321      321    RHC         3.084E-01 -
         19      624      624    RHC         4.728E-01 dnr1
         38      554      554    RHC         4.495E-01 adld3p +edu
        76T      479      479    RHC         4.071E-01 -
        77T       75       75    NoRHC       2.800E-01 -
        39T       70       70    NoRHC       3.429E-01 -
         5T      509      509    NoRHC       3.340E-01 resp1 +adld3p
          3     4052     4052    NoRHC       3.147E-01 pafi1 +adld3p
          6     3330     3330    NoRHC       3.526E-01 aps1 +hema1
        12T     1092     1092    NoRHC       1.795E-01 pafi1 +scoma1
         13     2238     2238    NoRHC       4.370E-01 pafi1 +resp1
        26T      390      390    RHC         3.000E-01 cat2
         27     1848     1848    NoRHC       3.815E-01 aps1 +adld3p
        54T       74       74    NoRHC       2.432E-01 -
         55     1774     1774    NoRHC       3.873E-01 aps1 +wtkilo1
       110T      607      607    NoRHC       2.636E-01 card
        111     1167     1167    NoRHC       4.516E-01 meanbp1 +pafi1
        222      602      602    RHC         4.485E-01 paco21 +wtkilo1
       444T       94       94    RHC         2.340E-01 -
        445      508      508    RHC         4.882E-01 scoma1
        890      260      260    RHC         4.269E-01 bili1 +pot1
      1780T      155      155    RHC         3.226E-01 resp
      1781T      105      105    NoRHC       4.190E-01 -
       891T      248      248    NoRHC       4.476E-01 sex
       223T      565      565    NoRHC       3.451E-01 crea1 +pafi1
         7T      722      722    NoRHC       1.399E-01 card
```

```
Number of terminal nodes of final tree: 16
Total number of nodes of final tree: 31
Second best split variable (based on curvature test) at root node is aps1

Classification tree:
For categorical variable splits, values not in training data go to the right

Node 1: cat1 = "CHF", "MOSF w/Sepsis"
  Node 2: 0.24316737 * pafi1 + meanbp1 <= 153.28329 or NA
    Node 4: 48.127695 * surv2md1 + resp1 <= 43.437797 or NA
      Node 8: RHC
    Node 4: 48.127695 * surv2md1 + resp1 > 43.437797
      Node 9: ninsclas = "No insurance", "Private"
        Node 18: RHC
      Node 9: ninsclas /= "No insurance", "Private"
        Node 19: dnr1 = "No"
          Node 38: -23.826398 * edu + adld3p <= -282.91678 or NA
            Node 76: RHC
          Node 38: -23.826398 * edu + adld3p > -282.91678
            Node 77: NoRHC
        Node 19: dnr1 /= "No"
          Node 39: NoRHC
  Node 2: 0.24316737 * pafi1 + meanbp1 > 153.28329
    Node 5: NoRHC
Node 1: cat1 /= "CHF", "MOSF w/Sepsis"
  Node 3: 11.508773 * adld3p + pafi1 <= 149.35252 or NA
    Node 6: -1.3120163 * hema1 + aps1 <= 0.84337055
      Node 12: NoRHC
    Node 6: -1.3120163 * hema1 + aps1 > 0.84337055 or NA
      Node 13: 4.0975611 * resp1 + pafi1 <= 207.99333
        Node 26: RHC
      Node 13: 4.0975611 * resp1 + pafi1 > 207.99333 or NA
        Node 27: -23.161068 * adld3p + aps1 <= 66.838932
          Node 54: NoRHC
        Node 27: -23.161068 * adld3p + aps1 > 66.838932 or NA
          Node 55: 1.0116045 * wtkilo1 + aps1 <= 121.69374 or NA
            Node 110: NoRHC
          Node 55: 1.0116045 * wtkilo1 + aps1 > 121.69374
            Node 111: 0.35358803 * pafi1 + meanbp1 <= 134.65949 or NA
              Node 222: -0.42185873 * wtkilo1 + paco21 <= -7.0243280
                Node 444: RHC
              Node 222: -0.42185873 * wtkilo1 + paco21 > -7.0243280 or NA
                Node 445: scoma1 <= 4.5000000
                  Node 890: 5.8542561 * pot1 + bili1 <= 25.404949
                    Node 1780: RHC
                  Node 890: 5.8542561 * pot1 + bili1 > 25.404949 or NA
```

```
                     Node 1781: NoRHC
                 Node 445: scoma1 > 4.5000000 or NA
                   Node 891: NoRHC
             Node 111: 0.35358803 * pafi1 + meanbp1 > 134.65949
               Node 223: NoRHC
   Node 3: 11.508773 * adld3p + pafi1 > 149.35252
     Node 7: NoRHC

 ***************************************************************

 Predictor means below are means of cases with no missing values.

Node 1: Intermediate node
 A case goes into Node 2 if cat1 = "CHF", "MOSF w/Sepsis"
 cat1 mode = "ARF"
 Class      Number   Posterior
 NoRHC        3551  0.6192E+00
 RHC          2184  0.3808E+00
 Number of training cases misclassified = 2184
 Predicted class is NoRHC
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if 0.24316737 * pafi1 + meanbp1 <= 153.28329
 Linear combination mean = 133.36641
 Class      Number   Posterior
 NoRHC         774  0.4599E+00
 RHC           909  0.5401E+00
 Number of training cases misclassified = 774
 Predicted class is RHC
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if 48.127695 * surv2md1 + resp1 <= 43.437797
 Linear combination mean = 57.487146
 Class      Number   Posterior
 NoRHC         435  0.3705E+00
 RHC           739  0.6295E+00
 Number of training cases misclassified = 435
 Predicted class is RHC
 ----------------------------
Node 8: Terminal node
 Class      Number   Posterior
 NoRHC          41  0.1790E+00
 RHC           188  0.8210E+00
 Number of training cases misclassified = 41
 Predicted class is RHC
 ----------------------------
```

```
Node 9: Intermediate node
 A case goes into Node 18 if ninsclas = "No insurance", "Private"
 ninsclas mode = "Private"
 Class      Number   Posterior
 NoRHC         394  0.4169E+00
 RHC           551  0.5831E+00
 Number of training cases misclassified = 394
 Predicted class is RHC
 ----------------------------
:
:
Node 223: Terminal node
 Class      Number   Posterior
 NoRHC         370  0.6549E+00
 RHC           195  0.3451E+00
 Number of training cases misclassified = 195
 Predicted class is NoRHC
 ----------------------------
Node 7: Terminal node
 Class      Number   Posterior
 NoRHC         621  0.8601E+00
 RHC           101  0.1399E+00
 Number of training cases misclassified = 101
 Predicted class is NoRHC
 ----------------------------

 Classification matrix for training sample:
 Predicted      True class
 class          NoRHC       RHC
 NoRHC           3027      1040
 RHC              524      1144
 Total           3551      2184

 Number of cases used for tree construction: 5735
 Number misclassified: 1564
 Resubstitution estimate of mean misclassification cost: 0.27271142

 Observed and fitted values are stored in linearfit.txt
 LaTeX code for tree is in linear.tex
 R code is stored in linearpred.r
```

The LaTeX tree is shown in Figure 2, where each node that is split on a pair of ordinal variables is painted light gray. For example, node 2 is split on variables

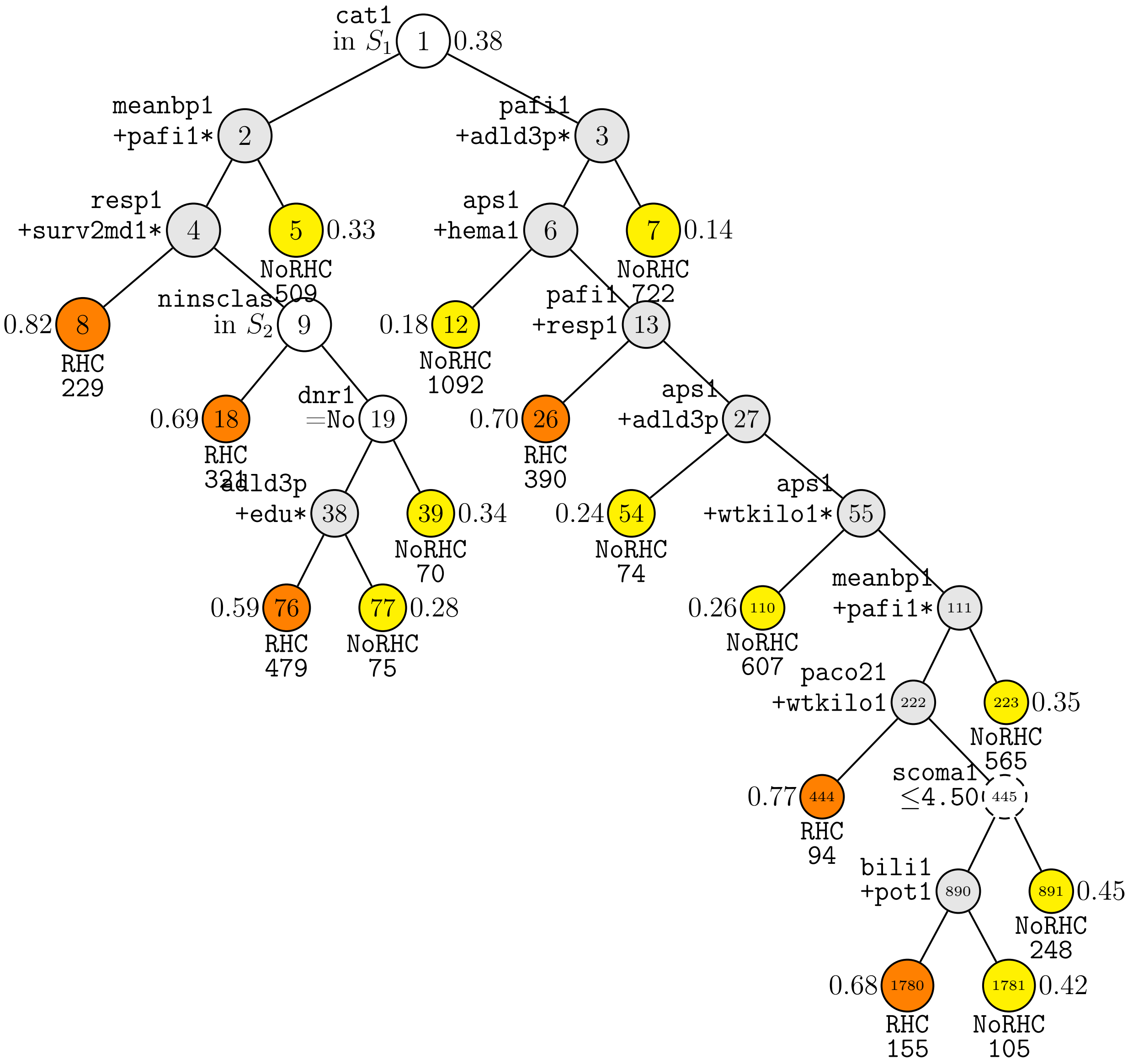

Figure 2: GUIDE v.38.0 0.25-SE classification tree for predicting swang1 using linear split priority, estimated priors and unit misclassification costs. An asterisk at a bivariate split indicates that missing values in either variable go to the left node. Set S_1 = {CHF, MOSF w/Sepsis}. Set S_2 = {No insurance, Private}. Circles with dashed lines are nodes with no significant split variables. Intermediate nodes in lightgray indicate linear splits. Predicted classes and sample sizes printed below terminal nodes; class sample proportion for swang1 = RHC beside nodes.

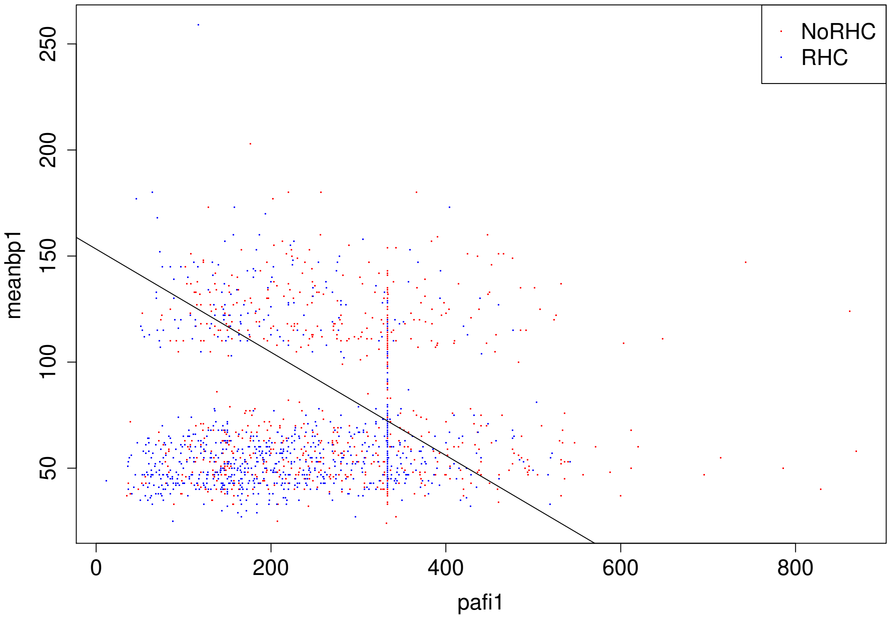

Figure 3: Plot of `meanbp1` vs `pafi1` for data and split in node 2 of tree in Figure 2

`meanbp1` and `pafi1`, with observations going left if and only if

$$0.24316737 \times \texttt{pafi1} + \texttt{meanbp1} \leq 153.28329.$$

The asterisk beside the node indicates that observations with missing values in either of the split variables go left. A plot of the data in this node is shown in Figure 3. The R code for making the plot is below. It reads `linearfit.txt` to extract the observations in the node.

4.2.4 R code for plot

```
z0 <- read.table("rhcdata.txt",header=TRUE)
z1 <- read.table("linearfit.txt",header=TRUE)
gp <- z1$node == 5 | z1$node == 8 | z1$node == 18 | z1$node == 39 |
      z1$node == 76 | z1$node == 77
x <- z0$pafi1[gp]
y <- z0$meanbp1[gp]
leg.txt <- c("NoRHC","RHC")
leg.col <- c("red","blue")
leg.pch <- c(1,4)
```

```
plot(x,y,xlab="pafi1",ylab="meanbp1",type="n")
g1 <- z0$swang1[gp] == "NoRHC"
points(x[g1],y[g1],pch=leg.pch[1],col=leg.col[1])
points(x[!g1],y[!g1],pch=leg.pch[2],col=leg.col[2])
abline(c(161.61473,-0.26651164))
legend("topright",legend=leg.txt,col=leg.col,pch=leg.pch,cex=1.5)
```

4.3 Kernel discriminant models

Another way to reduce the size of a classification tree is to fit a kernel discriminant model in each node.

4.3.1 Input file generation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: ker2.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: ker2.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
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
Input 1 for univariate, 2 for bivariate preference ([1:2], <cr>=2):
Input 1 for interaction tests, 2 to skip them ([1:2], <cr>=1):
Input 0 to specify tree with fixed no. of nodes, 1 to prune by CV,
      2 by test sample, 3 for no pruning ([0:3], <cr>=1):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: rhcdsc1.txt
Reading data description file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
20 N variables changed to S
Dependent variable is swang1
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
```

```
Missing values found among non-categorical variables
Number of classes: 2
Assigning integer codes to values of 30 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Class  #Cases     Proportion
NoRHC    3551     0.61918047
RHC      2184     0.38081953
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157       10        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       30        0
Number of cases used for training: 5735
Number of split variables: 53
Number of 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>=0.25):
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: 15
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 57
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): ker2.tex
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 sample sizes, 2 for sample proportions,
      3 for posterior probs, 4 for nothing
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: ker2.fit
Input rank of top variable to split root node ([1:50], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < ker2.in
```

4.3.2 Contents of ker2.out

```
Classification tree
Pruning by cross-validation
Data description file: rhcdsc1.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
23 N variables changed to S
D variable is swang1
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 2
Training sample class proportions of D variable swang1:
Class  #Cases     Proportion
NoRHC    3551     0.61918047
RHC      2184     0.38081953

Summary information for training sample of size 5735
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
      2  cat1       c                                  9
      3  cat2       c                                  6     4535
      4  ca         c                                  3
      :
     44  ph1        s    6.579        7.770
     45  swang1     d                                  2
     46  wtkilo1    s    19.50        244.0                   515
     47  dnr1       c                                  2
     48  ninsclas   c                                  6
     49  resp       c                                  2
     50  card       c                                  2
     51  neuro      c                                  2
     52  gastr      c                                  2
     53  renal      c                                  2
     54  meta       c                                  2
     55  hema       c                                  2
     56  seps       c                                  2
     57  trauma     c                                  2
     58  ortho      c                                  2
     59  adld3p     s    0.000        7.000                  4296
     60  urin1      s    0.000        9000.                  3028
     61  race       c                                  3
     62  income     c                                  4

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157       10        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       30        0
Number of cases used for training: 5735
Number of split variables: 53
Number of cases excluded due to 0 weight or missing D: 0

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Kernel density node models
Bivariate preference
Estimated priors
Unit misclassification costs
Bivariate split highest priority
Interaction splits 2nd priority; no linear splits
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 15
Minimum node sample size: 57
Non-univariate split at root node
Size and CV mean cost and SE of subtrees:
 Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   1       76   3.170E-01   6.144E-03   7.391E-03   3.206E-01   1.024E-02
   2       75   3.170E-01   6.144E-03   7.391E-03   3.206E-01   1.024E-02
```

```
    :
   46++      9   3.053E-01   6.081E-03   5.101E-03   3.049E-01   4.787E-03
   47**      7   3.039E-01   6.074E-03   5.098E-03   3.092E-01   7.207E-03
   48        6   3.107E-01   6.111E-03   4.164E-03   3.121E-01   4.682E-03
   49        5   3.180E-01   6.150E-03   5.979E-03   3.145E-01   8.560E-03
   50        4   3.229E-01   6.175E-03   4.475E-03   3.194E-01   6.704E-03
   51        3   3.236E-01   6.178E-03   4.577E-03   3.211E-01   7.707E-03
   52        2   3.316E-01   6.217E-03   6.964E-03   3.235E-01   1.044E-02
   53        1   3.688E-01   6.371E-03   2.637E-03   3.670E-01   2.864E-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 9 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    Total    Train    Predicted        Node    Split variable followed by
       label    cases    cases    class            cost    (+)fit variable(s)
           1     5735     5735    NoRHC       3.643E-01 cat1 +cat1 +pafi1
           2     1683     1683    RHC         4.225E-01 adld3p +adld3p +pafi1
           4     1183     1183    RHC         3.567E-01 wtkilo1 +wtkilo1 +pafi1
          8T      452      452    NoRHC       3.540E-01 pafi1 +pafi1 +hema1
          9T      731      731    RHC         3.010E-01 pafi1 +pafi1 +meanbp1
           5      500      500    NoRHC       4.100E-01 card +card +meanbp1
          10      345      345    NoRHC       3.333E-01 pot1 +pot1 +meanbp1
         20T      181      181    RHC         2.873E-01 meanbp1 +meanbp1 +resp1
         21T      164      164    NoRHC       2.500E-01 meanbp1 +meanbp1 +edu
         11T      155      155    NoRHC       3.677E-01 resp1 +resp1
           3     4052     4052    NoRHC       2.850E-01 pafi1 +pafi1 +crea1
          6T     1281     1281    NoRHC       3.599E-01 aps1 +aps1 +resp1
          7T     2771     2771    NoRHC       2.324E-01 meanbp1 +meanbp1 +crea1

 Number of terminal nodes of final tree: 7
 Total number of nodes of final tree: 13
 Second best split variable (based on interaction test) at root node is pafi1

 Classification tree:
```

```
 For categorical variable splits, values not in training data go to the right

 Node 1: cat1 = "CHF", "MOSF w/Sepsis"
   Node 2: adld3p = NA
     Node 4: wtkilo1 <= 70.249970
       Node 8: Mean cost = 0.35398230
     Node 4: wtkilo1 > 70.249970 or NA
       Node 9: Mean cost = 0.30095759
   Node 2: adld3p /= NA
     Node 5: card = "Yes"
       Node 10: pot1 <= 3.9499510
         Node 20: Mean cost = 0.28729282
       Node 10: pot1 > 3.9499510 or NA
         Node 21: Mean cost = 0.25000000
     Node 5: card /= "Yes"
       Node 11: Mean cost = 0.36774194
 Node 1: cat1 /= "CHF", "MOSF w/Sepsis"
   Node 3: pafi1 <= 141.85938
     Node 6: Mean cost = 0.35987510
   Node 3: pafi1 > 141.85938 or NA
     Node 7: Mean cost = 0.23240707

 ***************************************************************

 Predictor means below are means of cases with no missing values.

Node 1: Intermediate node
 A case goes into Node 2 if cat1 = "CHF", "MOSF w/Sepsis"
 cat1 mode = ARF
 pafi1 mean = 222.27371
                                 Bandwidth
 Class      Number   Posterior   cat1   pafi1
 NoRHC        3551  0.6192E+00               1.4868E-02
 RHC          2184  0.3808E+00               1.2981E-02
 Number of training cases misclassified = 2089
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if adld3p = NA
 adld3p mean = 1.2340000
 pafi1 mean = 249.20858
                                     Bandwidth
 Class      Number   Posterior   adld3p   pafi1   Correlation
 NoRHC         774  0.4599E+00   1.1959E+00   7.6307E+01     0.0944
 RHC           909  0.5401E+00   6.3364E-01   6.8628E+01     0.0222
 Number of training cases misclassified = 711
 If node model is inapplicable due to missing values, predicted class is "RHC"
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if wtkilo1 <= 70.249970
 wtkilo1 mean = 77.015038
 pafi1 mean = 231.38524
                                     Bandwidth
 Class      Number   Posterior   wtkilo1   pafi1   Correlation
 NoRHC         488  0.4125E+00   1.3035E+01   9.4062E+01    -0.1043
 RHC           695  0.5875E+00   1.2650E+01   7.1161E+01    -0.0544
 Number of training cases misclassified = 422
 If node model is inapplicable due to missing values, predicted class is "RHC"
 ----------------------------
Node 8: Terminal node
 pafi1 mean = 244.88658
 hema1 mean = 30.163116
                                     Bandwidth
 Class      Number   Posterior   pafi1   hema1   Correlation
 NoRHC         238  0.5265E+00   1.1248E+02   5.8918E+00    -0.1432
 RHC           214  0.4735E+00   9.2951E+01   3.9603E+00     0.0123
 ----------------------------
Node 9: Terminal node
 pafi1 mean = 223.03694
 meanbp1 mean = 70.605663
                                     Bandwidth
 Class      Number   Posterior   pafi1   meanbp1   Correlation
 NoRHC         250  0.3420E+00   9.5522E+01   2.9541E+01     0.1432
 RHC           481  0.6580E+00   7.5520E+01   1.1345E+01    -0.0287
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if card = "Yes"
 card mode = Yes
 meanbp1 mean = 78.048290
                                 Bandwidth
 Class      Number   Posterior   card   meanbp1
 NoRHC         286  0.5720E+00                2.9763E-02
 RHC           214  0.4280E+00                5.1896E-02
 Number of training cases misclassified = 205
 If node model is inapplicable due to missing values, predicted class is "RHC"
 ----------------------------
Node 10: Intermediate node
 A case goes into Node 20 if pot1 <= 3.9499510
 pot1 mean = 4.1646597
 meanbp1 mean = 80.576023
                                     Bandwidth
 Class      Number   Posterior   pot1   meanbp1   Correlation
 NoRHC        188  0.5449E+00   7.8030E-01   2.9193E+01    -0.1243
 RHC          157  0.4551E+00   6.0649E-01   1.3535E+01     0.0534
 Number of training cases misclassified = 115
 If node model is inapplicable due to missing values, predicted class is "RHC"
 ----------------------------
Node 20: Terminal node
 meanbp1 mean = 82.834254
 resp1 mean = 26.088889
                                     Bandwidth
 Class      Number   Posterior   meanbp1   resp1   Correlation
 NoRHC         84  0.4641E+00   3.2167E+01   6.5093E+00     0.0640
 RHC           97  0.5359E+00   1.4846E+01   8.9075E+00    -0.0159
 ----------------------------
Node 21: Terminal node
 meanbp1 mean = 78.037267
 edu mean = 11.300223
                                     Bandwidth
 Class      Number   Posterior   meanbp1   edu   Correlation
 NoRHC        104  0.6341E+00   3.3514E+01   2.1961E+00     0.0705
 RHC           60  0.3659E+00   1.5686E+01   3.2686E+00    -0.0921
 ----------------------------
Node 11: Terminal node
 resp1 mean = 29.032258
                                 Bandwidth
 Class      Number   Posterior   resp1
 NoRHC         98  0.6323E+00   9.2596E+00
 RHC           57  0.3677E+00   1.5413E+01
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if pafi1 <= 141.85938
 pafi1 mean = 211.08630
 crea1 mean = 1.8973326
                                     Bandwidth
 Class      Number   Posterior   pafi1   crea1   Correlation
 NoRHC       2777  0.6853E+00   5.7260E+01   3.7948E-01     0.0483
 RHC         1275  0.3147E+00   5.6018E+01   7.0942E-01     0.0733
 Number of training cases misclassified = 1155
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------
Node 6: Terminal node
 aps1 mean = 60.373927
 resp1 mean = 30.854487
                                     Bandwidth
 Class      Number   Posterior   aps1   resp1   Correlation
 NoRHC        661  0.5160E+00   1.1125E+01   8.1589E+00     0.3789
 RHC          620  0.4840E+00   1.2805E+01   9.8982E+00     0.3688
 ----------------------------
Node 7: Terminal node
 meanbp1 mean = 85.416758
 crea1 mean = 1.8756021
                                     Bandwidth
 Class      Number   Posterior   meanbp1    crea1   Correlation
 NoRHC        2116  0.7636E+00   2.0881E+01   4.0068E-01    -0.0610
 RHC           655  0.2364E+00   2.3948E+01   8.6122E-01    -0.0970
 ----------------------------

 Classification matrix for training sample:
 Predicted      True class
 class          NoRHC       RHC
 NoRHC           3004      1088
 RHC              547      1096
 Total           3551      2184

 Number of cases used for tree construction: 5735
 Number misclassified: 1635
 Resubstitution estimate of mean misclassification cost: 0.28509154

 Observed and fitted values are stored in ker2.fit
 LaTeX code for tree is in ker2.tex
```

The kernel discriminant tree is shown in Figure 4. The row with two asterisks (**) in the output file `ker2.out` shows that the tree has 6 terminal nodes and a cross-validation estimate of misclassification cost of 0.3165. Unlike the default and linear-split trees, the class of each observation in a terminal node is predicted based on kernel discrimination and therefore is not constant within the node. The file `ker2.fit` contains the terminal node number, estimated posteriors class probabilities, and observed and predicted class of each observation. Following are the first 5 lines.

```
  train  node "P(NoRHC)"  "P(RHC)"  observed  predicted
     y    6    0.47392    0.52608   "NoRHC"   "RHC"
     y    8    0.45177    0.54823   "RHC"     "RHC"
     y    7    0.60626    0.39374   "RHC"     "NoRHC"
     y    7    0.77436    0.22564   "NoRHC"   "NoRHC"
     y    9    0.32030    0.67970   "RHC"     "RHC"
```

Figure 5 shows plots of the data and the predicted values in terminal node 8 of the tree in the space of variables `hema1` and `pafi1` selected by GUIDE (see the information for these terminal nodes in `ker2.out`). The R code for making the plot is below.

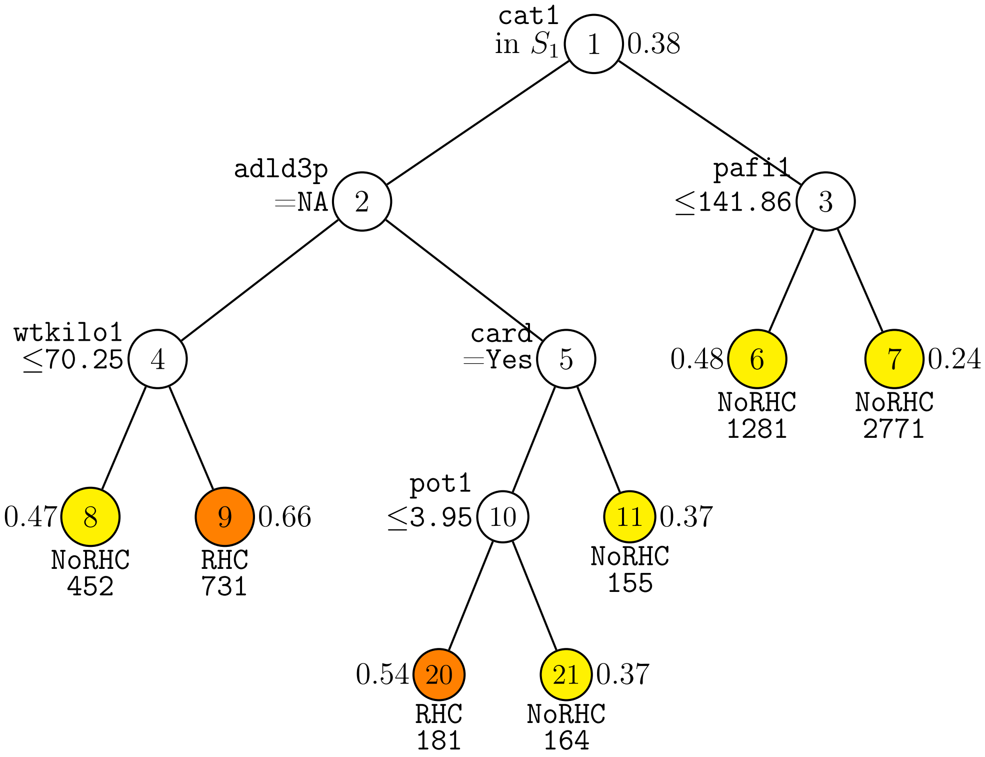

Figure 4: GUIDE v.38.0 0.25-SE classification tree for predicting `swang1` using bivariate kernel discriminant node models, estimated priors and unit misclassification costs. Tree constructed with 5735 observations. Maximum number of split levels is 15 and minimum node sample size is 57. At each split, an observation goes to the left branch if and only if the condition is satisfied. Set S_1 = {`CHF`, `MOSF w/Sepsis`}. Predicted classes and sample sizes printed below terminal nodes; class sample proportion for `swang1` = `RHC` beside nodes. Second best split variable (based on interaction test) at root node is `pafi1`.

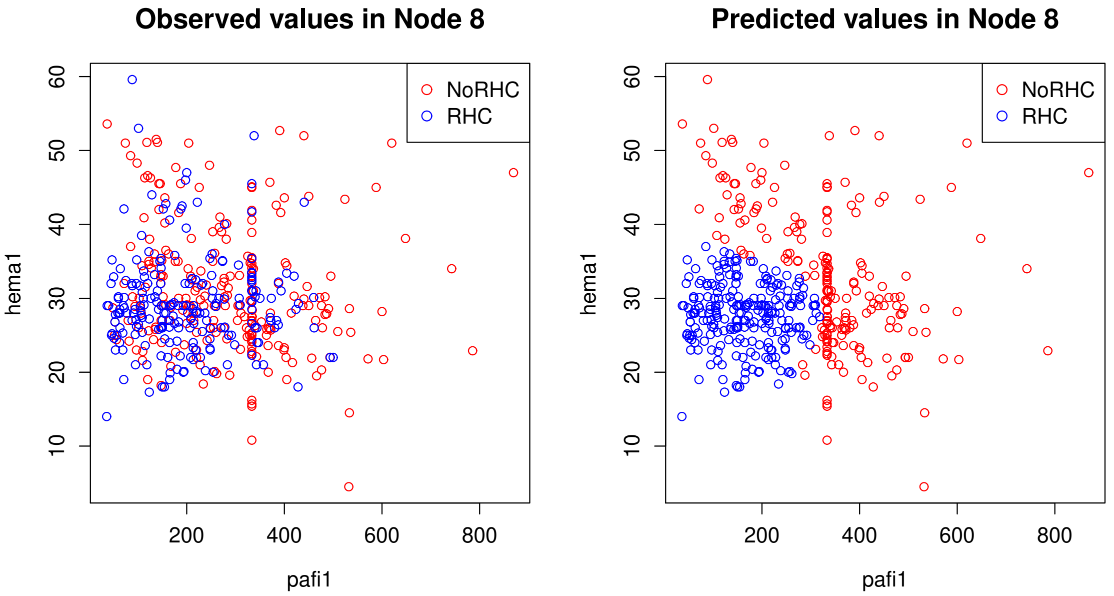

Figure 5: Plots of observed and predicted values for data in node 8 of tree in Figure 4

```
par(mfrow=c(1,2),pty="s",cex.lab=1.2,cex.axis=1.2,cex.main=1.5)
z1 <- read.table("ker2.fit",header=TRUE)
leg.txt <- c("NoRHC","RHC")
leg.col <- c("red","blue")
leg.pch <- rep(1,2)
gp <- z1$node == 8
x <- z0$pafi1[gp]
y <- z0$hema1[gp]
classv <- z0$swang1[gp]
plot(x,y,ylab="hema1",xlab="pafi1",type="n")
g1 <- classv == "NoRHC"
points(x[g1],y[g1],pch=leg.pch[1],col=leg.col[1])
points(x[!g1],y[!g1],pch=leg.pch[2],col=leg.col[2])
legend("topright",legend=leg.txt,col=leg.col,pch=leg.pch,cex=1.2)
title("Observed values in Node 8")
plot(x,y,ylab="hema1",xlab="pafi1",type="n")
pred <- z1$predicted[gp]
g1 <- pred == "NoRHC"
points(x[g1],y[g1],pch=leg.pch[1],col=leg.col[1])
points(x[!g1],y[!g1],pch=leg.pch[2],col=leg.col[2])
legend("topright",legend=leg.txt,col=leg.col,pch=leg.pch,cex=1.2)
title("Predicted values in Node 8")
```

4.4 Nearest-neighbor models

Yet another way to reduce the size of the default classification tree is to fit a nearest-neighbor model in each node. GUIDE can use univariate or bivariate nearest neigbors. We show this with bivariate neighbors here.

4.4.1 Input file generation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: nn2.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: nn2.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
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
Input 1 for univariate, 2 for bivariate preference ([1:2], <cr>=2):
Input 1 for interaction tests, 2 to skip them ([1:2], <cr>=1):
Input 0 to specify tree with fixed no. of nodes, 1 to prune by CV, 2 by test sample,
      3 for no pruning ([0:3], <cr>=1):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: rhcdsc1.txt
Reading data description file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
20 N variables changed to S
Dependent variable is swang1
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 2
Assigning integer codes to values of 30 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Re-checking data ...
```

```
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Class  #Cases     Proportion
NoRHC    3551     0.61918047
RHC      2184     0.38081953
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157       10        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       30        0
Number of cases used for training: 5735
Number of split variables: 53
Number of 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>=0.25):
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: 15
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 57
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): nn2.tex
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 sample sizes, 2 for sample proportions,
      3 for posterior probs, 4 for nothing
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: nn2.fit
Input rank of top variable to split root node ([1:50], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < nn2.in
```

4.4.2 Contents of nn2.out

```
Classification tree
Pruning by cross-validation
Data description file: rhcdsc1.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
23 N variables changed to S
D variable is swang1
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 2
Training sample class proportions of D variable swang1:
Class  #Cases     Proportion
NoRHC    3551     0.61918047
RHC      2184     0.38081953

Summary information for training sample of size 5735
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
      2  cat1       c                                   9
      3  cat2       c                                   6      4535
      4  ca         c                                   3
      :
     44  ph1        s    6.579        7.770
     45  swang1     d                                   2
     46  wtkilo1    s    19.50        244.0                     515
     47  dnr1       c                                   2
     48  ninsclas   c                                   6
     49  resp       c                                   2
     50  card       c                                   2
     51  neuro      c                                   2
     52  gastr      c                                  2
     53  renal      c                                  2
     54  meta       c                                  2
     55  hema       c                                  2
     56  seps       c                                  2
     57  trauma     c                                  2
     58  ortho      c                                  2
     59  adld3p     s    0.000        7.000                     4296
     60  urin1      s    0.000        9000.                     3028
     61  race       c                                  3
     62  income     c                                  4

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157       10        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       30        0
Number of cases used for training: 5735
Number of split variables: 53
Number of cases excluded due to 0 weight or missing D: 0

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nearest-neighbor node models
Bivariate preference
Estimated priors
Unit misclassification costs
Bivariate split highest priority
Interaction splits 2nd priority; no linear splits
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 15
Minimum node sample size: 57
Non-univariate split at root node
Size and CV mean cost and SE of subtrees:
 Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   1       76   3.151E-01   6.134E-03   6.384E-03   3.188E-01   1.012E-02
   2       75   3.151E-01   6.134E-03   6.384E-03   3.188E-01   1.012E-02
   :
  40++     34   3.149E-01   6.133E-03   5.175E-03   3.139E-01   7.342E-03
  41       32   3.163E-01   6.141E-03   6.259E-03   3.173E-01   9.416E-03
  42       31   3.163E-01   6.141E-03   6.111E-03   3.173E-01   8.898E-03
  43**     29   3.163E-01   6.141E-03   6.111E-03   3.173E-01   8.898E-03
  44       27   3.172E-01   6.145E-03   6.350E-03   3.200E-01   9.397E-03
  45       23   3.179E-01   6.149E-03   6.020E-03   3.200E-01   9.328E-03
  46       17   3.193E-01   6.156E-03   5.574E-03   3.243E-01   8.883E-03
  47       16   3.187E-01   6.153E-03   5.883E-03   3.243E-01   8.883E-03
  48       15   3.189E-01   6.154E-03   5.949E-03   3.243E-01   8.909E-03
  49       14   3.184E-01   6.152E-03   5.997E-03   3.261E-01   8.891E-03
  50        9   3.184E-01   6.152E-03   5.997E-03   3.261E-01   8.891E-03
  51        7   3.173E-01   6.146E-03   4.736E-03   3.176E-01   7.308E-03
  52        5   3.250E-01   6.185E-03   6.166E-03   3.243E-01   1.047E-02
  53        1   3.439E-01   6.272E-03   4.168E-03   3.458E-01   7.691E-03

0-SE tree based on mean is marked with * and has 34 terminal nodes
0-SE tree based on median is marked with + and has 34 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.

Node cost is node misclassification cost divided by number of training cases
       Node    Total    Train    Predicted       Node    Split variable followed by
      label    cases    cases    class           cost    (+)fit variable(s)
          1     5735     5735    NoRHC      2.961E-01 cat1 +cat1 +pafi1
          2     1683     1683    RHC        4.029E-01 adld3p +adld3p +pafi1
          4     1183     1183    RHC        3.271E-01 wtkilo1 +wtkilo1 +pafi1
          8      452      452    NoRHC      2.942E-01 pafi1 +pafi1 +hema1
         16T     257      257    RHC        2.646E-01 hema1 +hema1 +ph1
         17      195      195    NoRHC      2.872E-01 age +age
         34T     137      137    NoRHC      3.139E-01 das2d3pc +das2d3pc
         35T      58       58    NoRHC      1.034E-01 -
          9      731      731    RHC        2.791E-01 pafi1 +pafi1 +meanbp1
         18      420      420    RHC        2.619E-01 sex +sex
         36      300      300    RHC        2.233E-01 resp1 +resp1 +edu
         72T      90       90    RHC        6.667E-02 -
         73      210      210    RHC        2.905E-01 edu +edu
        146T     138      138    RHC        2.826E-01 aps1 +aps1
        147T      72       72    RHC        1.806E-01 -
         37T     120      120    RHC        3.000E-01 crea1 +crea1
         19      311      311    RHC        2.990E-01 meanbp1 +meanbp1
         38      237      237    RHC        3.418E-01 resp1 +resp1
         76T      92       92    RHC        2.609E-01 -
         77T     145      145    RHC            3.103E-01 age +age
         39T      74       74    NoRHC          2.432E-01 -
          5      500      500    NoRHC          3.220E-01 card +card +meanbp1
         10      345      345    NoRHC          2.986E-01 pot1 +pot1 +meanbp1
         20T     181      181    RHC            2.597E-01 meanbp1 +meanbp1 +resp1
         21T     164      164    NoRHC          2.622E-01 meanbp1 +meanbp1 +edu
         11T     155      155    NoRHC          3.226E-01 resp1 +resp1
          3     4052     4052    NoRHC          2.848E-01 pafi1 +pafi1 +crea1
          6     1281     1281    NoRHC          3.052E-01 aps1 +aps1 +resp1
         12      855      855    NoRHC          4.234E-01 card +card +adld3p
         24T     272      272    RHC            3.088E-01 alb1 +alb1 +meanbp1
         25      583      583    NoRHC          3.585E-01 resp +resp
         50T     182      182    NoRHC          3.462E-01 edu +edu
         51T     401      401    NoRHC          2.693E-01 immunhx +immunhx +temp1
         13      426      426    RHC            3.427E-01 resp +resp +resp1
         26      224      224    RHC            3.080E-01 resp1 +resp1 +age
         52T     139      139    RHC            2.302E-01 ph1 +ph1
         53T      85       85    NoRHC          3.059E-01 -
         27      202      202    RHC            2.723E-01 paco21 +paco21
         54T      69       69    RHC            1.304E-01 -
         55T     133      133    RHC            2.857E-01 surv2md1 +surv2md1
          7     2771     2771    NoRHC          2.317E-01 meanbp1 +meanbp1 +crea1
         14     1456     1456    NoRHC          3.043E-01 adld3p +adld3p +crea1
         28     1095     1095    NoRHC          2.749E-01 wtkilo1 +wtkilo1 +aps1
         56T     316      316    NoRHC          1.677E-01 card +card +hema1
         57      779      779    NoRHC          3.389E-01 dementhx +dementhx +crea1
        114      695      695    NoRHC          3.367E-01 dnr1 +dnr1 +crea1
        228      617      617    NoRHC          2.966E-01 pafi1 +pafi1 +crea1
        456T     262      262    RHC            2.595E-01 cat2 +cat2 +crea1
        457      355      355    NoRHC          3.014E-01 paco21 +paco21 +crea1
        914      190      190    NoRHC          2.684E-01 ph1 +ph1 +crea1
       1828T     125      125    RHC            2.160E-01 crea1 +crea1 +pot1
       1829T      65       65    NoRHC          2.615E-01 -
        915T     165      165    NoRHC          2.667E-01 ph1 +ph1 +edu
        229T      78       78    NoRHC          2.692E-01 -
        115T      84       84    NoRHC          2.143E-01 -
         29T     361      361    NoRHC          1.856E-01 age +age +card
         15T    1315     1315    NoRHC          1.612E-01 hema1 +hema1 +card
Warning: tree very large, omitting node numbers in LaTeX file

Number of terminal nodes of final tree: 29
Total number of nodes of final tree: 57
Second best split variable (based on interaction test) at root node is pafi1

Classification tree:
For categorical variable splits, values not in training data go to the right
```

```
Node 1: cat1 = "CHF", "MOSF w/Sepsis"
  Node 2: adld3p = NA
    Node 4: wtkilo1 <= 70.249970
      Node 8: pafi1 <= 254.50000
        Node 16: Mean cost = 0.26459144
      Node 8: pafi1 > 254.50000 or NA
        Node 17: age <= 75.961460
          Node 34: Mean cost = 0.31386861
        Node 17: age > 75.961460 or NA
          Node 35: Mean cost = 0.10344828
    Node 4: wtkilo1 > 70.249970 or NA
      Node 9: pafi1 <= 227.75000
        Node 18: sex = "Male"
          Node 36: resp1 <= 17.000000 or NA
            Node 72: Mean cost = 0.66666667E-1
          Node 36: resp1 > 17.000000
            Node 73: edu <= 12.410785
              Node 146: Mean cost = 0.28260870
            Node 73: edu > 12.410785 or NA
              Node 147: Mean cost = 0.18055556
        Node 18: sex /= "Male"
          Node 37: Mean cost = 0.30000000
      Node 9: pafi1 > 227.75000 or NA
        Node 19: meanbp1 <= 106.50000 or NA
          Node 38: resp1 <= 25.500000 or NA
            Node 76: Mean cost = 0.26086957
          Node 38: resp1 > 25.500000
            Node 77: Mean cost = 0.31034483
        Node 19: meanbp1 > 106.50000
          Node 39: Mean cost = 0.24324324
  Node 2: adld3p /= NA
    Node 5: card = "Yes"
      Node 10: pot1 <= 3.9499510
        Node 20: Mean cost = 0.25966851
      Node 10: pot1 > 3.9499510 or NA
        Node 21: Mean cost = 0.26219512
    Node 5: card /= "Yes"
      Node 11: Mean cost = 0.32258065
Node 1: cat1 /= "CHF", "MOSF w/Sepsis"
  Node 3: pafi1 <= 141.85938
    Node 6: aps1 <= 66.500000
      Node 12: card = "Yes"
        Node 24: Mean cost = 0.30882353
      Node 12: card /= "Yes"
        Node 25: resp = "No"
```

```
           Node 50: Mean cost = 0.34615385
         Node 25: resp /= "No"
           Node 51: Mean cost = 0.26932668
     Node 6: aps1 > 66.500000 or NA
       Node 13: resp = "Yes"
         Node 26: resp1 <= 41.000000
           Node 52: Mean cost = 0.23021583
         Node 26: resp1 > 41.000000 or NA
           Node 53: Mean cost = 0.30588235
       Node 13: resp /= "Yes"
         Node 27: paco21 <= 31.500000
           Node 54: Mean cost = 0.13043478
         Node 27: paco21 > 31.500000 or NA
           Node 55: Mean cost = 0.28571429
   Node 3: pafi1 > 141.85938 or NA
     Node 7: meanbp1 <= 69.500000 or NA
       Node 14: adld3p = NA
         Node 28: wtkilo1 <= 57.399995 or NA
           Node 56: Mean cost = 0.16772152
         Node 28: wtkilo1 > 57.399995
           Node 57: dementhx = "0"
             Node 114: dnr1 = "No"
               Node 228: pafi1 <= 216.15625
                 Node 456: Mean cost = 0.25954198
               Node 228: pafi1 > 216.15625 or NA
                 Node 457: paco21 <= 36.500000
                   Node 914: ph1 <= 7.4648440
                     Node 1828: Mean cost = 0.21600000
                   Node 914: ph1 > 7.4648440 or NA
                     Node 1829: Mean cost = 0.26153846
                 Node 457: paco21 > 36.500000 or NA
                   Node 915: Mean cost = 0.26666667
             Node 114: dnr1 /= "No"
               Node 229: Mean cost = 0.26923077
           Node 57: dementhx /= "0"
             Node 115: Mean cost = 0.21428571
       Node 14: adld3p /= NA
         Node 29: Mean cost = 0.18559557
     Node 7: meanbp1 > 69.500000
       Node 15: Mean cost = 0.16121673

 ***************************************************************

 Predictor means below are means of cases with no missing values.

Node 1: Intermediate node
```

```
 A case goes into Node 2 if cat1 = "CHF", "MOSF w/Sepsis"
 Number of nearest neighbors = 9
 cat1 mode = ARF
 pafi1 mean = 222.27371
 Class      Number   Posterior
 NoRHC        3551  0.6192E+00
 RHC          2184  0.3808E+00
 Number of training cases misclassified = 1698
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if adld3p = NA
 Number of nearest neighbors = 8
 adld3p mean = 1.2340000 SD = 1.8633799
 pafi1 mean = 249.20858 SD = 104.96492
              correlation = 0.63530716E-1
 Class      Number   Posterior
 NoRHC         774  0.4599E+00
 RHC           909  0.5401E+00
 Number of training cases misclassified = 678
 If node model is inapplicable due to missing values, predicted class is "RHC"
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if wtkilo1 <= 70.249970
 Number of nearest neighbors = 8
 wtkilo1 mean = 77.015038 SD = 22.059655
 pafi1 mean = 231.38524 SD = 115.76460
              correlation = -0.75261308E-1
 Class      Number   Posterior
 NoRHC         488  0.4125E+00
 RHC           695  0.5875E+00
 Number of training cases misclassified = 387
 If node model is inapplicable due to missing values, predicted class is "RHC"
 ----------------------------
Node 8: Intermediate node
 A case goes into Node 16 if pafi1 <= 254.50000
 Number of nearest neighbors = 7
 pafi1 mean = 244.88658 SD = 127.32603
 hema1 mean = 30.163116 SD = 7.6481547
              correlation = -0.69577606E-1
 Class      Number   Posterior
 NoRHC         238  0.5265E+00
 RHC           214  0.4735E+00
 Number of training cases misclassified = 133
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------
```

```
Node 16: Terminal node
 Number of nearest neighbors = 6
 hema1 mean = 30.549003 SD = 7.5321117
 ph1 mean = 7.3749811 SD = 0.11946464
               correlation = 0.23498459E-2
 Class      Number   Posterior
 NoRHC         102  0.3969E+00
 RHC           155  0.6031E+00
 ----------------------------
Node 17: Intermediate node
 A case goes into Node 34 if age <= 75.961460
 Number of nearest neighbors = 6
 age mean = 63.982335
                                Fit variable
 Class      Number   Posterior  age
 NoRHC         136  0.6974E+00
 RHC            59  0.3026E+00
 Number of training cases misclassified = 56
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------
Node 34: Terminal node
 Number of nearest neighbors = 5
 das2d3pc mean = 19.203281
                                Fit variable
 Class      Number   Posterior  das2d3pc
 NoRHC          84  0.6131E+00
 RHC            53  0.3869E+00
 ----------------------------
Node 35: Terminal node
 Using maximum likelihood
 Class      Number   Posterior
 NoRHC          52  0.8966E+00
 RHC             6  0.1034E+00
 ----------------------------
:
:
Node 115: Terminal node
 Using maximum likelihood
 Class      Number   Posterior
 NoRHC          66  0.7857E+00
 RHC            18  0.2143E+00
 ----------------------------
Node 29: Terminal node
 Number of nearest neighbors = 6
 age mean = 62.145410
 card mode = No
 Class       Number   Posterior
 NoRHC          294  0.8144E+00
 RHC             67  0.1856E+00
 ----------------------------
Node 15: Terminal node
 Number of nearest neighbors = 8
 hema1 mean = 33.662565
 card mode = No
 Class       Number   Posterior
 NoRHC         1103  0.8388E+00
 RHC            212  0.1612E+00
 ----------------------------

 Classification matrix for training sample:
 Predicted       True class
 class           NoRHC       RHC
 NoRHC            3111       885
 RHC               440      1299
 Total            3551      2184

 Number of cases used for tree construction: 5735
 Number misclassified: 1325
 Resubstitution estimate of mean misclassification cost: 0.23103749

 Observed and fitted values are stored in nn2.fit
 LaTeX code for tree is in nn2.tex
```

The nearest-neighbor density tree is shown in Figure 6. It is a supertree of the kernel discriminant tree in Figure 4. The row with two asterisks (**) in the output file nn2.out shows that the tree has 29 terminal nodes and a cross-validation estimate of misclassification cost of 0.3163. Unlike the default and linear-split trees, the class of each observation in a terminal node is predicted based on the classes of its neighbors and therefore is not constant within the node. Figure 7 shows plots of the data and the predicted values in terminal node 16 (leftmost node) of the tree in the space of variables hema1 and ph1 selected by GUIDE (see the information for these terminal nodes in nn2.out).

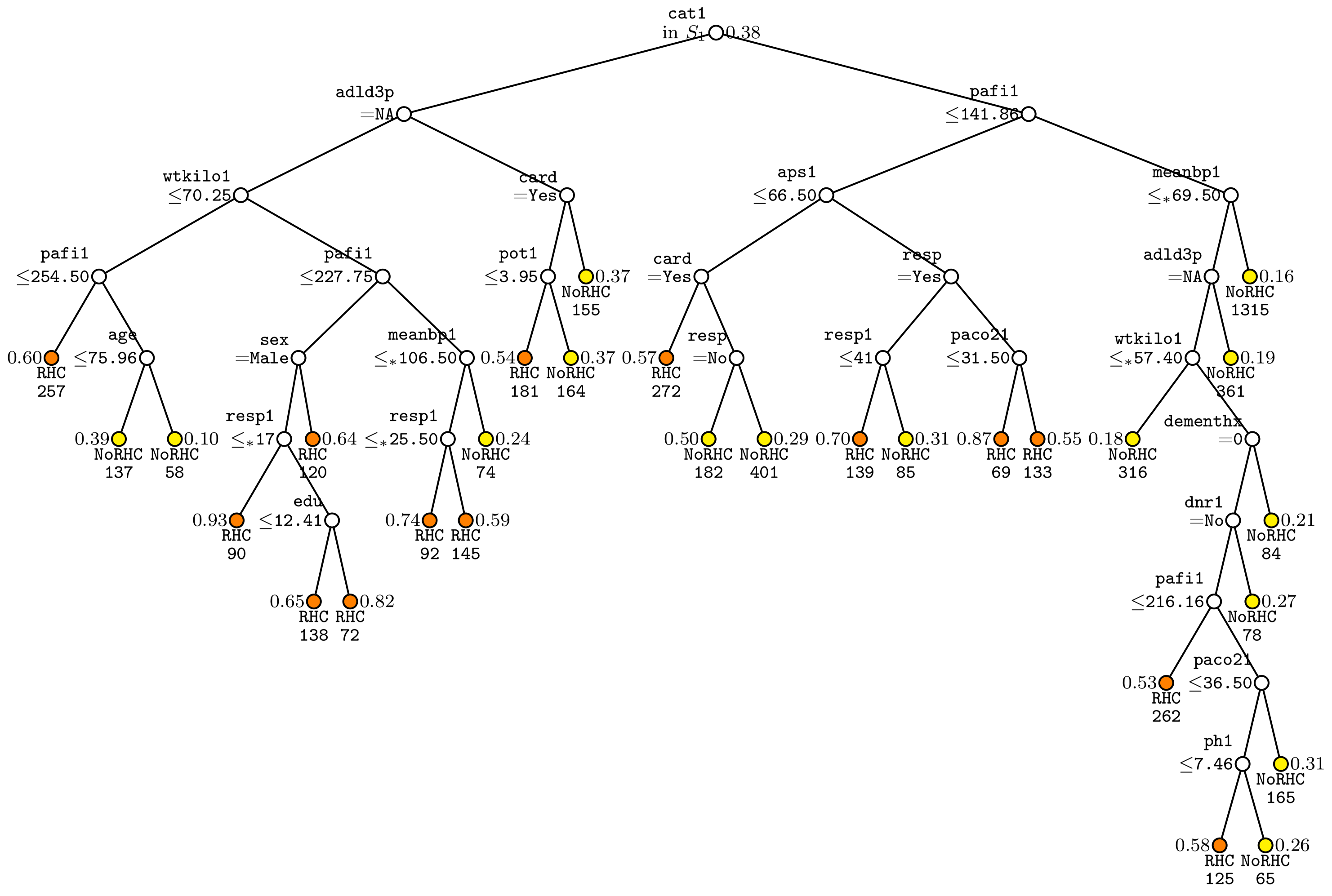

Figure 6: GUIDE v.38.0 0.25-SE classification tree for predicting `swang1` using bivariate nearest-neighbor node models, estimated priors and unit misclassification costs. Tree constructed with 5735 observations. Maximum number of split levels is 15 and minimum node sample size is 57. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. Set $S_1 = \{$`CHF`, `MOSF w/Sepsis`$\}$. Predicted classes and sample sizes printed below terminal nodes; class sample proportion for `swang1` = `RHC` beside nodes. Second best split variable (based on interaction test) at root node is `pafi1`.

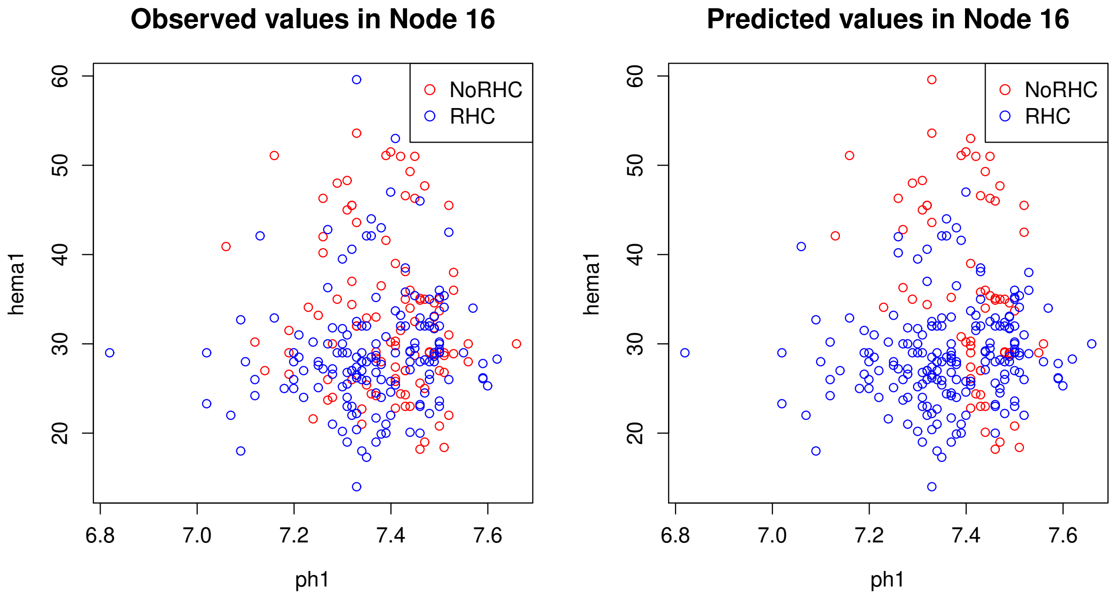

Figure 7: Plots of observed and predicted values for data in node 16 of tree in Figure 6

File `nn2.fit` gives the terminal node number and observed and predicted classes of each observation in the data file. Below are the first 5 rows. The first column is "`y`" (for yes) or "`n`" (for no) if the observation is used or not used to train the model. Unlike the kernel discriminant model, there are no estimated posterior class probabilities.

```
 train      node   observed    predicted
     y        24   "NoRHC"     "RHC"
     y        16   "RHC"       "RHC"
     y        56   "RHC"       "RHC"
     y        56   "NoRHC"     "NoRHC"
     y        77   "RHC"       "RHC"
```

5 Missing-value flag variables: CE data

Table 8: Codes and definitions of missing value flag variables

A	valid nonresponse: a response is not anticipated
B	invalid response
C	"don't know", refusal or other type of nonresponse
D	valid data value
T	topcoding applied to value

GUIDE can analyze data with more than one missing value code. Consider the data set from a 2013 Consumer Expenditure Survey of the Bureau of Labor Statistics (BLS) where there are 4693 observations and more than 600 variables. For each variable that has missing values, there is typically an associated *missing-value flag variable* that takes values A, B, C, D, and T (see Table 8 for definitions). The BLS uses the convention that all variable names are limited to 8 characters and the name of a missing-value flag variable is taken from the name of its associated variable with the addition of an underscore character or the replacement of a character with an underscore. For example, the missing-value flag variable associated with age of spouse, AGE2, is AGE2_ and the missing-value flag variable for BUILDING is BUIL_ING.

A T code for AGE2_ indicates that the value of AGE2 is "top-coded." Top-coding is a method used by the BLS to protect the privacy of the respondents in the top 3 percent of the data. The true values of the respondents in this group are replaced by their group mean. For example, below are the values of AGE2 and AGE2_ in the first 4 rows of the data:

```
   AGE2 AGE2_
1    87      T
2    NA      A
3    43      D
4    59      D
```

The first respondent has AGE2 $=$ 87 and AGE2_ $=$ T, which means that its actual AGE2 value is changed by BLS to the topcoded value of 87. The latter is the mean of the top 3 percent of AGE2 values in the data. The second respondent's AGE2 is missing (NA) and AGE2_ $=$ A, meaning that the nonresponse is valid (most likely due to the respondent not having a spouse). The 3rd and 4th respondents have valid AGE2 values of 43 and 59, as indicated by AGE2_ $=$ D. The data in the file cedata.txt give the responses of 4693 people for whom `INTRDVX_` $\neq$ `A`, where `INTRDVX` is the amount of interest and dividends. See `https://www.bls.gov/cex/pumd_doc.htm` for names of all the variables and Loh et al. (2019b, 2020) for an analysis of a similar dataset.

Missing-value flag variables are indicated by the letter "`m`" or "`M`" in the description file. To indicate to GUIDE to which variable is associated with each `M` variable, each `M` variable must follow immediately a `B`, `C`, `N`, `P`, or `S` variable in the description file. For example, the following lines from the file `ceclass.dsc` show that `DIRACC_` is the missing-value flag variable for `C` variable `DIRACC`, `AGE_REF_` is the missing-value flag for `N` variable `AGE_REF`, etc. The 21st variable `BLS_URBN` is an `N` variable that has no missing-value flag variable.

```
1 DIRACC C
2 DIRACC_ M
3 AGE_REF N
4 AGE_REF_ M
5 AGE2 N
6 AGE2_ M
7 AS_COMP1 N
8 AS_C_MP1 M
9 AS_COMP2 N
10 AS_C_MP2 M
11 AS_COMP3 N
12 AS_C_MP3 M
13 AS_COMP4 N
14 AS_C_MP4 M
15 AS_COMP5 N
16 AS_C_MP5 M
17 BATHRMQ N
18 BATHRMQ_ M
19 BEDROOMQ N
20 BEDR_OMQ M
21 BLS_URBN N
22 BUILDING C
23 BUIL_ING M
```

A split on an `N`, `P`, or `S` variable that has an associated missing-value flag variable can take several forms. For example, a split on `RETSURVX` (retirement, survivor, or disability pensions in past 12 months) with flag variable `RETS_RVX` can take 7 forms:

1. `RETS_RVX = A` (only `A` flag values go left)

Table 9: Some variable names and definitions in CE data

Name	Definition
AGE_REF	Age of reference person
AGE2	Age of spouse
CUTENURE	Housing tenure
ELCTRCCQ	Electricity this quarter
EMOTRVHC	Outlays for motored recreational vehicles this quarter
EMRTPNOP	Mortgage principal outlays last quarter for owned home
EOTHLODP	Outlays for other lodging last quarter
ETOTALP	Total outlays last quarter
FEDRFNDX	Federal income tax refund to all CU members
FEDR_NDX	Flag variable for FEDRFNDX
FEDTAXX	Amount Federal income tax paid in past 12 mos.
FEDTAXX_	Flag variable for FEDTAXX
FFTAXOWE	Estimated Federal tax liabilities for entire CU
FINCATAX	CU income after taxes in past 12 months
FINCBTAX	CU income before taxes in past 12 months
FRRETIRX	Social security and railroad retirement income
FJSSDEDX	Amount contributed to Social Security by all CU members past 12 mos.
FSALARYX	Wage and salary income of all members past 12 mos.
FSTAXOWE	Estimated state tax owed
HLFBATHQ	How many half bathrooms are there in this unit?
HEALTHCQ	Health care this quarter
HEALTHPQ	Health care last quarter
HIGH_EDU	Highest level of education
INC_RANK	Weighted percent income ranking of CU
INCLASS	Income class of CU based on income before taxes
INCLASS2	Income class based on INC_RANK
INC_HRS1	Number hours worked per week by reference person
INCNONW1	Reason for not working during past 12 months
INCN_NW1	Flag variable for INCNONW1
INCNONW2	Reason spouse did not work during past 12 months
INCN_NW2	Flag variable for INCNONW2
INCOMEY1	Employer paying most earnings in past 12 months
INCOMEY2	Employer from which spouse received most earnings in past 12 months

Table 10: Some variable names and definitions in CE data (cont'd.)

Name	Definition
LIQUIDX	Total value of checking, savings, CD, etc., accounts
LIQUIDX_	Flag variable for LIQUIDX
MEDSUPCQ	Medical supplies this quarter
NO_EARNR	Number of earners
OTHLODPQ	quarterOther lodging last quarter
OCCUCOD1	Highest paid occupation last 12 months
OCCU_OD1	Flag variable for OCCUCOD1
PERINSPQ	Personal insurance and pensions past quarter
PERSOT64	Number of persons over 64 in CU
POV_PY	Is income below previous year's poverty threshold?
PROPTXCQ	Property taxes current quarter
PROPTXPQ	Property taxes last quarter
PSU	Primary sampling unit
RENTEQVX	Monthly rent if home rented today
RETSURVX	Retirement, survivor, disability pensions past 12 mos.
RETS_RVX	Flag variable for RETSURVX
SLOCTAXX	Total amount paid for state and local income taxes
SLOC_AXX	Flag variable for SLOCTAXX
SLRFUNDX	State and local income tax refund received by all CU members
SLRF_NDX	Flag variable for SLRFUNDX
SMLAPPCQ	Small appliances, miscellaneous housewares this quarter
STATE	State identifier
STOCKX	Value of directly-held stocks, bonds, mutual funds
STOCKX_	Flag variable for STOCKX
STOCKYRX	Median value of bracket range of STOCKX
STOCKX_	Flag variable for STOCKX
TEXTILPQ	Household textiles last quarter
TOBACCPQ	Tobacco and smoking supplies last quarter
TOTEXPPQ	Total expenditures last quarter
TOTTXPDX	Personal taxes paid by CU in past 12 months
TOTXEST	Estimated total taxes paid
TRANSCQ	Transportation this quarter
TVRDIOCQ	Televisions, radios, and sound equipment this quarter
UNISTRQ	How many housing units are in this structure?
UTILRNTC	Expenditures on rented vacation home utilities this quarter

Table 11: CHILDAGE codes

0	No children
1	All children less than 6
2	Oldest child between 6 and 11 and at least one child less than 6
3	All children between 6 and 11
4	Oldest child between 12 and 17 and at least one child less than 12
5	All children between 12 and 17
6	Oldest child greater than 17 and at least one child less than 17
7	All children greater than 17

2. RETS_RVX = C (only C flag values go left)

3. RETSURVX = NA (all missing values go left)

4. RETSURVX $\leq c$

5. RETSURVX $\leq_* c$ (the symbol "$\leq_*$" means "$\leq$ or is missing")

6. RETSURVX $\leq c$ or RETS_RVX = A

7. RETSURVX $\leq c$ or RETS_RVX = C

Similarly, a split on a C variable such as INCNONW2 that has missing-value flag variable INCN_NW2 can take these forms (see Figure 15):

1. INCNONW2 in S

2. INCNONW2 = NA

3. INCNONW2in S or INCN_NW2 in S^*

The M descriptor can also be used if a predictor variable takes values that are partly ordinal and partly categorical. For example, Table 11 shows the value codes of CHILDAGE in the data. Although codes 1-7 are ordinal, it is not obvious that code 0 should be treated as less than 1, because then every split on CHILDAGE of the form "CHILDAGE $\leq c$" would necessarily send observations with CHILDAGE = 0 to the left subnode. To allow splits of the form "1 $\leq$ CHILDAGE $\leq c$" (which sends CHILDAGE = 0 to the right subnode), we recode CHILDAGE = 0 to CHILDAGE = NA and create a missing-value flag variable CHIL_AGE that takes value 0 if CHILDAGE = 0, 1 if CHILDAGE = NA, and D otherwise; see Table 12. This allows 5 types of splits:

Table 12: Original and new CHILDAGE variables

Original	New	
CHILDAGE	CHILDAGE	CHIL_AGE
0	NA	0
1	1	D
2	2	D
3	3	D
4	4	D
5	5	D
6	6	D
7	7	D
NA	NA	1

1. New CHILDAGE = NA (equivalent to original CHILDAGE = 0 or NA)

2. New CHILDAGE $\leq c$ (equivalent to original CHILDAGE $= 1, 2, \ldots, c$)

3. New CHILDAGE $\leq_* c$ (equivalent to original CHILDAGE $= 0, 1, \ldots, c$)

4. CHIL_AGE = 0 (equivalent to original CHILDAGE $= 0$)

5. CHIL_AGE = 1 (equivalent to original CHILDAGE = NA)

5.1 Classification tree

Splits on M variables can be demonstrated by fitting a classification tree to predict INTRDVX_, which takes values C (37.7%), D (60.5%), and T (1.8%). The description file is ceclass.dsc and the data file is cedata.txt.

5.1.1 Input file generation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: ceclass.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: ceclass.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
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: ceclass.dsc
Reading data description file ...
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
422 N variables changed to S
D variable is INTRDVX_
Reading data file ...
Number of records in data file: 4693
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 3
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 42 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: S variable MISC2PQ is constant
Warning: S variable MISC2CQ is constant
Warning: S variable TCARTRKP is constant
Warning: S variable TCARTRKC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable VMISCHEP is constant
Warning: S variable VMISCHEC is constant
Warning: S variable ROTHRFLP is constant
Warning: S variable ROTHRFLC is constant
```

```
Smallest positive weight:  1.3507E+03
Largest positive weight:   7.0269E+04
Class  #Cases     Proportion
C        1771     0.37737055
D        2838     0.60473045
T          84     0.01789900
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693          0       4693       16        0        0      422
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      171        0       42        0
Number of cases used for training: 4693
Number of split variables: 464
Number of 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 to input 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):
Warning: All positive weights treated as 1
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): ceclass.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: ceclass.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input rank of top variable to split root node ([1:464], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < ceclass.in
```

5.1.2 Contents of output file

```
Classification tree
Pruning by cross-validation
Data description file: ceclass.dsc
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
422 N variables changed to S
D variable is INTRDVX_
Number of records in data file: 4693
Length of longest entry in data file: 11
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
```

```
Missing values found among non-categorical variables
Number of classes: 3
Warning: S variable MISC2PQ is constant
Warning: S variable MISC2CQ is constant
Warning: S variable TCARTRKP is constant
Warning: S variable TCARTRKC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable VMISCHEP is constant
Warning: S variable VMISCHEC is constant
Warning: S variable ROTHRFLP is constant
Warning: S variable ROTHRFLC is constant
Smallest and largest positive weights are  1.3507E+03 and 7.0269E+04
Training sample class proportions of D variable INTRDVX_:
Class  #Cases     Proportion
C        1771     0.37737055
D        2838     0.60473045
T          84     0.01789900

Summary information for training sample of size 4693
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
Levels of M variables are for missing values in associated variables
                                                 #Codes/
                                                 Levels/
 Column  Name           Minimum      Maximum    Periods   #Missing
      1  DIRACC     c                                  2      155
      2  DIRACC_    m                                  1
      3  AGE_REF    s    18.00        87.00
      4  AGE_REF_   m                                  0
      :
     50  FINLWT21   w    1351.       0.7027E+05
      :
    514  INTRDVX_   d                                  3
      :
    651  FSTAXOWE   s   -2505.       0.5991E+05
    652  FSTA_OWE   m                                  0
    653  ETOTA      s    1199.       0.2782E+06

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals    #X-var   #N-var   #F-var   #S-var
      4693          0       4693       16        0        0      422
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      171        0       42        0
```

```
Number of cases used for training: 4693
Number of split variables: 464
Number of cases excluded due to 0 weight or missing D: 0

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Warning: No interaction tests; too many predictor variables
Simple node models
Estimated priors
Unit misclassification costs
Warning: All positive weights treated as 1
Univariate split highest priority
No interaction splits
No linear splits
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 14
Minimum node sample size: 46
Top-ranked variables and chi-squared values at root node
     1  0.3454E+03   INCLASS2
     2  0.3424E+03   INC_RANK
     3  0.3222E+03   RESPSTAT
     :
   417  0.5888E-03   WOMSIXCQ
   418  0.7182E-04   STDNTYRB
   419  0.7182E-04   STDTYRBX

Size and CV mean cost and SE of subtrees:
 Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   1       75   3.060E-01   6.727E-03   6.920E-03   3.067E-01   7.374E-03
   2       74   3.060E-01   6.727E-03   6.920E-03   3.067E-01   7.374E-03
   :
  31       39   3.060E-01   6.727E-03   6.920E-03   3.067E-01   7.374E-03
  32+      32   3.041E-01   6.715E-03   7.273E-03   3.046E-01   8.009E-03
  33       28   3.045E-01   6.718E-03   7.628E-03   3.056E-01   9.185E-03
  34       25   3.039E-01   6.714E-03   7.476E-03   3.056E-01   9.445E-03
  35       20   3.041E-01   6.715E-03   7.415E-03   3.056E-01   9.172E-03
  36       17   3.045E-01   6.718E-03   7.715E-03   3.053E-01   1.080E-02
  37**     14   3.039E-01   6.714E-03   7.619E-03   3.053E-01   1.071E-02
  38       12   3.092E-01   6.746E-03   7.721E-03   3.120E-01   1.284E-02
  39       11   3.228E-01   6.825E-03   7.433E-03   3.280E-01   8.858E-03
  40        8   3.360E-01   6.895E-03   6.699E-03   3.412E-01   1.075E-02
  41        6   3.437E-01   6.933E-03   7.122E-03   3.461E-01   8.912E-03
  42        2   3.443E-01   6.936E-03   7.081E-03   3.489E-01   9.582E-03
  43        1   3.953E-01   7.137E-03   8.408E-03   4.036E-01   1.140E-02
```

```
0-SE tree based on mean is marked with * and has 14 terminal nodes
0-SE tree based on median is marked with + and has 32 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
* 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    Total    Train    Predicted        Node    Split          Interacting
      label    cases    cases    class            cost    variables      variable
          1     4693     4693    D           3.953E-01 INCLASS2
          2     4326     4326    D           3.588E-01 STATE
          4     2039     2039    D           4.586E-01 INCOMEY2
          8T      73       73    C           4.932E-01 -
          9     1966     1966    D           4.532E-01 PSU
         18      241      241    C           3.361E-01 ELCTRCCQ
         36      108      108    C           4.167E-01 UNISTRQ
         72T      61       61    D           4.262E-01 -
         73T      47       47    C           2.129E-01 -
         37T     133      133    C           2.707E-01 RETPENCQ
         19     1725     1725    D           4.232E-01 FEDTAXX
         38     1523     1523    D           3.940E-01 FEDRFNDX
         76      648      648    D           4.213E-01 RENTEQVX
        152T     468      468    D           3.397E-01 FINCBTAX
        153T     180      180    C           4.000E-01 IRAX
         77      875      875    D           3.737E-01 FEDRFNDX
        154T     111      111    C           3.064E-01 POPSIZE
        155T     764      764    D           3.272E-01 INCOMEY1
         39      202      202    C           4.406E-01 TOTTXPDX
         78T     152      152    C           3.224E-01 BUILT
         79T      50       50    D           4.400E-01 -
          5     2287     2287    D           2.698E-01 RETSURVX
         10T    1618     1618    D           2.608E-01 INCNONW1
         11      669      669    D           2.915E-01 RETSURVX
         22T      73       73    C           6.861E-02 -
         23T     596      596    D           2.131E-01 POPSIZE
```

```
          3T     367      367    C             1.745E-01 FINCBTAX

Number of terminal nodes of final tree: 14
Total number of nodes of final tree: 27
Second best split variable (based on curvature test) at root node is INC_RANK

Classification tree:
For categorical variable splits, values not in training data go to the right

Node 1: INCLASS2 <= 6.5000000
  Node 2: STATE = "10", "12", "15", "17", "22", "25", "26", "34", "36", "39",
           "42", "45", "47", "53", "55", "8"
    Node 4: INCOMEY2 = "5", "6"
      Node 8: C
    Node 4: INCOMEY2 /= "5", "6"
      Node 9: PSU = "1102", "1423"
        Node 18: ELCTRCCQ <= 5.0000000
          Node 36: UNISTRQ <= 3.5000000
            Node 72: D
          Node 36: UNISTRQ > 3.5000000 or NA
            Node 73: C
        Node 18: ELCTRCCQ > 5.0000000 or NA
          Node 37: C
      Node 9: PSU /= "1102", "1423"
        Node 19: FEDTAXX <= 3078.5000  or FEDTAXX = NA & FEDTAXX_ = "A"
          Node 38: FEDRFNDX = NA & FEDR_NDX = "A"
            Node 76: RENTEQVX <= 1731.0000 or NA
              Node 152: D
            Node 76: RENTEQVX > 1731.0000
              Node 153: C
          Node 38: not (FEDRFNDX = NA & FEDR_NDX = "A")
            Node 77: FEDRFNDX = NA
              Node 154: C
            Node 77: FEDRFNDX /= NA
              Node 155: D
        Node 19: not (FEDTAXX <= 3078.5000 or FEDTAXX = NA & FEDTAXX_ = "A")
          Node 39: TOTTXPDX <= 11911.500
            Node 78: C
          Node 39: TOTTXPDX > 11911.500 or NA
            Node 79: D
  Node 2: STATE /= "10", "12", "15", "17", "22", "25", "26", "34", "36", "39",
           "42", "45", "47", "53", "55", "8"
    Node 5: RETSURVX = NA & RETS_RVX = "A"
      Node 10: D
    Node 5: not (RETSURVX = NA & RETS_RVX = "A")
      Node 11: RETSURVX = NA
```

```
         Node 22: C
       Node 11: RETSURVX /= NA
         Node 23: D
 Node 1: INCLASS2 > 6.5000000 or NA
   Node 3: C

 ***************************************************************

 Predictor means below are weighted means of cases with no missing values.

Node 1: Intermediate node
 A case goes into Node 2 if INCLASS2 <= 6.5000000
 INCLASS2 mean = 4.5074794
 Class      Number   Posterior
 C            1771  0.3774E+00
 D            2838  0.6047E+00
 T              84  0.1790E-01
 Number of training cases misclassified = 1855
 Predicted class is D
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 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            1468  0.3393E+00
 D            2774  0.6412E+00
 T              84  0.1942E-01
 Number of training cases misclassified = 1552
 Predicted class is D
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if INCOMEY2 = "5", "6"
 INCO_EY2 mode = "A"
 Class      Number   Posterior
 C             889  0.4360E+00
 D            1104  0.5414E+00
 T              46  0.2256E-01
 Number of training cases misclassified = 935
 Predicted class is D
 ----------------------------
Node 8: Terminal node
 Class      Number   Posterior
 C              37  0.5068E+00
 D              29  0.3973E+00
 T               7  0.9589E-01
 Number of training cases misclassified = 36
 Predicted class is C
 ----------------------------
Node 9: Intermediate node
 A case goes into Node 18 if PSU = "1102", "1423"
 PSU mode = "NA"
 Class      Number   Posterior
 C             852  0.4334E+00
 D            1075  0.5468E+00
 T              39  0.1984E-01
 Number of training cases misclassified = 891
 Predicted class is D
 ----------------------------
Node 18: Intermediate node
 A case goes into Node 36 if ELCTRCCQ <= 5.0000000
 ELCTRCCQ mean = 101.99524
 Class      Number   Posterior
 C             160  0.6639E+00
 D              80  0.3320E+00
 T               1  0.4149E-02
 Number of training cases misclassified = 81
 Predicted class is C
 ----------------------------
Node 36: Intermediate node
 A case goes into Node 72 if UNISTRQ <= 3.5000000
 UNISTRQ mean = 4.5811036
 Class      Number   Posterior
 C              63  0.5833E+00
 D              45  0.4167E+00
 T               0  0.3813E-05
 Number of training cases misclassified = 45
 Predicted class is C
 ----------------------------
Node 72: Terminal node
 Class      Number   Posterior
 C              26  0.4262E+00
 D              35  0.5738E+00
 T               0  0.3813E-05
 Number of training cases misclassified = 26
 Predicted class is D
 ----------------------------
Node 73: Terminal node
 Class      Number   Posterior
 C              37  0.7871E+00
 D              10  0.2128E+00
 T               0  0.3813E-05
 Number of training cases misclassified = 10
 Predicted class is C
 ----------------------------
Node 37: Terminal node
 Class      Number   Posterior
 C              97  0.7293E+00
 D              35  0.2632E+00
 T               1  0.7519E-02
 Number of training cases misclassified = 36
 Predicted class is C
 ----------------------------
Node 19: Intermediate node
 A case goes into Node 38 if FEDTAXX <= 3078.5000 or FEDTAXX_ = "A"
 FEDTAXX mean = 6760.9819
 Class      Number   Posterior
 C             692  0.4012E+00
 D             995  0.5768E+00
 T              38  0.2203E-01
 Number of training cases misclassified = 730
 Predicted class is D
 ----------------------------
Node 38: Intermediate node
 A case goes into Node 76 if FEDRFNDX = NA & FEDR_NDX = "A"
 FEDRFNDX mean = 3080.9067
 Class      Number   Posterior
 C             579  0.3802E+00
 D             923  0.6060E+00
 T              21  0.1379E-01
 Number of training cases misclassified = 600
 Predicted class is D
 ----------------------------
Node 76: Intermediate node
 A case goes into Node 152 if RENTEQVX <= 1731.0000 or NA
 RENTEQVX mean = 1566.4820
 Class      Number   Posterior
 C             259  0.3997E+00
 D             375  0.5787E+00
 T              14  0.2160E-01
 Number of training cases misclassified = 273
 Predicted class is D
 ----------------------------
Node 152: Terminal node
 Class      Number   Posterior
 C             151  0.3226E+00
 D             309  0.6603E+00
 T               8  0.1709E-01
 Number of training cases misclassified = 159
 Predicted class is D
 ----------------------------
Node 153: Terminal node
 Class      Number   Posterior
 C             108  0.6000E+00
 D              66  0.3667E+00
 T               6  0.3333E-01
 Number of training cases misclassified = 72
 Predicted class is C
 ----------------------------
Node 77: Intermediate node
 A case goes into Node 154 if FEDRFNDX = NA
 FEDRFNDX mean = 3080.9067
 Class      Number   Posterior
 C             320  0.3657E+00
 D             548  0.6263E+00
 T               7  0.8000E-02
 Number of training cases misclassified = 327
 Predicted class is D
 ----------------------------
Node 154: Terminal node
 Class      Number   Posterior
 C              77  0.6936E+00
 D              34  0.3064E+00
 T               0  0.3813E-05
 Number of training cases misclassified = 34
 Predicted class is C
 ----------------------------
Node 155: Terminal node
 Class      Number   Posterior
 C             243  0.3181E+00
 D             514  0.6728E+00
 T               7  0.9162E-02
 Number of training cases misclassified = 250
 Predicted class is D
 ----------------------------
Node 39: Intermediate node
 A case goes into Node 78 if TOTTXPDX <= 11911.500
 TOTTXPDX mean = 13353.797
 Class      Number   Posterior
 C             113  0.5594E+00
 D              72  0.3564E+00
 T              17  0.8416E-01
 Number of training cases misclassified = 89
 Predicted class is C
 ----------------------------
Node 78: Terminal node
 Class      Number   Posterior
 C             103  0.6776E+00
 D              44  0.2895E+00
 T               5  0.3289E-01
 Number of training cases misclassified = 49
 Predicted class is C
 ----------------------------
Node 79: Terminal node
 Class      Number   Posterior
 C              10  0.2000E+00
 D              28  0.5600E+00
 T              12  0.2400E+00
 Number of training cases misclassified = 22
 Predicted class is D
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if RETSURVX = NA & RETS_RVX = "A"
 RETSURVX mean = 26778.499
 Class      Number   Posterior
 C             579  0.2532E+00
 D            1670  0.7302E+00
 T              38  0.1662E-01
 Number of training cases misclassified = 617
 Predicted class is D
 ----------------------------
Node 10: Terminal node
 Class      Number   Posterior
 C             394  0.2435E+00
 D            1196  0.7392E+00
 T              28  0.1731E-01
 Number of training cases misclassified = 422
 Predicted class is D
 ----------------------------
Node 11: Intermediate node
 A case goes into Node 22 if RETSURVX = NA
 RETSURVX mean = 26778.499
 Class      Number   Posterior
 C             185  0.2765E+00
 D             474  0.7085E+00
 T              10  0.1495E-01
 Number of training cases misclassified = 195
 Predicted class is D
 ----------------------------
Node 22: Terminal node
```

```
 Class      Number   Posterior
 C              68  0.9314E+00
 D               5  0.6861E-01
 T               0  0.3813E-05
 Number of training cases misclassified = 5
 Predicted class is C
 ----------------------------
Node 23: Terminal node
 Class      Number   Posterior
 C             117  0.1963E+00
 D             469  0.7869E+00
 T              10  0.1678E-01
 Number of training cases misclassified = 127
 Predicted class is D
 ----------------------------
Node 3: Terminal node
 Class      Number   Posterior
 C             303  0.8255E+00
 D              64  0.1745E+00
 T               0  0.3813E-05
 Number of training cases misclassified = 64
 Predicted class is C
 ----------------------------

 Classification matrix for training sample:
 Predicted      True class
 class              C         D         T
 C                830       287        19
 D                941      2551        65
 T                  0         0         0
 Total           1771      2838        84

 Number of cases used for tree construction: 4693
 Number misclassified: 1312
 Resubstitution estimate of mean misclassification cost: 0.27956531

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

Figure 8 shows the classification tree. Five different kinds of splits on missing values are exhibited in these intermediate nodes:

Node 1: Split on `N` variable `INCLASS2` $\leq$ 6.50 with all missing values going right

Nodes 5 and 38: Splits on `M` variables `RETS_RVX` and `FEDR_NDX`, respectively.

Nodes 11 and 77: Splits on missing values of N variables RETSURVX and FEDRFNDX, respectively.

Node 19: Split on N variable FEDTAXX $\leq$ 3078.5 or its M variable FEDTAXX_ $=$ A.

Node 76: Split on N variable RENTEQVX $\leq_*$ 1731 with all missing values going left.

Owing to the small number of cases of INTRDVX_ $=$ T, the tree has no terminal node that predicts this class. The top several lines of the file of fitted values ceclass.fit are given below. They show that the posterior probability of predicting class T is very low (see Section 4.1.4 for the calculation of the posterior probabilities).

```
train     node  observed predicted     "P(C)"        "P(D)"        "P(T)"
    y       10     "D"       "D"     0.24351E+00  0.73918E+00  0.17305E-01
    y      152     "D"       "D"     0.32265E+00  0.66026E+00  0.17094E-01
    y       10     "D"       "D"     0.24351E+00  0.73918E+00  0.17305E-01
    y       10     "D"       "D"     0.24351E+00  0.73918E+00  0.17305E-01
    y       23     "D"       "D"     0.19631E+00  0.78691E+00  0.16779E-01
    y       23     "D"       "D"     0.19631E+00  0.78691E+00  0.16779E-01
    y      154     "C"       "C"     0.69363E+00  0.30637E+00  0.38132E-05
    y      152     "D"       "D"     0.32265E+00  0.66026E+00  0.17094E-01
```

6 Priors and periodic variables: NHTSA data

Periodic variables that have a cyclic property, such as angular measurements, hour of day, day of week, and month of year, can be designated as P variables 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. This version of GUIDE does not allow P variables to have missing-value flag (M) variables.

The files nhtsadata.csv and nhtsaclass.dsc have P variables. The data are from National Highway Transportation Safety Administration (NHTSA) vehicle crash tests (www-nrd.nhtsa.dot.gov/database/veh/). Variable HIC (head injury criterion) is a measure of severity of head injury. Experts believe that HIC $>$ 999 is absolutely life threatening. For this illustration, we use the binary response variable HIC2, which equals 1 if HIC $>$ 999, and equals 0 otherwise. Table 13 gives the definitions of the variables appearing in the models below. The values of periodic variables in this example are measured clockwise starting with 0 in front. The contents of nhtsaclass.dsc are partially reproduced below.

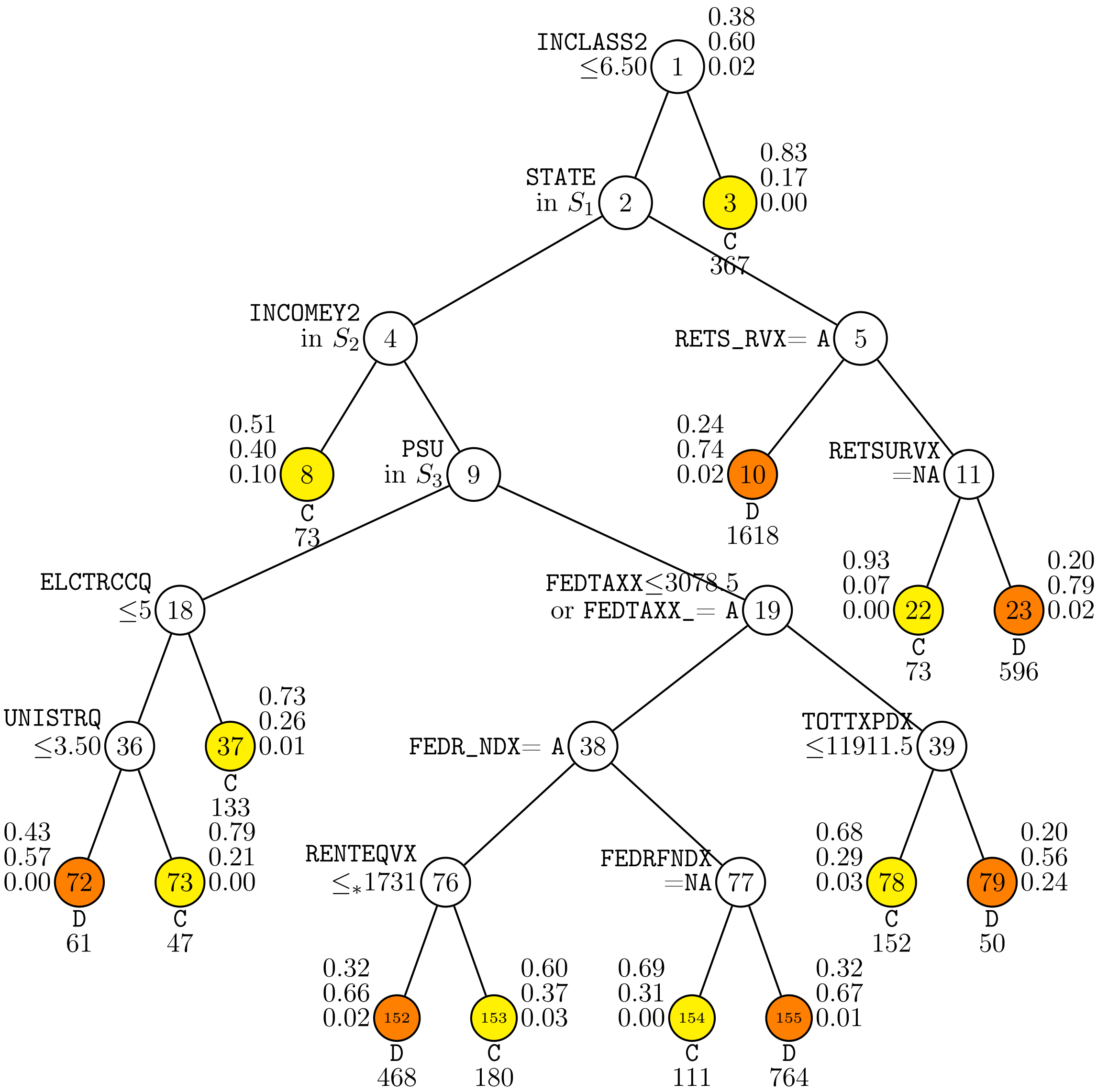

Figure 8: GUIDE v.38.0 0.25-SE classification tree for predicting INTRDVX_ using estimated priors and unit misclassification costs. Tree constructed with 4693 observations. Maximum number of split levels is 14 and minimum node sample size is 46. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '≤ or missing'. Set S_1 = {10, 12, 15, 17, 22, 25, 26, 34, 36, 39, 42, 45, 47, 53, 55, 8}. Set S_2 = {5, 6}. Set S_3 = {1102, 1423}. Predicted classes and sample sizes printed below terminal nodes; class sample proportions for INTRDVX_ = C, D, and T, respectively, beside nodes. Second best split variable at root node is INC_RANK.

```
nhtsadata.csv
NA
2
1 TSTNO x
2 BARRIG c
3 BARSHP c
4 BARANG p 360
:
28 HIC x
:
36 IMPANG p 360
:
77 CRBANG p 360
78 PDOF p 360
:
112 CARANG p 360
113 VEHOR p 360
:
147 HIC2 d
148 HIC3 x
```

In a tree with estimated priors and unit misclassification costs, the predicted class of each terminal node is the one with the largest proportion of observations. If there are two classes, this means that the predicted class is the one whose proportion of observations is greater than 0.50. Since the proportion of observations with HIC2=1 in the data is small (0.085) is very likely that each terminal node is predicted as HIC2=0 and a trivial tree results. Besides, because the data are from a designed experiment, the sample proportions of HIC2=0 and 1 are not representative of those in real accidents. If we knew the class prior probabilities in real accidents, we can use them to build a model for predicting HIC2. But since we do not know the class priors, we instead use *equal priors*, which effectively classifies each terminal node by comparing its sample proportions with 0.085 instead of 0.50. Specifically, a terminal node is predicted to class HIC2=1 if its node proportion is greater than 0.085. The result is **not** a class prediction model, but a model for estimating $P(\texttt{HIC2} = 1)$, similar to logistic regression. Following are the steps to construct an input file using equal priors.

6.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
```

Table 13: Some variable definitions for NHTSA data

Variable	Meaning
BARSHP	barrier shape (21 values)
BX2	distance from rear surface of vehicle to front of engine (mm)
BX5	distance from rear surface of vehicle to upper leading edge of left door (mm)
BX8	distance from rear surface of vehicle to upper trailing edge of right door (mm)
BX12	distance from rear surface of vehicle to bottom of a post of right side (mm)
COLMEC	steering column collapse mechanism (9 values)
ENGDSP	engine displacement (liters)
IMPANG	impact angle (clockwise with 0 degrees being straight ahead)
OCCAGE	dummy occupant age
PDOF	principal direction of force (degrees)
TRANSM	transmission type (9 values)
VEHTWT	vehicle test weight (kg)
VEHSPD	vehicle speed (km/h)
VEHWID	vehicle width (mm)
WHLBAS	wheel base (mm)
YEAR	vehicle model year (1972–2017)

```
Name of batch input file: equalp.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: equalp.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
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: nhtsaclass.dsc
Reading data description file ...
Training sample file: nhtsadata.csv
Missing value code: NA
Records in data file start on line 2
48 N variables changed to S
D variable is HIC2
Reading data file ...
Number of records in data file: 3310
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
```

```
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 2
Assigning integer codes to values of 52 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Finished assigning codes to 50 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: C variable RST5PT takes only 1 value
Warning: C variable RSTABT takes only 1 value
Warning: C variable RSTBSS takes only 1 value
Warning: C variable RSTCSR takes only 1 value
Warning: C variable RSTFSS takes only 1 value
Warning: C variable RSTISS takes only 1 value
Warning: C variable RSTOT takes only 1 value
Warning: C variable RSTSBK takes only 1 value
Warning: C variable RSTSHE takes only 1 value
Warning: C variable RSTVES takes only 1 value
Class  #Cases     Proportion
0        2999     0.91544567
1         277     0.08455433
     Total #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310         34       2891       40        0        0       49
    #P-var   #M-var   #B-var   #C-var   #I-var
         6        0        0       52        0
Number of cases used for training: 3276
Number of split variables: 101
Number of cases excluded due to 0 weight or missing D: 34
Finished reading data file
Choose 1 for estimated priors, 2 for equal priors, 3 to input priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1): 2
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
```

```
Input file name to store LaTeX code (use .tex as suffix): equalp.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: equalp.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: equalp.r
Input rank of top variable to split root node ([1:107], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < equalp.in
```

6.2 Contents of equalp.out

```
Classification tree
Pruning by cross-validation
Data description file: nhtsaclass.dsc
Training sample file: nhtsadata.csv
Missing value code: NA
Records in data file start on line 2
48 N variables changed to S
D variable is HIC2
Number of records in data file: 3310
Length of longest entry in data file: 19
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 2
Warning: C variable RST5PT takes only 1 value
Warning: C variable RSTABT takes only 1 value
Warning: C variable RSTBSS takes only 1 value
Warning: C variable RSTCSR takes only 1 value
Warning: C variable RSTFSS takes only 1 value
Warning: C variable RSTISS takes only 1 value
Warning: C variable RSTOT takes only 1 value
Warning: C variable RSTSBK takes only 1 value
Warning: C variable RSTSHE takes only 1 value
Warning: C variable RSTVES takes only 1 value
Training sample class proportions of D variable HIC2:
Class  #Cases     Proportion
0        2999     0.91544567
1         277     0.08455433

Summary information for training sample of size 3276 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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
      2  BARRIG     c                                  3
      3  BARSHP     c                                 21
      4  BARANG     p    0.000        330.0          360        14
      6  OCCTYP     c                                 13
      7  OCCAGE     s    0.000        99.00                   1242
      :
     36  IMPANG     p    0.000        330.0          360         4
      :
     77  CRBANG     p    0.000        315.0          360        24
     78  PDOF       p    0.000        345.0          360        23
     79  BMPENG     c                                  4      2055
     80  SILENG     c                                  3      2688
     81  APLENG     c                                  3      2881
    112  CARANG     p    0.000        99.00          360       991
    113  VEHOR      p    0.000        90.00          360       995
      :
    146  RSTVES     c                                  1
    147  HIC2       d                                  2

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310         34       2891       40        0        0       49
    #P-var   #M-var   #B-var   #C-var   #I-var
         6        0        0       52        0
Number of cases used for training: 3276
Number of split variables: 101
Number of cases excluded due to 0 weight or missing D: 34

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Simple node models
Equal priors
Unit misclassification costs
Univariate split highest priority
Interaction and linear splits 2nd and 3rd priorities
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 13
Minimum node sample size: 32
```

```
Top-ranked variables and chi-squared values at root node
     1  0.4697E+03   COLMEC
     2  0.3907E+03   OCCTYP
     3  0.3441E+03   YEAR
     :
    86  0.1605E+00   IMPANG
    87  0.1188E+00   RSTPS2

Size and CV mean cost and SE of subtrees:
 Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   1       35   1.850E-01   1.250E-02   8.955E-03   1.890E-01   8.765E-03
   2       34   1.850E-01   1.250E-02   8.955E-03   1.890E-01   8.765E-03
   :
  14       15   1.776E-01   1.111E-02   8.857E-03   1.804E-01   9.050E-03
  15*      14   1.763E-01   1.099E-02   8.719E-03   1.748E-01   7.172E-03
  16**      8   1.784E-01   1.113E-02   7.079E-03   1.729E-01   7.771E-03
  17        7   1.848E-01   1.179E-02   9.233E-03   1.760E-01   1.373E-02
  18        4   1.885E-01   1.180E-02   7.543E-03   1.818E-01   8.682E-03
  19        3   1.952E-01   1.166E-02   9.566E-03   1.884E-01   1.104E-02
  20        2   2.135E-01   1.560E-02   1.011E-02   2.107E-01   1.273E-02
  21        1   5.000E-01   2.875E-02   7.460E-17   5.000E-01   7.552E-17

0-SE tree based on mean is marked with * and has 14 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
+ 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    Total    Train    Predicted        Node    Split          Interacting
      label    cases    cases    class            cost    variables      variable
          1     3276     3276    0             4.949E-01 COLMEC
          2     2596     2596    0             2.310E-01 OCCTYP
          4      234      234    1             3.645E-01 BARSHP
         8T      112      112    1             2.147E-01 HW
         9T      122      122    0             2.657E-01 MODELD
          5     2362     2362    0             1.522E-01 OCCAGE
          10T     430      430     0              3.421E-01 MODELD
          11     1932     1932     0              9.609E-02 PDOF
          22T    1570     1570     0              4.577E-02 BMPENG
          23      362      362     0              2.679E-01 IMPANG
          46       89       89     1              4.175E-01 CS
          92T      39       39     1              2.330E-01 -
          93T      50       50     0              1.791E-01 -
          47T     273      273     0              7.323E-02 MODELD :YEAR
           3T     680      680     1              1.735E-01 BARSHP

 Number of terminal nodes of final tree: 8
 Total number of nodes of final tree: 15
 Second best split variable (based on curvature test) at root node is OCCTYP

 Classification tree:
 For categorical variable splits, values not in training data go to the right

 Node 1: COLMEC = "BWU", "NA", "NAP", "UNK"
   Node 2: OCCTYP = "E2", "OT", "P5", "S3", "WS"
     Node 4: BARSHP = "LCB", "POL"
       Node 8: 1
     Node 4: BARSHP /= "LCB", "POL"
       Node 9: 0
   Node 2: OCCTYP /= "E2", "OT", "P5", "S3", "WS"
     Node 5: OCCAGE = NA
       Node 10: 0
     Node 5: OCCAGE /= NA
       Node 11: PDOF in (-31, 31)
         Node 22: 0
       Node 11: PDOF not in (-31, 31) or NA
         Node 23: IMPANG in (-77, 1)
           Node 46: CS <= 274.50000
             Node 92: 1
           Node 46: CS > 274.50000 or NA
             Node 93: 0
         Node 23: IMPANG not in (-77, 1) or NA
           Node 47: 0
 Node 1: COLMEC /= "BWU", "NA", "NAP", "UNK"
   Node 3: 1

 ***************************************************************

 Predictor means below are means of cases with no missing values.

Node 1: Intermediate node
 A case goes into Node 2 if COLMEC = "BWU", "NA", "NAP", "UNK"
 COLMEC mode = "UNK"
 Class      Number   Posterior
 0            2999  0.5000E+00
 1             277  0.5000E+00
 Number of training cases misclassified = 277
 Predicted class is 0
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if OCCTYP = "E2", "OT", "P5", "S3", "WS"
 OCCTYP mode = "H3"
 Class      Number   Posterior
 0            2525  0.7666E+00
 1              71  0.2334E+00
 Number of training cases misclassified = 71
 Predicted class is 0
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if BARSHP = "LCB", "POL"
 BARSHP mode = "FLB"
 Class      Number   Posterior
 0             202  0.3683E+00
 1              32  0.6317E+00
 Number of training cases misclassified = 202
 Predicted class is 1
 ----------------------------
Node 8: Terminal node
 Class      Number   Posterior
 0              84  0.2170E+00
 1              28  0.7830E+00
 Number of training cases misclassified = 84
 Predicted class is 1
 ----------------------------
Node 9: Terminal node
 Class      Number   Posterior
 0             118  0.7315E+00
 1               4  0.2685E+00
 Number of training cases misclassified = 4
 Predicted class is 0
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if OCCAGE = NA
 OCCAGE mean = 27.055901
 Class      Number   Posterior
 0            2323  0.8462E+00
 1              39  0.1538E+00
 Number of training cases misclassified = 39
 Predicted class is 0
 ----------------------------
Node 10: Terminal node
 Class      Number   Posterior
 0             410  0.6544E+00
 1              20  0.3456E+00
 Number of training cases misclassified = 20
 Predicted class is 0
 ----------------------------
Node 11: Intermediate node
 A case goes into Node 22 if PDOF in [-31, 31]
 PDOF mean = 52.934783
 Class      Number   Posterior
 0            1913  0.9029E+00
 1              19  0.9709E-01
 Number of training cases misclassified = 19
 Predicted class is 0
 ----------------------------
Node 22: Terminal node
 Class      Number   Posterior
 0            1563  0.9538E+00
 1               7  0.4625E-01
 Number of training cases misclassified = 7
 Predicted class is 0
 ----------------------------
Node 23: Intermediate node
 A case goes into Node 46 if IMPANG in [-100, 22]
 IMPANG mean = 220.44199
 Class      Number   Posterior
 0             350  0.7293E+00
 1              12  0.2707E+00
 Number of training cases misclassified = 12
 Predicted class is 0
 ----------------------------
Node 46: Intermediate node
 A case goes into Node 92 if CS <= 274.50000
 CS mean = 262.79775
 Class      Number   Posterior
 0              79  0.4219E+00
 1              10  0.5781E+00
 Number of training cases misclassified = 79
 Predicted class is 1
 ----------------------------
Node 92: Terminal node
 Class      Number   Posterior
 0              30  0.2354E+00
 1              9  0.7646E+00
 Number of training cases misclassified = 30
 Predicted class is 1
 ----------------------------
Node 93: Terminal node
 Class      Number   Posterior
 0             49  0.8190E+00
 1              1  0.1810E+00
 Number of training cases misclassified = 1
 Predicted class is 0
 ----------------------------
Node 47: Terminal node
 Class      Number   Posterior
 0            271  0.9260E+00
 1              2  0.7399E-01
 Number of training cases misclassified = 2
 Predicted class is 0
 ----------------------------
Node 3: Terminal node
 Class      Number   Posterior
 0            474  0.1753E+00
 1            206  0.8247E+00
 Number of training cases misclassified = 474
 Predicted class is 1
 ----------------------------

 Classification matrix for training sample:
 Predicted      True class
 class              0         1
 0               2411        34
 1                588       243
 Total           2999       277

 Number of cases used for tree construction: 3276
 Number misclassified: 622
 Resubstitution estimate of mean misclassification cost: 0.15940452

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

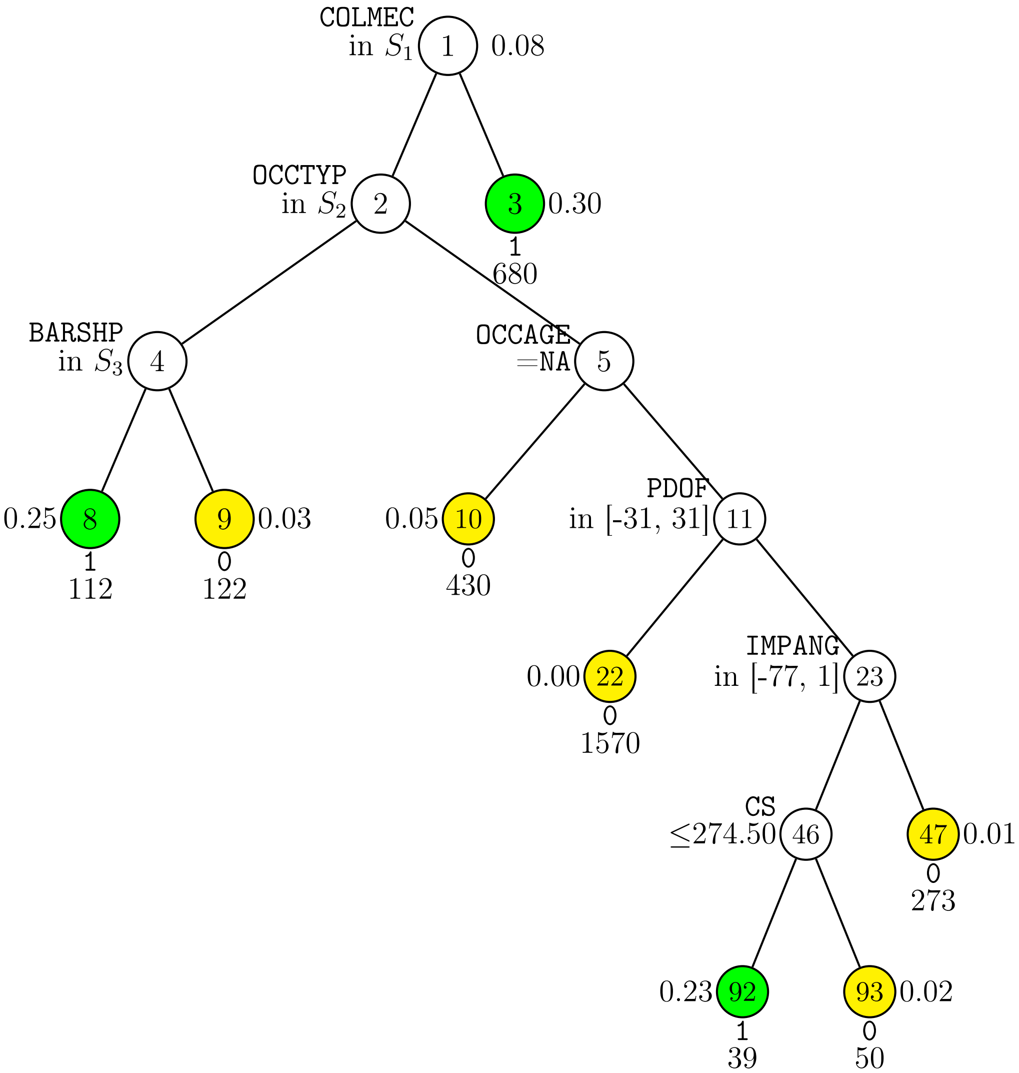

Figure 9: GUIDE v.38.0 0.25-SE classification tree for predicting HIC2 using equal priors and unit misclassification costs. Tree constructed with 3276 observations (excluding observations with non-positive weight or missing values in D, T, R or Z variables). Maximum number of split levels is 13 and minimum node sample size is 32. At each split, an observation goes to the left branch if and only if the condition is satisfied. Set S_1 = {BWU, NA, NAP, UNK}. Set S_2 = {E2, OT, P5, S3, WS}. Set S_3 = {LCB, POL}. Predicted classes and sample sizes printed below terminal nodes; class sample proportion for HIC2 = 1 beside nodes. Second best split variable at root node is OCCTYP.

7 Least squares regression: CE data

GUIDE can fit least-squares (LS), quantile, Poisson, proportional hazards, and least-median-of-squares (LMS) regression tree models. We illustrate least squares and quantile models with the CE data, using INTRDVX as the dependent variable. The description file is cereg.dsc, which sets FINLWT21 as a weight (w) variable.

7.1 Piecewise constant

7.1.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: 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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <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 (recommended if there are no missing values)
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple polynomial in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
0: stepwise linear, 1: multiple linear, 2: simple polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <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: cereg.dsc
Reading data description file ...
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
409 N variables changed to S
D variable is INTRDVX
Reading data file ...
```

```
Number of records in data file: 4693
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 44 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: S variable OTHRINCB is constant
Warning: S variable NETRENTB is constant
Warning: S variable NETRNTBX is constant
Warning: S variable OTHLONBX is constant
Warning: S variable OTHLONB is constant
Smallest positive weight:  1.3507E+03
Largest positive weight:   7.0269E+04
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693       1771       4693       30        0        0      409
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      168        0       44        0
Weight variable FINLWT21 in column: 50
Number of cases used for training: 2922
Number of split variables: 453
Number of cases excluded due to 0 weight or missing D: 1771
Finished reading data file
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction tests; too many predictor variables
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): cons.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: cons.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input rank of top variable to split root node ([1:453], <cr>=1):
```

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

7.1.2 Contents of cons.out

```
Least squares regression tree
Pruning by cross-validation
Data description file: cereg.dsc
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
409 N variables changed to S
D variable is INTRDVX
Piecewise constant model
Number of records in data file: 4693
Length of longest entry in data file: 11
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Warning: S variable OTHRINCB is constant
Warning: S variable NETRENTB is constant
Warning: S variable NETRNTBX is constant
Warning: S variable OTHLONBX is constant
Warning: S variable OTHLONB is constant
Smallest and largest positive weights are  1.3507E+03 and 7.0269E+04

Summary information for training sample of size 2922 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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
Levels of M variables are for missing values in associated variables
                                                      #Codes/
                                                      Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  DIRACC     c                                  2      116
      2  DIRACC_    m                                  1
      3  AGE_REF    s    18.00        87.00
      4  AGE_REF_   m                                  0
      :
     50  FINLWT21   w    1351.       0.7027E+05
      :
```

```
    513  INTRDVX    d    1.000       0.9834E+05
      :
    651  FSTAXOWE   s   -2505.       0.5991E+05
    652  FSTA_OWE   m                                     0
    653  ETOTA      s    1199.       0.2782E+06

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693       1771       4693       30        0        0      409
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      168        0       44        0
 Weight variable FINLWT21 in column: 50
 Number of cases used for training: 2922
 Number of split variables: 453
 Number of cases excluded due to 0 weight or missing D: 1771

 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: 0.2500

 Weighted error estimates used for pruning
 Warning: No interaction tests; too many predictor variables
 No nodewise interaction tests
 Split values for N and S variables based on exhaustive search
 Maximum number of split levels: 12
 Minimum node sample size: 29
 Top-ranked variables and chi-squared values at root node
      1  0.1648E+03   STOCKX
      2  0.1569E+03   STOCKYRX
      3  0.1212E+03   CUTENURE
      4  0.1084E+03   AGE_REF
      :
    410  0.1101E-02   ESHELTRC
    411  0.1091E-02   TVRDIOCQ

 Size and CV MSE and SE of subtrees:
  Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
    1       75   4.443E+12   4.075E+11   2.542E+11   4.543E+12   4.739E+11
    2       74   4.443E+12   4.075E+11   2.542E+11   4.543E+12   4.739E+11
    3       73   4.443E+12   4.075E+11   2.542E+11   4.543E+12   4.739E+11
    4       72   4.443E+12   4.075E+11   2.542E+11   4.543E+12   4.739E+11
    5*      71   4.443E+12   4.075E+11   2.542E+11   4.543E+12   4.739E+11
    6+      70   4.443E+12   4.075E+11   2.542E+11   4.543E+12   4.739E+11
    :
   41**     13   4.480E+12   4.202E+11   2.680E+11   4.610E+12   5.082E+11
   42       10   4.601E+12   4.474E+11   2.782E+11   4.805E+12   4.819E+11
  43        5   4.991E+12   4.896E+11   2.562E+11   4.855E+12   3.662E+11
  44        2   5.588E+12   5.911E+11   2.712E+11   5.581E+12   2.200E+11
  45        1   5.572E+12   5.900E+11   2.831E+11   5.540E+12   2.166E+11

 0-SE tree based on mean is marked with * and has 71 terminal nodes
 0-SE tree based on median is marked with + and has 70 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-mean is weighted mean of INTRDVX 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    Total    Cases Matrix    Node      Node         Split
        label   cases      fit  rank    D-mean     MSE          variable
           1     2922     2922     1  4.697E+03  5.572E+12   STOCKX
           2     2891     2891     1  4.288E+03  4.948E+12   RENTEQVX
           4     2750     2750     1  3.513E+03  3.680E+12   AGE_REF
           8T    1153     1153     1  1.398E+03  1.693E+12   STATE
           9     1597     1597     1  5.110E+03  5.001E+12   RENTEQVX
          18T     845      845     1  3.046E+03  2.812E+12   STATE
          19      752      752     1  7.871E+03  7.234E+12   EMRTPNOP
          38      421      421     1  1.071E+04  9.838E+12   FFTAXOWE
          76      283      283     1  6.941E+03  4.538E+12   FFTAXOWE
         152T     161      161     1  3.836E+03  1.367E+12   INC_RANK
         153      122      122     1  1.079E+04  8.284E+12   INCNONW1
         306T      50       50     1  1.907E+03  3.700E+11   -
         307       72       72     1  1.951E+04  1.147E+13   STATE
         614T      30       30     1  3.070E+04  1.842E+13   -
         615T      42       42     1  9.062E+03  3.693E+12   -
          77      138      138     1  1.850E+04  1.926E+13   FJSSDEDX
         154T      46       46     1  3.739E+04  3.251E+13   -
         155T      92       92     1  9.204E+03  8.334E+12   SEX_REF
          39T     331      331     1  4.371E+03  3.544E+12   PRINEARN
           5      141      141     1  2.158E+04  2.449E+13   STATE
          10T      31       31     1  2.796E+03  3.773E+11   -
          11      110      110     1  2.564E+04  2.965E+13   STATE
          22T      80       80     1  5.393E+03  3.933E+12   FEDTAXX
          23T      30       30     1  6.450E+04  4.664E+13   -
          3T      31       31     1  4.774E+04  3.242E+13   -

 Number of terminal nodes of final tree: 13
 Total number of nodes of final tree: 25
 Second best split variable (based on curvature test) at root node is STOCKYRX

 Regression tree:
 For categorical variable splits, values not in training data go to the right

 Node 1: STOCKX <= 583000.00 or NA
   Node 2: RENTEQVX <= 3947.0000 or NA
     Node 4: AGE_REF <= 53.500000
       Node 8: INTRDVX-mean = 1397.6608
     Node 4: AGE_REF > 53.500000 or NA
       Node 9: RENTEQVX <= 1261.5000 or NA
         Node 18: INTRDVX-mean = 3046.3296
       Node 9: RENTEQVX > 1261.5000
         Node 19: EMRTPNOP <= 3.1665000
           Node 38: FFTAXOWE <= 10182.500
             Node 76: FFTAXOWE <= 1443.0000
               Node 152: INTRDVX-mean = 3836.1467
             Node 76: FFTAXOWE > 1443.0000 or NA
               Node 153: INCNONW1 = "2"
                       or (INCNONW1 = NA & INCN_NW1 = "A")
                 Node 306: INTRDVX-mean = 1906.8797
               Node 153: INCNONW1 /= "2"
                       & not (INCNONW1 = NA & INCN_NW1 = "A")
                 Node 307: STATE = "12", "17", "23", "25", "36", "45", "48", "51",
                        "8", "9", "NA"
                   Node 614: INTRDVX-mean = 30695.698
                 Node 307: STATE /= "12", "17", "23", "25", "36", "45", "48", "51",
                        "8", "9", "NA"
                   Node 615: INTRDVX-mean = 9062.2070
           Node 38: FFTAXOWE > 10182.500 or NA
             Node 77: FJSSDEDX <= 3557.5000
               Node 154: INTRDVX-mean = 37391.540
             Node 77: FJSSDEDX > 3557.5000 or NA
               Node 155: INTRDVX-mean = 9204.1056
         Node 19: EMRTPNOP > 3.1665000 or NA
           Node 39: INTRDVX-mean = 4371.0642
   Node 2: RENTEQVX > 3947.0000
     Node 5: STATE = "1", "12", "15", "2", "31", "48", "49", "51", "53"
       Node 10: INTRDVX-mean = 2796.3030
     Node 5: STATE /= "1", "12", "15", "2", "31", "48", "49", "51", "53"
       Node 11: STATE = "17", "24", "25", "36", "6"
         Node 22: INTRDVX-mean = 5393.3215
```

```
      Node 11: STATE /= "17", "24", "25", "36", "6"
        Node 23: INTRDVX-mean = 64504.443
 Node 1: STOCKX > 583000.00
   Node 3: INTRDVX-mean = 47739.942

 ***************************************************************

 Predictor means below are weighted means of cases with no missing values.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if STOCKX <= 583000.00 or NA
 STOCKX mean = 453208.43
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant      4697.        14.01        0.000
 INTRDVX mean = 4696.62
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if RENTEQVX <= 3947.0000 or NA
 RENTEQVX mean = 1549.7905
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if AGE_REF <= 53.500000
 AGE_REF mean = 55.210006
 ----------------------------
Node 8: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      1398.        4.841       0.1466E-05
 INTRDVX mean = 1397.66
 ----------------------------
 :
 :
Node 3: Terminal node
 Coefficients of least squares regression functions:
```

```
Regressor    Coefficient  t-stat      p-value
Constant     0.4774E+05    6.053       0.1203E-05
INTRDVX mean = 47739.9
----------------------------
Proportion of variance (R-squared) explained by tree model: 0.3068

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

In the above results, the pruned tree is marked with two asterisks (tree #41). It has 13 terminal nodes and a cross-validation estimate of prediction mean squared error of 4.480E+12. Figure 10 shows the tree. The first split is on amount of stocks, with `STOCKX` $\leq$ \$583000 or missing going to node 2 (in the tree diagram, the symbol "$\leq_*$" stands for "$\leq$ or missing"). Node 3 consists of 31 observations with a mean `INTRDVX` of \$47740. The file `cons.fit` gives the predicted value of `INTRDVX` of each observation, including those for which the observed value of `INTRDVX` is missing. For example, the first 7 entries of `cons.fit` below show that the 7th observation, for which `INTRDVX` is missing (the letter "`n`" in the first column indicates that it is not used to train the model), belongs to node 18 and has a predicted value of \$3046.

```
train         node      observed         predicted
    y           18     1.300000E+01     3.046330E+03
    y           18     2.000000E+00     3.046330E+03
    y            8     2.270000E+02     1.397661E+03
    y            8     2.000000E+02     1.397661E+03
    y            8     9.000000E+01     1.397661E+03
    y            3     3.150000E+04     4.773994E+04
    n           18               NA     3.046330E+03
```

7.2 Piecewise simple linear

GUIDE can also fit a simple linear regression model in each node, where "simple" means that only one predictor variable is used in each node. The selected variable is the one that yields the smallest sum of squared residuals.

7.2.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
```

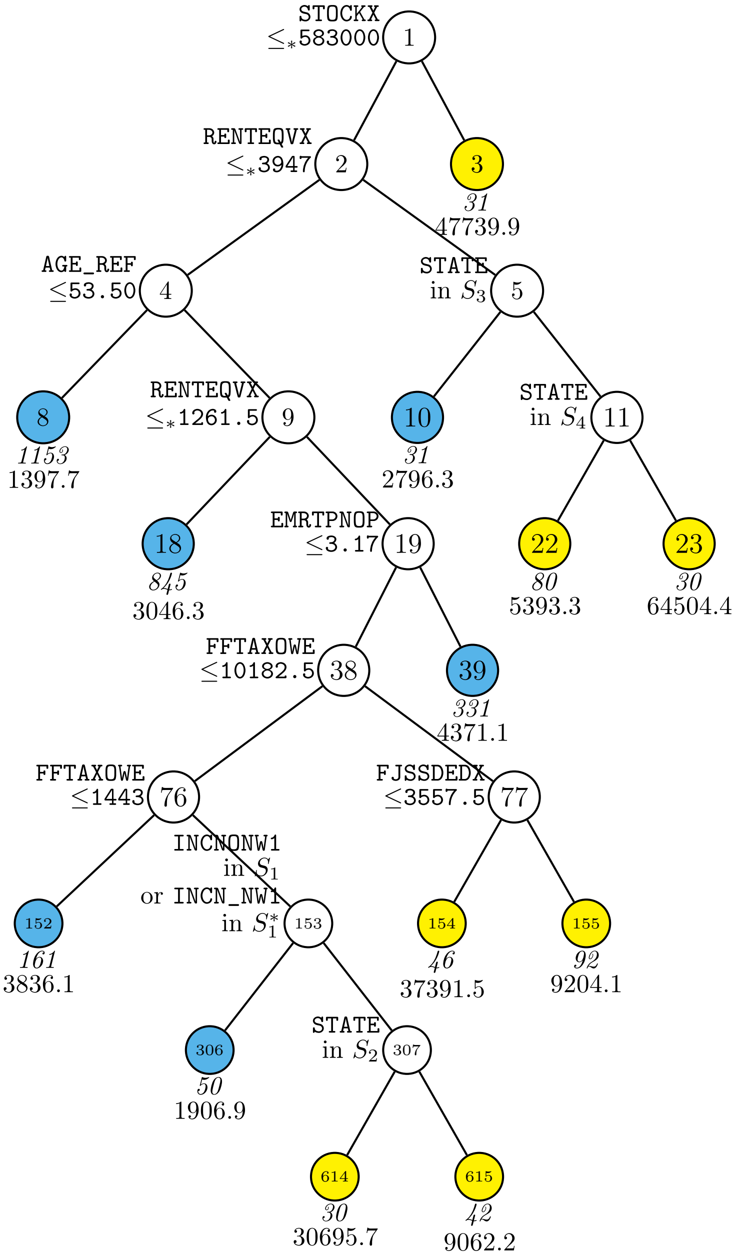

Figure 10: GUIDE v.38.0 0.25-SE piecewise constant weighted least-squares regression tree for predicting INTRDVX. Sample size (*in italics*) and mean of INTRDVX printed below nodes. Terminal nodes with means above and below value of 4696.6 at root node are colored yellow and skyblue respectively. Second best split variable at root node is STOCKYRX.

```
Name of batch input file: simple.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: simple.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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <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 (recommended if there are no missing values)
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple polynomial in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
0: stepwise linear, 1: multiple linear, 2: simple polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <cr>=3): 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):
Input 1 for interaction tests, 2 to skip them ([1:2], <cr>=1):
Input 0 to get tree with fixed no. of nodes, 1 to prune by CV, 2 for no pruning ([0:2], <cr>=1):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: cereg.dsc
Reading data description file ...
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
D variable is INTRDVX
Reading data file ...
Number of records in data file: 4693
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
```

```
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 44 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: N variable OTHRINCB is constant
Warning: N variable NETRENTB is constant
Warning: N variable NETRNTBX is constant
Warning: N variable OTHLONBX is constant
Warning: N variable OTHLONB is constant
Smallest positive weight:  1.3507E+03
Largest positive weight:   7.0269E+04
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693       1771       4693       30      409        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      168        0       44        0
Weight variable FINLWT21 in column: 50
Number of cases used for training: 2922
Number of split variables: 453
Number of cases excluded due to 0 weight or missing D: 1771
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>=0.25):
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction tests; too many predictor variables
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: 12
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 30
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): simple.tex
Choose color(s) for the terminal nodes:
(0) white
(1) yellow-skyblue
(2) yellow-purple
(3) yellow-orange
(4) orange-skyblue
(5) yellow-red
(6) orange-purple
(7) grayscale
Input your choice ([0:7], <cr>=2):
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): 2
Input file name: simple.var
Input 2 to save regressor names in a file, 1 otherwise ([1:2], <cr>=1): 2
Input file name: simple.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: simple.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: simple.r
Input rank of top variable to split root node ([1:453], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < simple.in
```

7.2.2 Results

The tree is shown in Figure 11. Below each terminal node are printed the sample size (in italics), the sample mean of INTRDVX and the signed simple linear predictor, with the sign being that of the slope coefficient. Nodes with negative and positive slopes are colored yellow and green, respectively. The regression coefficient estimates are given in the output file below.

```
Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       61   4.342E+12  4.344E+11   5.889E+11   3.935E+12   9.722E+11
```

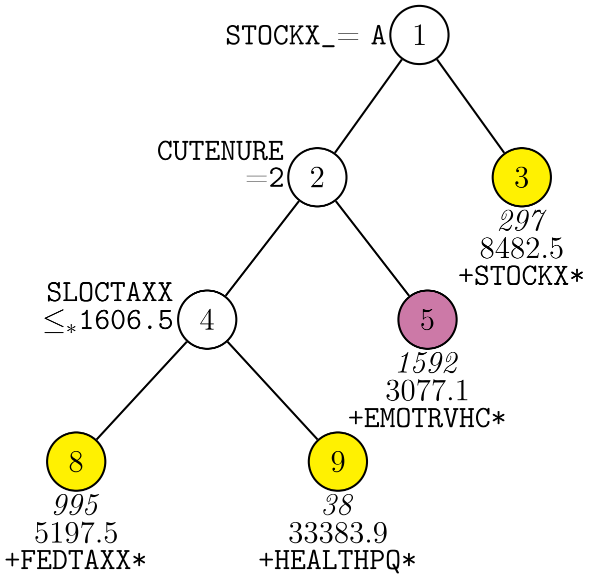

Figure 11: GUIDE v.38.0 0.25-SE piecewise simple linear weighted least-squares regression tree for predicting `INTRDVX`. Tree constructed with 2922 observations (excluding observations with non-positive weight or missing values in `D`, `T`, `R` or `Z` variables). Maximum number of split levels is 12 and minimum node sample size is 30. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '≤ or missing'. Sample size (*in italics*), mean of `INTRDVX`, and sign and name of regressor variable printed below nodes. Terminal nodes with means above and below value of 4696.6 at root node are colored yellow and purple respectively. Terminal nodes with means above and below value of 4696.6 at root node are colored yellow and purple respectively. Asterisk appended to regressor name indicates its slope is significant at the 0.05 level (unadjusted for multiplicity and model fitting). Second best split variable at root node is `STOCKYRX`.

```
   2       60   4.342E+12   4.344E+11   5.889E+11   3.935E+12   9.722E+11
   :
  36*      14   4.001E+12   4.235E+11   4.823E+11   3.667E+12   9.546E+11
  37       13   4.015E+12   4.236E+11   4.859E+11   3.667E+12   9.689E+11
  38+      12   4.230E+12   4.546E+11   5.114E+11   3.667E+12   1.073E+12
  39        9   4.238E+12   4.547E+11   4.855E+11   3.771E+12   1.031E+12
  40        8   4.216E+12   4.495E+11   4.958E+11   3.771E+12   9.887E+11
  41**      4   4.082E+12   4.318E+11   4.155E+11   3.849E+12   9.017E+11
  42        3   4.351E+12   4.618E+11   3.694E+11   4.462E+12   6.027E+11
  43        1   5.061E+12   5.671E+11   3.281E+11   5.014E+12   5.487E+11

0-SE tree based on mean is marked with * and has 14 terminal nodes
0-SE tree based on median is marked with + and has 12 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-mean is weighted mean of INTRDVX in the node
Cases fit give the number of cases used to fit node
MSE and R^2 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
          1     2922     2922     2  4.697E+03  5.052E+12  0.0938 STOCKX  +STOCKX
          2     2625     2625     2  4.306E+03  4.881E+12  0.0551 CUTENURE  +RENTEQVX
          4     1033     1033     2  6.235E+03  5.876E+12  0.1272 SLOCTAXX  +SLOCTAXX
          8T     995      995     2  5.197E+03  4.712E+12  0.0568 PSU  +FEDTAXX
          9T      38       38     2  3.338E+04  1.995E+13  0.4917 -  +HEALTHPQ
          5T    1592     1592     2  3.077E+03  3.146E+12  0.2286 STATE  +EMOTRVHC
          3T     297      297     2  8.482E+03  3.787E+12  0.5775 STOCKX  +STOCKX

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 STOCKYRX

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: STOCKX = NA & STOCKX_ = "A"
  Node 2: CUTENURE = "2"
```

```
     Node 4: SLOCTAXX <= 1606.5000 or NA
       Node 8: INTRDVX-mean = 5197.4665
     Node 4: SLOCTAXX > 1606.5000
       Node 9: INTRDVX-mean = 33383.851
   Node 2: CUTENURE /= "2"
     Node 5: INTRDVX-mean = 3077.0644
 Node 1: not (STOCKX = NA & STOCKX_ = "A")
   Node 3: INTRDVX-mean = 8482.4790

 ***************************************************************

 Predictor means below are weighted means of cases with no missing values.
 Regression coefficients are computed from the complete cases.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if STOCKX = NA & STOCKX_ = "A"
 STOCKX mean = 453208.43
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -1422.       -3.086       0.2050E-02
 STOCKX       0.1350E-01    17.38       0.1110E-15    25.00       0.4532E+06   0.6587E+07
 INTRDVX mean = 4696.62
 Predicted values truncated at 1.00000 & 98338.0
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if CUTENURE = "2"
 CUTENURE mode = "1"
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if SLOCTAXX <= 1606.5000 or NA
 SLOCTAXX mean = 2431.3388
 ----------------------------
Node 8: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      1366.        1.955       0.5084E-01
 FEDTAXX      0.8279        7.733        0.000         2.000         4627.        0.8223E+05
 INTRDVX mean = 5197.47
 Predicted values truncated at 1.00000 & 98338.0
 ----------------------------
Node 9: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.1213E+05    1.921       0.6270E-01
 HEALTHPQ      11.96        5.901       0.9419E-06    0.000         1778.       0.1303E+05
 INTRDVX mean = 33383.9
 Predicted values truncated at 1.00000 & 98338.0
 ----------------------------
Node 5: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      2585.        8.170       0.1665E-14
 EMOTRVHC      143.6        21.71        0.000         0.000         3.431        667.0
 INTRDVX mean = 3077.06
 Predicted values truncated at 1.00000 & 98338.0
 ----------------------------
Node 3: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      2364.        2.638       0.8770E-02
 STOCKX       0.1350E-01    20.08        0.000         25.00       0.4532E+06   0.6587E+07
 INTRDVX mean = 8482.48
 Predicted values truncated at 1.00000 & 98338.0
 ----------------------------
 Proportion of variance (R-squared) explained by tree model: 0.2969

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

The pruned tree (marked with two asterisks) has 4 terminal nodes and a cross-validation estimate of prediction mean squared error of 4.085E+12.

7.2.3 Plots of data

Figure 12 shows plots of the data and fitted regression lines in the terminal nodes of the tree. The plots are drawn using the R code in Figure 13, which reads the file

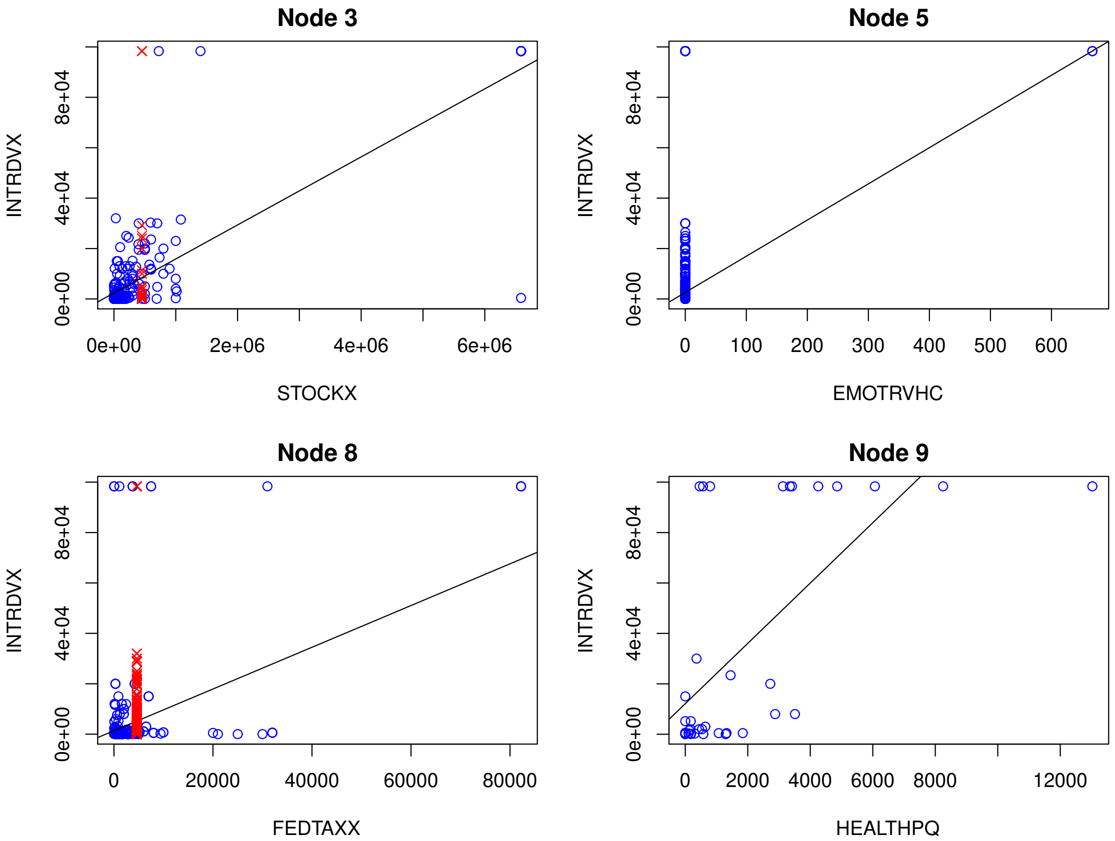

Figure 12: Plots of data and regression lines in terminal nodes of tree in Figure 11. Red colored points are imputed with node means.

```
par(mfrow=c(3,2),mar=c(5,4.5,2,1),cex=1.1)
z1 <- read.table("simple.fit",header=TRUE)
z2 <- read.table("simple.reg",header=TRUE)
nodes <- unique(sort(z1$node))
y <- z$INTRDVX
for(n in nodes){
    gp <- z1$node == n & z1$train == "y"
    vrow <- z2$node == n
    b0 <- z2$beta0[vrow]
    b1 <- z2$beta1[vrow]
    reg <- z2$variable[vrow]
    k <- which(names(z) %in% reg)
    x <- z[,k]
    plot(y[gp] ~ x[gp],xlab=reg,ylab="INTRDVX",col="blue")
    abline(c(b0,b1))
    nomiss <- z1$node == n & z1$train == "y" & !is.na(x)
    if(sum(nomiss) < sum(gp)){
        x0 <- z[nomiss,k]
        w <- z$FINLWT21[nomiss]
        xmean <-sum(x0*w)/sum(w)
        miss <- z1$node == n & z1$train == "y" & is.na(x)
        points(rep(xmean,sum(miss)),y[miss],col="red",pch=4)
    }
    title(paste("Node",n))
}
```

Figure 13: R code for Figure 12

simple.reg whose contents are below. The first row is a header line. Each subsequent row gives the terminal node number, predictor variable name, intercept and slope of the regression line, and lower and upper truncation limits on the predicted values (the latter defaults are the global minimum and maximum observed values of the dependent variable).

```
node variable  beta0          beta1        lower  upper
8    FEDTAXX   1366.          0.8279       1.000  0.9834E+005
9    HEALTHPQ  0.1213E+005    11.96        1.000  0.9834E+005
5    EMOTRVHC  2585.          143.6        1.000  0.9834E+005
3    STOCKX    2364.          0.1350E-001  1.000  0.9834E+005
```

Missing values in the linear predictor are replaced by the mean of the nonmissing values in the node in estimation of the regression line.

7.3 Stepwise linear

GUIDE can also use stepwise regression to fit a multiple linear model in each node. Quite often, such a models yields even higher prediction accuracy, as measured by the cross-validation estimates of MSE in the output, as is the case here.

7.3.1 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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <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 (recommended if there are no missing values)
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple polynomial in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
0: stepwise linear, 1: multiple linear, 2: simple polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <cr>=3): 0
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: cereg.dsc
Reading data description file ...
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
D variable is INTRDVX
Reading data file ...
```

```
Number of records in data file: 4693
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 44 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: N variable OTHRINCB is constant
Warning: N variable NETRENTB is constant
Warning: N variable NETRNTBX is constant
Warning: N variable OTHLONBX is constant
Warning: N variable OTHLONB is constant
Smallest positive weight:  1.3507E+03
Largest positive weight:   7.0269E+04
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693       1771       4693       30      409        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      168        0       44        0
Weight variable FINLWT21 in column: 50
Number of cases used for training: 2922
Number of split variables: 453
Number of cases excluded due to 0 weight or missing D: 1771
Finished reading data file
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction tests; too many predictor variables
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): step.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: step.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: step.r
Input rank of top variable to split root node ([1:453], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < step.in
```

7.3.2 Results

```
Least squares regression tree
Predictions truncated at global min. and max. of D sample values
Pruning by cross-validation
Data description file: cereg.dsc
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
D variable is INTRDVX
Piecewise forward and backward stepwise regression
F-to-enter and F-to-delete: 4.000 3.990
Using as many variables as needed
Number of records in data file: 4693
Length of longest entry in data file: 11
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Warning: N variable OTHRINCB is constant
Warning: N variable NETRENTB is constant
Warning: N variable NETRNTBX is constant
Warning: N variable OTHLONBX is constant
Warning: N variable OTHLONB is constant
Smallest and largest positive weights are  1.3507E+03 and 7.0269E+04

Summary information for training sample of size 2922 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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
Levels of M variables are for missing values in associated variables
                                                     #Codes/
                                                     Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  DIRACC     c                                  2      116
      2  DIRACC_    m                                  1
      3  AGE_REF    n    18.00        87.00
      4  AGE_REF_   m                                  0
      :
     50  FINLWT21   w    1351.       0.7027E+05
      :
    513  INTRDVX    d    1.000       0.9834E+05
      :
    651  FSTAXOWE   n   -2505.       0.5991E+05
    652  FSTA_OWE   m                                 0
    653  ETOTA      n    1199.       0.2782E+06

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693       1771       4693       30      409        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      168        0       44        0
 Weight variable FINLWT21 in column: 50
 Number of cases used for training: 2922
 Number of split variables: 453
 Number of cases excluded due to 0 weight or missing D: 1771

 Missing values imputed with node means for fitting regression models in nodes
 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: .2500

 Weighted error estimates used for pruning
 Warning: No interaction tests; too many predictor variables
 No nodewise interaction tests
 Fraction of cases used for splitting each node: 1.0000
 Maximum number of split levels: 12
 Minimum node sample size: 25
 Top-ranked variables and chi-squared values at root node
      1  0.7816E+03   RETSURV
      2  0.4748E+03   RETSURVX
      3  0.9677E+02   ROYESTX
      :
    394  0.1977E-03   WHLFYRX
    395  0.5401E-04   WINDOWAC

 Size and CV MSE and SE of subtrees:
  Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
    1       10   1.227E+12   1.336E+11   1.450E+11   1.034E+12   2.072E+11
    2        9   1.227E+12   1.336E+11   1.450E+11   1.034E+12   2.072E+11
    3        8   1.227E+12   1.336E+11   1.450E+11   1.034E+12   2.072E+11
    4        5   1.227E+12   1.336E+11   1.450E+11   1.034E+12   2.072E+11
    5        4   1.227E+12   1.336E+11   1.450E+11   1.034E+12   2.072E+11
    6**      2   8.646E+11   5.654E+10   6.029E+10   8.156E+11   7.544E+10
    7        1   1.481E+12   1.132E+11   1.138E+11   1.317E+12   1.390E+11

 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.

 D-mean is weighted mean of INTRDVX in the node
 Cases fit give the number of cases used to fit node
 MSE and R^2 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
           1     2922     2922    45  4.697E+03  1.562E+12  0.7240 RETSURV
           2T     812      812    42  6.280E+03  1.045E+12  0.8405 ROYESTX
           3T    2110     2110    27  4.139E+03  7.727E+11  0.8560 NETRENTX

 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 RETSURVX

 Regression tree:
 For categorical variable splits, values not in training data go to the right

 Node 1: RETSURV = "1"
   Node 2: INTRDVX-mean = 6279.5195
 Node 1: RETSURV /= "1"
   Node 3: INTRDVX-mean = 4138.8576

 ***************************************************************

 Predictor means below are weighted means of cases with no missing values.
 Regression coefficients are computed from the complete cases.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.
 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if RETSURV = "1"
 RETSURV mode = "2"
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.8372E+05    4.964      0.7321E-06
 AGE_REF     -52.05        -3.872       0.1103E-03    18.00        55.40        87.00
 FINCBTAX     0.6396        70.45        0.000      -0.3430E+06   0.9699E+05   0.1410E+07
 FRRETIRX    -0.7917       -32.45        0.000        0.000        7036.       0.5241E+05
 FSALARYX    -0.6365       -68.50        0.000        0.000       0.6786E+05   0.5301E+06
 FSSIX       -0.9345       -2.572       0.1016E-01    0.000        24.41       0.3048E+05
 INCWEEK1      51.80         5.492       0.4311E-07    0.000        31.18        52.00
 INCWEEK2      34.33         3.092       0.2009E-02    0.000        32.50        52.00
 LUMPSUMX    -0.5825E-01   -4.517       0.6525E-05    4.000       0.5649E+05   0.5492E+06
 NONINCMX    -0.5726       -39.19        0.000        0.000        3791.       0.5492E+06
 OTHRINCX    -0.7307       -8.454        0.000        2.000        9799.       0.5788E+05
 RENTEQVX      1.370         5.553       0.3059E-07    1.000        1561.        4694.
 SLOCTAXX     0.3004        3.863       0.1143E-03    1.000        2248.       0.2657E+05
 VEHQ        -65.45       -0.5880       0.5566        0.000        2.366        17.00
 FDHOMEPQ     0.9952        3.474       0.5209E-03    0.000        902.8        8450.
 FDHOMECQ    -1.602        -3.963       0.7583E-04    0.000        440.4        6067.
 PROPTXPQ    -1.525        -4.201       0.2737E-04    0.000        479.3        4870.
 PROPTXCQ      1.610         2.610       0.9094E-02    0.000        234.1        4247.
 ALLFULCQ    -3.163        -3.008       0.2649E-02    0.000        29.78        3081.
 TEXTILPQ    -7.564        -3.363       0.7805E-03    0.000        16.87        4000.
 TEXTILCQ      6.800         2.695       0.7075E-02    0.000        9.375        2946.
 FLRCVRPQ      1.754         2.513       0.1201E-01    0.000        25.36       0.1000E+05
 CARTKNPQ    -0.1266       -2.488       0.1291E-01    0.000        549.3       0.8700E+05
 GASMOPQ     -2.034        -4.178       0.3024E-04    0.000        480.0        4832.
 MAINRPPQ    -1.179        -2.794       0.5244E-02    0.000        173.0        4984.
 MEDSRVPQ     0.7514        3.120       0.1828E-02   -475.0        238.0       0.1198E+05
 PETTOYCQ    -2.673        -2.791       0.5292E-02    0.000        43.48        5657.
 EDUCAPQ      0.4678        4.267       0.2045E-04    0.000        299.4       0.3500E+05
 LIFINSCQ    -1.074        -1.558       0.1194        0.000        54.04        5842.
 TOTHRLOC      1.033         1.751       0.8011E-01    0.000        60.79        7498.
 VOTHRFLP    -39.05        -5.040       0.4947E-06    0.000        1.826        547.0
 VELECTRP      27.46         4.884       0.1098E-05    0.000        4.360        1360.
 MRTPRNOP    -0.7381       -2.653       0.8028E-02    0.000        28.16       0.2643E+05
 UTILRNTC      38.89         4.068       0.4872E-04    0.000       0.8167        628.0
 ETRANPTP     0.2461        3.713       0.2084E-03    0.000        1802.       0.8868E+05
 FSMPFRMX    -0.6482       -65.00        0.000      -0.4000E+06    4794.       0.1090E+07
 NETRENTX    -0.5793       -20.73        0.000      -0.5499E+05    8909.       0.1148E+06
 OTHREGBX    -0.6712       -5.477       0.4697E-07    488.0        0.1985E+05   0.5000E+05
 OTHREGX     -0.6038       -12.56        0.000        100.0        0.1052E+05   0.6367E+05
 RETSURVX    -0.6462       -44.63        0.000        30.00        0.2454E+05   0.1269E+06
 RETSURVB     -2905.       -4.041       0.5473E-04    1.000         6.976        12.00
 ROYESTBX     -1.830      -0.5118       0.6088        1300.         4415.        6000.
 ROYESTX     -0.6067       -25.97        0.000        1.000        0.1681E+05   0.1592E+06
 STOCKX       0.4833E-02    10.37        0.000        25.00        0.4532E+06   0.6587E+07
 WHLFYRX     -0.2304E-01   -3.378       0.7397E-03    0.000        0.5156E+05   0.7674E+06
 INTRDVX mean = 4696.62
 Predicted values truncated at 1.00000 & 98338.0
 ----------------------------
Node 2: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.6444E+05    9.121        0.000
 AGE2         -139.0       -3.054       0.2332E-02    22.00         66.74        87.00
 FEDTAXX      0.1963        4.652       0.3878E-05    2.000         6965.       0.8223E+05
 FINCBTAX     0.6232        35.04        0.000        50.00        0.7759E+05   0.6717E+06
 FRRETIRX    -0.7928       -23.18        0.000        0.000        0.1663E+05   0.5241E+05
 FSALARYX    -0.7188       -40.06        0.000        0.000        0.2276E+05   0.2950E+06
 FSSIX       -0.4970       -1.386       0.1662        0.000         48.75       0.3048E+05
 HLFBATHQ      1516.        2.801       0.5215E-02    0.000        0.4072        3.000
 INCWEEK1     -41.60       -2.314       0.2092E-01    0.000         11.29        52.00
 MISCTAXX     0.8294        1.904       0.5726E-01    30.00         3760.       0.1376E+05
 LUMPSUMX    -0.1803       -7.808       0.1943E-13    4.000        0.4387E+05   0.5492E+06
 NONINCMX    -0.5140       -18.85        0.000        0.000         4166.       0.5492E+06
 OTHRINCX    -0.8129       -3.598       0.3409E-03    250.0         7826.       0.2600E+05
 PERSOT64      2452.        4.326       0.1716E-04    0.000         1.104        3.000
 VEHQ         -1105.       -6.041       0.2384E-08    0.000         2.230        10.00
 PROPTXCQ      3.252        3.394       0.7229E-03    0.000         254.7        2580.
 ELCTRCCQ      4.704        2.741       0.6262E-02    0.000         139.5        2200.
 ALLFULPQ     -3.558       -2.826       0.4834E-02    0.000         56.96        2524.
 MENSIXCQ      14.06        2.345       0.1931E-01    0.000         11.96        674.0
 WOMGRLCQ     -10.22       -2.915       0.3656E-02    0.000         24.00        1174.
 FOOTWRPQ     -14.89       -3.960       0.8195E-04    0.000         28.01        1559.
 VEHFINPQ     -8.499       -2.247       0.2491E-01    0.000         29.70        561.0
 VRNTLOPQ      2.472        3.052       0.2351E-02    0.000         105.4        5439.
 FEEADMPQ      1.825        2.252       0.2458E-01    0.000         140.8        6279.
 READPQ        4.491        1.919       0.5533E-01    0.000         48.05        2794.
 MISCPQ       0.6091        1.747       0.8097E-01    0.000         163.8       0.1209E+05
 TFOODTOC     -16.43       -3.196       0.1448E-02    0.000         57.01        4305.
 TFOODAWC      27.25        4.370       0.1414E-04    0.000         47.30        4180.
 UTILRNTC      58.72        4.644       0.4016E-05    0.000        0.8257        628.0
 ETOTALP      0.1706        3.490       0.5114E-03    730.2         9628.       0.7568E+05
 INCLASS2      2169.        6.820       0.1841E-10    1.000         4.029        7.000
 ERANKHM      -5305.       -3.102       0.1990E-02  0.2467E-01     0.5909       0.9989
 CREDYRBX     -1.842       -3.110       0.1937E-02    250.0        5732.        0.2250E+05
 FSMPFRMX    -0.6933       -26.74       0.1110E-15  -0.1030E+05    2143.        0.5800E+06
 NETRENTX    -0.7539       -12.89       0.6661E-15  -0.5499E+05    6185.        0.1148E+06
 OTHLONX      1.130         4.428       0.1087E-04    1.000        9160.        0.3800E+05
 OTHREGX     -0.6880       -7.403       0.3496E-12    395.0      0.1367E+05     0.6367E+05
 RETSURVX    -0.7478       -39.02       0.4441E-15    30.00      0.2454E+05     0.1269E+06
 RETSURVB     -3999.       -6.650       0.5543E-10    1.000        6.976         12.00
 ROYESTX     -0.6943       -15.04        0.000        1.000      0.1002E+05     0.1592E+06
 STOCKX       0.2419E-02    2.643       0.8382E-02    200.0      0.4863E+06     0.6587E+07
 FFTAXOWE     0.3263        4.715       0.2873E-05   -4590.        8090.        0.1616E+06
 INTRDVX mean = 6279.52
 Predicted values truncated at 1.00000 & 98338.0
 ---------------------------
Node 3: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean        Maximum
 Constant     0.6368E+05    16.31        0.000
 FINCBTAX     0.7968        90.49        0.000      -0.3430E+06    0.1038E+06    0.1410E+07
 FJSSDEDX     0.1945        3.161       0.1597E-02    0.000        6419.        0.3042E+05
 FRRETIRX    -0.7935       -37.38        0.000        0.000        3657.        0.4935E+05
 FSALARYX    -0.8060       -83.36        0.000        0.000      0.8375E+05     0.5301E+06
 INCWEEK2     36.88         3.890       0.1032E-03    0.000        37.86         52.00
 LUMPSUMX    -0.6489E-01   -6.507       0.4453E-10    10.00      0.6385E+05     0.5492E+06
 NO_EARNR     -881.1       -4.527       0.6306E-05    0.000        1.505         6.000
 NONINCMX    -0.7247       -57.14        0.000        0.000        3658.        0.5492E+06
 OTHRINCX    -0.8788       -13.53        0.000        2.000      0.1034E+05     0.5788E+05
 WELFAREX     -3.019      -0.8521       0.3943        300.0        861.6         4344.
 TEXTILCQ     11.91         4.331       0.1558E-04    0.000        9.673         815.0
 OTHVEHPQ     0.9109        2.519       0.1184E-01    0.000        14.81        0.1166E+05
 TRNTRPPQ     0.3714        2.144       0.3218E-01    0.000        183.8        0.2067E+05
 HLTHINPQ    -0.5893       -3.356       0.8046E-03    0.000        522.2        0.1221E+05
 PETTOYCQ     -3.391       -4.518       0.6605E-05    0.000        42.75         5657.
 CASHCOCQ    -0.5230       -2.494       0.1271E-01    0.000        213.3        0.1250E+05
 TOTHRLOC     1.506         3.315       0.9305E-03    0.000        59.95         7498.
 VELECTRP     16.16         5.028       0.5382E-06    0.000        4.196         1360.
 EMOTRVHC     33.33         9.463        0.000        0.000        2.569         667.0
 FSMPFRMX    -0.8135       -84.57        0.000      -0.4000E+06    5728.        0.1090E+07
 MLPYQWKS     130.7         3.277       0.1067E-02    1.000        26.98         52.00
 NETRENTX    -0.7372       -33.22        0.000      -0.5499E+05    9644.        0.1148E+06
 OTHREGBX     -1.127       -12.95        0.000        488.0      0.1985E+05     0.5000E+05
 OTHREGX     -0.7990       -20.33        0.000        100.0        9602.        0.6367E+05
 ROYESTX     -0.8014       -41.21        0.000        30.00      0.2176E+05     0.1592E+06
 STOCKX       0.2605E-02    6.982        0.000        25.00      0.4396E+06     0.6587E+07
 INTRDVX mean = 4138.86
 Predicted values truncated at 1.00000 & 98338.0
 ---------------------------
```

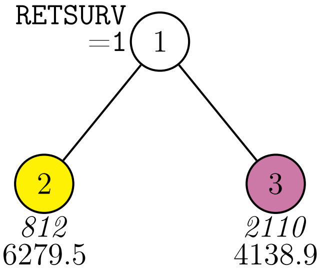

Figure 14: GUIDE v.38.0 0.25-SE piecewise linear weighted least-squares regression tree with stepwise variable selection for predicting INTRDVX. Tree constructed with 2922 observations (excluding observations with non-positive weight or missing values in D, T, R or Z variables). Maximum number of split levels is 12 and minimum node sample size is 25. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (*in italics*) and mean of INTRDVX printed below nodes. Terminal nodes with means above and below value of 4696.6 at root node are colored yellow and purple respectively. Second best split variable at root node is RETSURVX.

```
Proportion of variance (R-squared) explained by tree model: 0.8878

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

8 Quantile regression: CE data

GUIDE can build piecewise linear quantile regression models. We first show how to build a piecewise constant 0.50-quantile regression model.

8.1 Piecewise constant: one quantile

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: quantcon.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: quantcon.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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple polynomial in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
1: multiple linear, 2: simple polynomial, 3: constant ([1:3], <cr>=3): 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):
Input quantile probability ([0.00:1.00], <cr>=0.50):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: cereg.dsc
Reading data description file ...
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
409 N variables changed to S
D variable is INTRDVX
Reading data file ...
Number of records in data file: 4693
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 44 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
```

```
Creating missing value indicators ...
Rereading data ...
Warning: S variable OTHRINCB is constant
Warning: S variable NETRENTB is constant
Warning: S variable NETRNTBX is constant
Warning: S variable OTHLONBX is constant
Warning: S variable OTHLONB is constant
Smallest positive weight:  1.3507E+03
Largest positive weight:   7.0269E+04
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693       1771       4693       30        0        0      409
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      168        0       44        0
Number of cases used for training: 2922
Number of split variables: 453
Number of cases excluded due to 0 weight or missing D: 1771
Finished reading data file
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction tests; too many predictor variables
Warning: All positive weights treated as 1
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): quantcon.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: quantcon.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input rank of top variable to split root node ([1:453], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < quantcon.in
```

Contents of quantcon.out

```
Quantile regression tree with quantile probability 0.5000
Pruning by cross-validation
Data description file: cereg.dsc
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
409 N variables changed to S
D variable is INTRDVX
Piecewise constant model
Number of records in data file: 4693
Length of longest entry in data file: 11
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Warning: S variable OTHRINCB is constant
Warning: S variable NETRENTB is constant
Warning: S variable NETRNTBX is constant
Warning: S variable OTHLONBX is constant
Warning: S variable OTHLONB is constant
Smallest and largest positive weights are  1.3507E+03 and 7.0269E+04

Summary information for training sample of size 2922 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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
Levels of M variables are for missing values in associated variables
                                                    #Codes/
                                                    Levels/
 Column  Name           Minimum      Maximum    Periods   #Missing
      1  DIRACC     c                                 2      116
      2  DIRACC_    m                                 1
      3  AGE_REF    s    18.00        87.00
      4  AGE_REF_   m                                 0
      :
     50  FINLWT21   w    1351.       0.7027E+05
      :
    513  INTRDVX    d    1.000       0.9834E+05
      :
    651  FSTAXOWE   s   -2505.       0.5991E+05
    652  FSTA_OWE   m                                 0
    653  ETOTA      s    1199.       0.2782E+06

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693       1771       4693       30        0        0      409
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      168        0       44        0
Number of cases used for training: 2922
Number of split variables: 453
Number of cases excluded due to 0 weight or missing D: 1771

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500
```

```
Weighted error estimates used for pruning
Warning: No interaction tests; too many predictor variables
Warning: All positive weights treated as 1
No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 12
Minimum node sample size: 29
Top-ranked variables and chi-squared values at root node
     1  0.1728E+03   CUTENURE
     2  0.1492E+03   AGE_REF
     :
   410  0.5957E-03   TFOODTOC
   411  0.1145E-06   MENBOYPQ

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       79   4.414E+07   3.246E+06   1.855E+06   4.307E+07   1.302E+06
   2       78   4.414E+07   3.246E+06   1.855E+06   4.307E+07   1.302E+06
   :
  32+      32   4.405E+07   3.251E+06   1.865E+06   4.305E+07   1.212E+06
  33       31   4.412E+07   3.260E+06   1.870E+06   4.328E+07   1.165E+06
  34       30   4.411E+07   3.260E+06   1.870E+06   4.327E+07   1.164E+06
  35       29   4.411E+07   3.260E+06   1.869E+06   4.332E+07   1.167E+06
  36       27   4.401E+07   3.260E+06   1.889E+06   4.329E+07   1.196E+06
  37       25   4.398E+07   3.262E+06   1.895E+06   4.326E+07   1.247E+06
  38       24   4.398E+07   3.262E+06   1.895E+06   4.328E+07   1.249E+06
  39       22   4.400E+07   3.263E+06   1.899E+06   4.328E+07   1.246E+06
  40       20   4.389E+07   3.270E+06   1.939E+06   4.320E+07   1.325E+06
  41*      17   4.386E+07   3.274E+06   1.959E+06   4.318E+07   1.321E+06
  42       15   4.398E+07   3.280E+06   1.946E+06   4.331E+07   1.240E+06
  43++     14   4.400E+07   3.288E+06   1.987E+06   4.331E+07   1.357E+06
  44--     13   4.404E+07   3.292E+06   1.983E+06   4.341E+07   1.375E+06
  45       12   4.443E+07   3.299E+06   1.928E+06   4.391E+07   1.485E+06
  46**      7   4.456E+07   3.323E+06   1.925E+06   4.391E+07   1.481E+06
  47        6   4.470E+07   3.330E+06   1.874E+06   4.409E+07   1.240E+06
  48        1   4.558E+07   3.377E+06   1.823E+06   4.526E+07   1.219E+06

0-SE tree based on mean is marked with * and has 17 terminal nodes
0-SE tree based on median is marked with + and has 32 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-quant is quantile of INTRDVX in the node
Cases fit give the number of cases used to fit node
       Node    Total    Cases Matrix    Node    Split          Other
       label   cases      fit  rank    D-quant  variable       variables
          1     2922     2922     1  1.120E+02  CUTENURE
          2     1172     1172     1  4.100E+02  RENTEQVX
          4T     829      829     1  2.000E+02  OCCUCOD2
          5      343      343     1  3.000E+03  CHILDAGE
         10      286      286     1  4.800E+03  AGE_REF
         20      101      101     1  9.000E+03  HEALTHPQ
         40T      32       32     1  6.000E+02  -
         41T      69       69     1  1.500E+04  BEDROOMQ
         21      185      185     1  3.000E+03  TOTEXPPQ
         42T     143      143     1  2.000E+03  EDUC_REF
         43T      42       42     1  1.160E+04  -
         11T      57       57     1  5.610E+02  -
          3T    1750     1750     1  6.000E+01  STATE

Number of terminal nodes of final tree: 7
Total number of nodes of final tree: 13
Second best split variable (based on curvature test) at root node is AGE_REF

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: CUTENURE = "2"
  Node 2: RENTEQVX <= 1707.0000 or NA
    Node 4: INTRDVX sample quantile = 200.00000
  Node 2: RENTEQVX > 1707.0000
    Node 5: CHILDAGE <= 0.50000000
      Node 10: AGE_REF <= 63.500000
        Node 20: HEALTHPQ <= 341.50000
          Node 40: INTRDVX sample quantile = 600.00000
        Node 20: HEALTHPQ > 341.50000 or NA
          Node 41: INTRDVX sample quantile = 15000.000
      Node 10: AGE_REF > 63.500000 or NA
        Node 21: TOTEXPPQ <= 14270.100
          Node 42: INTRDVX sample quantile = 2000.0000
        Node 21: TOTEXPPQ > 14270.100 or NA
          Node 43: INTRDVX sample quantile = 11601.000
    Node 5: CHILDAGE > 0.50000000 or NA
      Node 11: INTRDVX sample quantile = 561.00000
Node 1: CUTENURE /= "2"
  Node 3: INTRDVX sample quantile = 60.000000
```

```
 ***************************************************************

 Predictor means below are weighted means of cases with no missing values.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if CUTENURE = "2"
 CUTENURE mode = "1"
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if RENTEQVX <= 1707.0000 or NA
 RENTEQVX mean = 1398.0139
 ----------------------------
Node 4: Terminal node
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if CHILDAGE <= 0.50000000
 CHILDAGE mean = 0.89016103
 ----------------------------
Node 10: Intermediate node
 A case goes into Node 20 if AGE_REF <= 63.500000
 AGE_REF mean = 67.265656
 ----------------------------
Node 20: Intermediate node
 A case goes into Node 40 if HEALTHPQ <= 341.50000
 HEALTHPQ mean = 1337.8140
 ----------------------------
Node 40: Terminal node
 ----------------------------
Node 41: Terminal node
 ----------------------------
Node 21: Intermediate node
 A case goes into Node 42 if TOTEXPPQ <= 14270.100
 TOTEXPPQ mean = 12283.417
 ----------------------------
Node 42: Terminal node
```

```
 ----------------------------
Node 43: Terminal node
 ----------------------------
Node 11: Terminal node
 ----------------------------
Node 3: Terminal node
 ----------------------------
 Observed and fitted values are stored in quantcon.fit
 LaTeX code for tree is in quantcon.tex
```

Figure 15 shows the quantile regression tree. The sample size (in *italics*) and 0.50-quantile are given beneath each terminal node. The split CHILDAGE $\leq$ 0.50 at node 5 indicates no children (see Table 11).

8.2 Simple linear

Here we show how to build a quantile regression tree that fits a simple linear (i.e., only one regressor) 0.90-quantile regression model in each node.

8.2.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: quantlin.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: quantlin.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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple polynomial in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
1: multiple linear, 2: simple polynomial, 3: constant ([1:3], <cr>=3): 2
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input quantile probability ([0.00:1.00], <cr>=0.50): 0.90
```

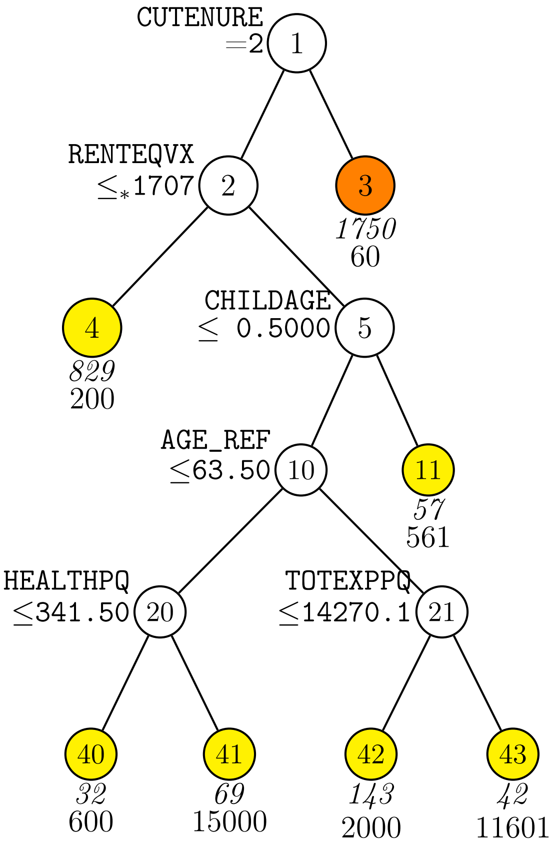

Figure 15: GUIDE v.38.0 0.25-SE piecewise constant 0.500-quantile regression tree for predicting INTRDVX. Tree constructed with 2922 observations (excluding observations with non-positive weight or missing values in D, T, R or Z variables). Maximum number of split levels is 12 and minimum node sample size is 29. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '≤ or missing'. Sample size (*in italics*) and 0.500-quantile of INTRDVX printed below nodes. Terminal nodes with quantiles above and below value of 112 at root node are colored yellow and orange respectively. Second best split variable at root node is AGE_REF.

```
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: cereg.dsc
Reading data description file ...
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
D variable is INTRDVX
Reading data file ...
Number of records in data file: 4693
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 44 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: N variable OTHRINCB is constant
Warning: N variable NETRENTB is constant
Warning: N variable NETRNTBX is constant
Warning: N variable OTHLONBX is constant
Warning: N variable OTHLONB is constant
Smallest positive weight:  1.3507E+03
Largest positive weight:   7.0269E+04
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693       1771       4693       30      409        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      168        0       44        0
Number of cases used for training: 2922
Number of split variables: 453
Number of cases excluded due to 0 weight or missing D: 1771
Finished reading data file
```

```
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction tests; too many predictor variables
Warning: All positive weights treated as 1
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): quantlin.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: quantlin.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input rank of top variable to split root node ([1:453], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < quantlin.in
```

Contents of `quantlin.out`

```
Quantile regression tree with quantile probability 0.9000
No truncation of predicted values
Pruning by cross-validation
Data description file: cereg.dsc
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
D variable is INTRDVX
Piecewise simple linear or constant model
Powers are dropped if they are not significant at level 1.0000
Number of records in data file: 4693
Length of longest entry in data file: 11
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Warning: N variable OTHRINCB is constant
Warning: N variable NETRENTB is constant
Warning: N variable NETRNTBX is constant
Warning: N variable OTHLONBX is constant
Warning: N variable OTHLONB is constant
Smallest and largest positive weights are  1.3507E+03 and 7.0269E+04

Summary information for training sample of size 2922 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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
Levels of M variables are for missing values in associated variables
```

```
                                                        #Codes/
                                                        Levels/
 Column  Name            Minimum      Maximum     Periods   #Missing
      1  DIRACC     c                                   2      116
      2  DIRACC_    m                                   1
      3  AGE_REF    n    18.00        87.00
      4  AGE_REF_   m                                   0
      5  AGE2       n   2.2000E+01    87.00                   1225
      6  AGE2_      m                                   1
      :
     50  FINLWT21   w    1351.       0.7027E+05
      :
    513  INTRDVX    d    1.000       0.9834E+05
      :
    652  FSTA_OWE   m                                   0
    653  ETOTA      n    1199.       0.2782E+06

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693       1771       4693       30      409        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      168        0       44        0
Number of cases used for training: 2922
Number of split variables: 453
Number of cases excluded due to 0 weight or missing D: 1771

Missing values imputed with node means for fitting regression models in nodes
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: .2500

Weighted error estimates used for pruning
Warning: No interaction tests; too many predictor variables
Warning: All positive weights treated as 1
No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 12
Minimum node sample size: 30
Top-ranked variables and chi-squared values at root node
     1  0.1527E+03   STOCKX
     2  0.1405E+03   STOCKYRX
     3  0.1088E+03   STATE
     :
   388  0.1395E-02   TOTHENTP
   389  0.1462E-04   EDUCAPQ
```

```
Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       72   3.922E+07   3.212E+06   3.368E+06   3.606E+07   4.119E+06
   2       71   3.922E+07   3.212E+06   3.369E+06   3.605E+07   4.120E+06
   :
  34       19   3.947E+07   3.215E+06   3.599E+06   3.707E+07   3.773E+06
  35**     18   3.911E+07   3.191E+06   3.571E+06   3.579E+07   3.875E+06
  36--     16   3.996E+07   3.214E+06   3.410E+06   3.738E+07   3.877E+06
  37       15   4.075E+07   3.283E+06   3.553E+06   3.738E+07   4.211E+06
  38++     14   4.097E+07   3.332E+06   4.088E+06   3.583E+07   4.363E+06
  39       13   4.312E+07   3.342E+06   3.846E+06   3.970E+07   5.616E+06
  40       11   4.272E+07   3.288E+06   3.755E+06   3.913E+07   5.526E+06
  41       10   4.263E+07   3.280E+06   3.767E+06   3.866E+07   5.513E+06
  42        9   4.245E+07   3.271E+06   3.830E+06   3.866E+07   5.500E+06
  43        8   4.257E+07   3.348E+06   3.856E+06   4.115E+07   5.712E+06
  44        7   4.628E+07   3.619E+06   3.797E+06   4.540E+07   5.506E+06
  45        5   4.681E+07   3.587E+06   4.079E+06   4.540E+07   6.618E+06
  46        4   4.940E+07   3.576E+06   3.320E+06   4.786E+07   4.287E+06
  47        3   6.321E+07   5.016E+06   3.154E+06   6.488E+07   5.080E+06
  48        1   6.513E+07   5.141E+06   3.249E+06   6.488E+07   5.117E+06

0-SE tree based on mean is marked with * and has 18 terminal nodes
0-SE tree based on median is marked with + and has 18 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 INTRDVX in the node
Cases fit give the number of cases used to fit node
       Node    Total    Cases Matrix    Node    Split          Other
       label   cases      fit   rank   D-quant  variable       variables
          1     2922     2922     2  9.500E+03  STOCKX
          2     2797     2797     2  7.582E+03  FINCATAX
          4     2024     2024     2  4.000E+03  INCNONW1
          8      706      706     2  1.200E+04  FFTAXOWE
         16T     365      365     2  4.000E+03  PSU
         17      341      341     2  2.100E+04  FRRETIRX
         34T      52       52     2  3.200E+04  -
         35      289      289     2  1.578E+04  FRRETIRX
         70T     134      134     2  1.200E+04  GASMOCQ
         71      155      155     2  2.100E+04  STATE
        142T      48       48     2  2.862E+04  -
        143T     107      107     2  1.200E+04  PERSCACQ
          9     1318     1318     2  1.418E+03  STATE
         18T     262      262     2  8.000E+03  SLOCTAXX
         19T    1056     1056     2  8.000E+02  EMRTPNOP
          5      773      773     2  2.206E+04  STATE
         10      107      107     2  9.834E+04  HIGH_EDU
         20       60       60     2  3.000E+04  OCCUCOD1
         40T      30       30     2  9.834E+04  -
         41T      30       30     2  1.328E+04  -
         21T      47       47     2  9.834E+04  -
         11      666      666     2  1.194E+04  AGE_REF
         22      585      585     2  5.500E+03  CUTENURE
         44      145      145     2  1.500E+04  NO_EARNR
         88T      37       37     2  9.834E+04  -
         89T     108      108     2  9.000E+03  SLRFUNDX
         45T     440      440     2  2.500E+03  FEDTAXX
         23       81       81     2  9.834E+04  FEDRFNDX
         46T      41       41     2  9.834E+04  -
         47T      40       40     2  2.200E+04  -
          3      125      125     2  9.834E+04  STOCKX
          6       94       94     2  2.400E+04  EOTHLODP
         12T      63       63     2  1.300E+04  CASHCOPQ
         13T      31       31     2  3.000E+04  -
          7T      31       31     2  9.834E+04  -

Number of terminal nodes of final tree: 18
Total number of nodes of final tree: 35
Second best split variable (based on curvature test) at root node is STOCKYRX

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: STOCKX <= 199500.00  or STOCKX = NA & STOCKX_ = "A"
  Node 2: FINCATAX <= 114116.50
    Node 4: INCNONW1 = "1", "5", "6"
      Node 8: FFTAXOWE <= 45.000000
        Node 16: INTRDVX sample quantile = 4000.0000
      Node 8: FFTAXOWE > 45.000000 or NA
        Node 17: FRRETIRX <= 833.50000
          Node 34: INTRDVX sample quantile = 32000.000
        Node 17: FRRETIRX > 833.50000 or NA
          Node 35: FRRETIRX <= 19265.000
            Node 70: INTRDVX sample quantile = 12000.000
          Node 35: FRRETIRX > 19265.000 or NA
            Node 71: STATE = "13", "23", "32", "41", "42", "45", "48", "8"
              Node 142: INTRDVX sample quantile = 28616.000
            Node 71: STATE /= "13", "23", "32", "41", "42", "45", "48", "8"
              Node 143: INTRDVX sample quantile = 12000.000
    Node 4: INCNONW1 /= "1", "5", "6"
      Node 9: STATE = "11", "15", "2", "23", "25", "26", "41", "48", "53", "8"
        Node 18: INTRDVX sample quantile = 8000.0000
      Node 9: STATE /= "11", "15", "2", "23", "25", "26", "41", "48", "53", "8"
        Node 19: INTRDVX sample quantile = 800.00000
  Node 2: FINCATAX > 114116.50 or NA
    Node 5: STATE = "18", "22", "26", "32", "33", "34", "45", "54", "8"
      Node 10: HIGH_EDU <= 15.500000
        Node 20: OCCUCOD1 = "10", "2", "3", "4"
          Node 40: INTRDVX sample quantile = 98338.000
        Node 20: OCCUCOD1 /= "10", "2", "3", "4"
          Node 41: INTRDVX sample quantile = 13277.000
      Node 10: HIGH_EDU > 15.500000 or NA
        Node 21: INTRDVX sample quantile = 98338.000
    Node 5: STATE /= "18", "22", "26", "32", "33", "34", "45", "54", "8"
      Node 11: AGE_REF <= 67.500000
        Node 22: CUTENURE = "2"
          Node 44: NO_EARNR <= 1.5000000
            Node 88: INTRDVX sample quantile = 98338.000
          Node 44: NO_EARNR > 1.5000000 or NA
            Node 89: INTRDVX sample quantile = 9000.0000
        Node 22: CUTENURE /= "2"
          Node 45: INTRDVX sample quantile = 2500.0000
      Node 11: AGE_REF > 67.500000 or NA
        Node 23: FEDRFNDX <= 92.500000  or FEDRFNDX = NA & FEDR_NDX = "A"
          Node 46: INTRDVX sample quantile = 98338.000
        Node 23: not (FEDRFNDX <= 92.500000 or FEDRFNDX = NA & FEDR_NDX = "A")
          Node 47: INTRDVX sample quantile = 22000.000
Node 1: not (STOCKX <= 199500.00 or STOCKX = NA & STOCKX_ = "A")
  Node 3: STOCKX <= 583000.00  or STOCKX = NA & STOCKX_ = "C"
    Node 6: EOTHLODP <= 55.000000
      Node 12: INTRDVX sample quantile = 13000.000
    Node 6: EOTHLODP > 55.000000 or NA
      Node 13: INTRDVX sample quantile = 30000.000
  Node 3: not (STOCKX <= 583000.00 or STOCKX = NA & STOCKX_ = "C")
    Node 7: INTRDVX sample quantile = 98338.000

***************************************************************

Predictor means below are weighted means of cases with no missing values.
Regression coefficients are computed from the complete cases.
```

```
 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if STOCKX <= 199500.00 or STOCKX_ = "A"
 STOCKX mean = 453208.43
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant      3099.
 STOCKX       0.1446E-01    25.00       0.4532E+06   0.6587E+07
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if FINCATAX <= 114116.50
 FINCATAX mean = 92406.011
:
:
 ----------------------------
Node 7: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant    -0.6982E+05
 INCLASS      0.1868E+05    5.000        8.472        9.000
 ----------------------------
 Observed and fitted values are stored in quantlin.fit
 LaTeX code for tree is in quantlin.tex
```

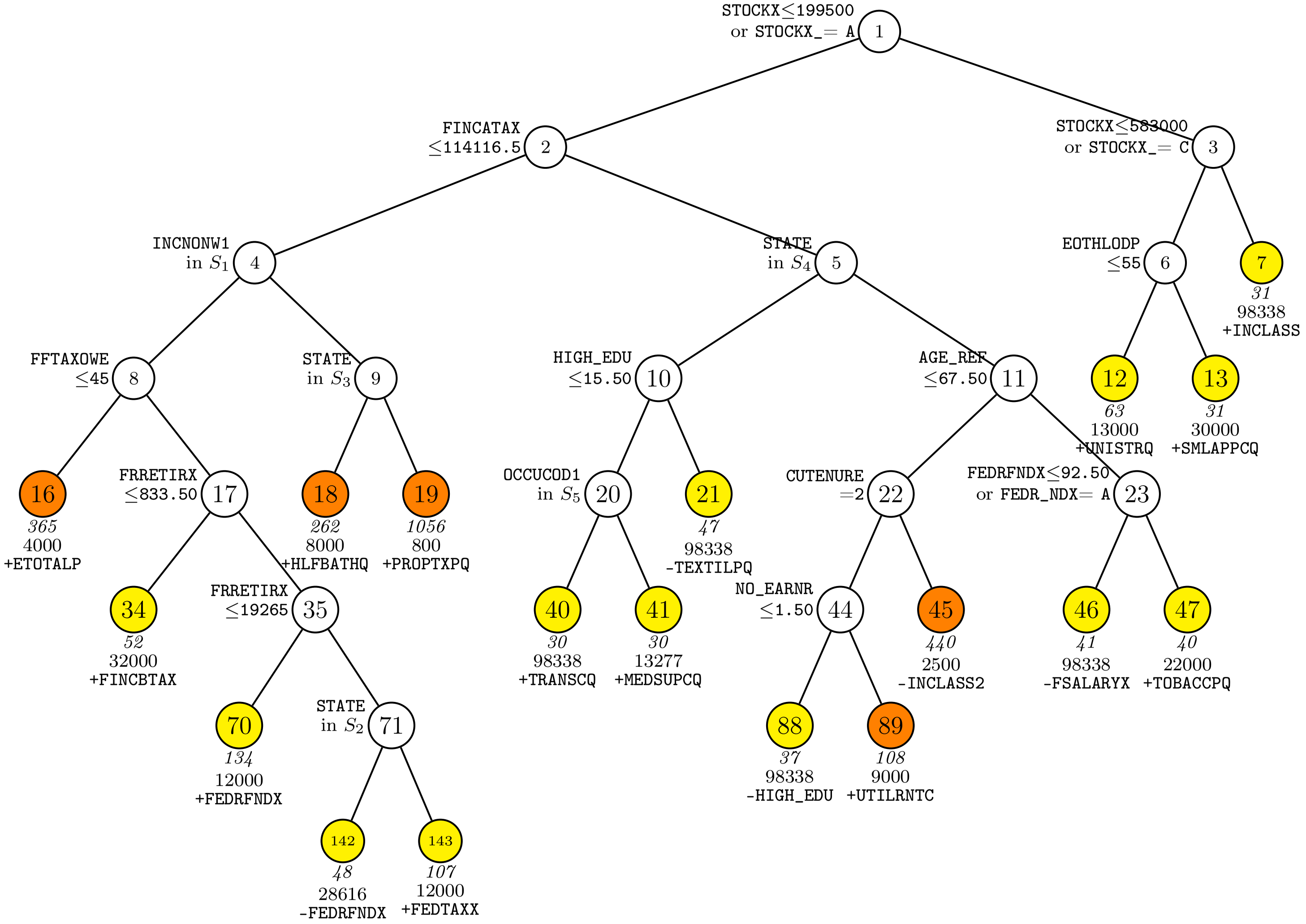

Figure 16: GUIDE v.38.0 0.25-SE piecewise simple linear 0.900-quantile regression tree for predicting INTRDVX. Tree constructed with 2922 observations (excluding observations with non-positive weight or missing values in D, T, R or Z variables). Maximum number of split levels is 12 and minimum node sample size is 30. At each split, an observation goes to the left branch if and only if the condition is satisfied. Set $S_1 = \{1, 5, 6\}$. Set $S_2 = \{13, 23, 32, 41, 42, 45, 48, 8\}$. Set $S_3 = \{11, 15, 2, 23, 25, 26, 41, 48, 53, 8\}$. Set $S_4 = \{18, 22, 26, 32, 33, 34, 45, 54, 8\}$. Set $S_5 = \{10, 2, 3, 4\}$. Sample size (*in italics*), 0.900-quantile of INTRDVX, and sign and name of best regressor printed below nodes. Terminal nodes with quantiles above and below value of 9500 at root node are colored yellow and orange respectively. Second best split variable at root node is STOCKYRX.

Figure 16 shows the 0.90-quantile regression tree.

8.3 Two quantiles: checking variance heterogeneity

Checking variance homogeneity in the residuals is a standard practice in fitting regression models. Here we demsnstrate how GUIDE can do this by constructing a quantile regression tree models for the 25th and 75th quantiles simultaneously.

8.3.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: twoquant.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: twoquant.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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple polynomial in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
1: multiple linear, 2: simple polynomial, 3: constant ([1:3], <cr>=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
Input 1st quantile probability ([0.00:1.00], <cr>=0.25):
Input 2nd quantile probability ([0.00:1.00], <cr>=0.75):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: cereg.dsc
Reading data description file ...
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
409 N variables changed to S
D variable is INTRDVX
Reading data file ...
Number of records in data file: 4693
```

```
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 44 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: S variable OTHRINCB is constant
Warning: S variable NETRENTB is constant
Warning: S variable NETRNTBX is constant
Warning: S variable OTHLONBX is constant
Warning: S variable OTHLONB is constant
Smallest positive weight:  1.3507E+03
Largest positive weight:   7.0269E+04
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals    #X-var   #N-var   #F-var   #S-var
      4693       1771       4693       30        0        0      409
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      168        0       44        0
Number of cases used for training: 2922
Number of split variables: 453
Number of cases excluded due to 0 weight or missing D: 1771
Finished reading data file
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction tests; too many predictor variables
Warning: All positive weights treated as 1
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): twoquant.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: twoquant.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input rank of top variable to split root node ([1:453], <cr>=1):
Input file is created!
```

```
Run GUIDE with the command: guide < twoquant.in
```

8.3.2 Output file

```
Dual-quantile regression tree with 0.2500 and 0.7500 quantiles
Pruning by cross-validation
Data description file: cereg.dsc
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
409 N variables changed to S
D variable is INTRDVX
Piecewise constant model
Number of records in data file: 4693
Length of longest entry in data file: 11
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Warning: S variable OTHRINCB is constant
Warning: S variable NETRENTB is constant
Warning: S variable NETRNTBX is constant
Warning: S variable OTHLONBX is constant
Warning: S variable OTHLONB is constant
Smallest and largest positive weights are  1.3507E+03 and 7.0269E+04

Summary information for training sample of size 2922 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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
Levels of M variables are for missing values in associated variables
                                                  #Codes/
                                                  Levels/
 Column  Name           Minimum      Maximum    Periods   #Missing
      1  DIRACC     c                                 2      116
      2  DIRACC_    m                                 1
      3  AGE_REF    s    18.00        87.00
      4  AGE_REF_   m                                 0
      :
     50  FINLWT21   w    1351.       0.7027E+05
      :
    513  INTRDVX    d    1.000       0.9834E+05
```

```
      :
    651  FSTAXOWE   s   -2505.       0.5991E+05
    652  FSTA_OWE   m                                  0
    653  ETOTA      s    1199.       0.2782E+06

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693       1771       4693       30        0        0      409
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      168        0       44        0
Number of cases used for training: 2922
Number of split variables: 453
Number of cases excluded due to 0 weight or missing D: 1771

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Weighted error estimates used for pruning
Warning: No interaction tests; too many predictor variables
Warning: All positive weights treated as 1
No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 12
Minimum node sample size: 29
Top-ranked variables and chi-squared values at root node
     1  0.1840E+03   AGE_REF
     2  0.1689E+03   CUTENURE
     3  0.1420E+03   RENTEQVX
     4  0.1393E+03   PERSOT64
     :
   410  0.1355E-02   TGASMOTC
   411  0.7307E-03   MAJAPPCQ

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       77   8.456E+07   6.167E+06   3.934E+06   8.212E+07   3.273E+06
   2       76   8.456E+07   6.167E+06   3.932E+06   8.212E+07   3.273E+06
   :
  36*      29   8.445E+07   6.189E+06   4.037E+06   8.194E+07   3.441E+06
  37       28   8.468E+07   6.200E+06   4.157E+06   8.193E+07   3.398E+06
  38+      24   8.475E+07   6.200E+06   4.136E+06   8.193E+07   3.384E+06
  39++     18   8.505E+07   6.208E+06   4.079E+06   8.260E+07   3.123E+06
  40       16   8.578E+07   6.265E+06   4.094E+06   8.472E+07   3.100E+06
  41**     14   8.556E+07   6.279E+06   4.184E+06   8.449E+07   3.392E+06
  42        3   8.694E+07   6.516E+06   3.972E+06   8.641E+07   2.607E+06
  43       1   8.957E+07   6.679E+06   3.534E+06   8.898E+07   2.373E+06

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 24 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    Total    Cases Matrix    Node    Split          Other
       label   cases      fit  rank     median  variable       variables
          1     2922     2922     1  2.000E+01  AGE_REF
          2T    1385     1385     1  1.200E+01  4.000E+02  STATE
          3     1537     1537     1  4.000E+01  STOCKX
          6     1507     1507     1  3.600E+01  STATE
         12T     503      503     1  1.500E+01  1.210E+03  EARNCOMP
         13     1004     1004     1  7.500E+01  RENTEQVX
         26T     181      181     1  4.300E+01  8.780E+02  RETSURVX
         27      823      823     1  9.600E+01  STATE
         54      315      315     1  4.300E+01  FFTAXOWE
        108      270      270     1  4.100E+01  FFTAXOWE
        216T      89       89     1  5.000E+01  1.765E+03  ELCTRCCQ
        217      181      181     1  3.000E+01  AGE_REF
        434T     145      145     1  2.000E+01  2.500E+03  OFSTPARK
        435T      36       36     1  1.392E+03  2.000E+04  -
        109T      45       45     1  2.150E+02  1.500E+04  -
         55      508      508     1  1.500E+02  INCOMEY1
        110T     132      132     1  8.250E+01  1.200E+03  PROPTXCQ
        111      376      376     1  2.000E+02  STATE
        222      246      246     1  1.500E+02  EMRTPNOP
        444      184      184     1  2.650E+02  TOTXEST
        888T      77       77     1  1.000E+02  4.034E+03  GASMOCQ
        889      107      107     1  4.500E+02  PERINSPQ
       1778T      51       51     1  2.400E+03  3.200E+04  -
       1779T      56       56     1  3.000E+02  7.750E+03  -
        445T      62       62     1  4.100E+01  5.000E+02  BUILT
        223T     130      130     1  5.400E+02  1.194E+04  STATE
          7T      30       30     1  1.160E+04  9.834E+04  -

Number of terminal nodes of final tree: 14
```

```
Total number of nodes of final tree: 27
Second best split variable (based on curvature test) at root node is CUTENURE

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: AGE_REF <= 56.500000
  Node 2: INTRDVX sample quantiles = 12.000000, 400.00000
Node 1: AGE_REF > 56.500000 or NA
  Node 3: STOCKX <= 583000.00 or NA
    Node 6: STATE = "11", "16", "17", "20", "21", "22", "27", "29", "31", "34",
             "36", "39", "4", "42", "47"
      Node 12: INTRDVX sample quantiles = 15.000000, 1210.0000
    Node 6: STATE /= "11", "16", "17", "20", "21", "22", "27", "29", "31", "34",
             "36", "39", "4", "42", "47"
      Node 13: RENTEQVX <= 742.00000 or NA
        Node 26: INTRDVX sample quantiles = 43.000000, 878.00000
      Node 13: RENTEQVX > 742.00000
        Node 27: STATE = "18", "23", "26", "45", "48", "49", "54", "55", "NA"
          Node 54: FFTAXOWE <= 19168.000
            Node 108: FFTAXOWE <= 19.500000
              Node 216: INTRDVX sample quantiles = 50.000000, 1765.0000
            Node 108: FFTAXOWE > 19.500000 or NA
              Node 217: AGE_REF <= 78.000000
                Node 434: INTRDVX sample quantiles = 20.000000, 2500.0000
              Node 217: AGE_REF > 78.000000 or NA
                Node 435: INTRDVX sample quantiles = 1391.5000, 20000.000
          Node 54: FFTAXOWE > 19168.000 or NA
            Node 109: INTRDVX sample quantiles = 215.00000, 15000.000
        Node 27: STATE /= "18", "23", "26", "45", "48", "49", "54", "55", "NA"
          Node 55: INCOMEY1 = "1"
            Node 110: INTRDVX sample quantiles = 82.500000, 1200.0000
          Node 55: INCOMEY1 /= "1"
            Node 111: STATE = "12", "13", "15", "51", "6", "9"
              Node 222: EMRTPNOP <= 213.00000
                Node 444: TOTXEST <= 150.50000
                  Node 888: INTRDVX sample quantiles = 100.00000, 4034.0000
                Node 444: TOTXEST > 150.50000 or NA
                  Node 889: PERINSPQ <= 9.6166500
                    Node 1778: INTRDVX sample quantiles = 2400.0000, 32000.000
                  Node 889: PERINSPQ > 9.6166500 or NA
                    Node 1779: INTRDVX sample quantiles = 300.00000, 7750.0000
              Node 222: EMRTPNOP > 213.00000 or NA
                Node 445: INTRDVX sample quantiles = 41.000000, 500.00000
            Node 111: STATE /= "12", "13", "15", "51", "6", "9"
              Node 223: INTRDVX sample quantiles = 540.00000, 11938.000
```

```
   Node 3: STOCKX > 583000.00
     Node 7: INTRDVX sample quantiles = 11601.000, 98338.000

 ***************************************************************

 Predictor means below are weighted means of cases with no missing values.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if AGE_REF <= 56.500000
 AGE_REF mean = 55.397812
 Sample 0.250-quantile, 0.750-quantile, and median:
     2.0000E+01     1.2100E+03     1.5000E+02
 ----------------------------
Node 2: Terminal node
 Sample 0.250-quantile, 0.750-quantile, and median:
     1.2000E+01     4.0000E+02     7.0000E+01
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if STOCKX <= 583000.00 or NA
 STOCKX mean = 782050.25
 ----------------------------
:
Node 445: Terminal node
 Sample 0.250-quantile, 0.750-quantile, and median:
     4.1000E+01     5.0000E+02     2.0000E+02
 ----------------------------
Node 223: Terminal node
 Sample 0.250-quantile, 0.750-quantile, and median:
     5.4000E+02     1.1938E+04     1.7560E+03
 ----------------------------
Node 7: Terminal node
 Sample 0.250-quantile, 0.750-quantile, and median:
     1.1601E+04     9.8338E+04     3.0150E+04
 ----------------------------
 Observed and fitted values are stored in twoquant.fit
 LaTeX code for tree is in twoquant.tex
```

Figure 17 shows the tree. Beneath each terminal node are three numbers. The first (in *italics*) is the node sample size. The other two are the sample 0.75 and 0.25-quantiles in the node. Based on the large between-node variations in the interquartile ranges in the nodes, it is clear that there is substantial variance heterogeneity.

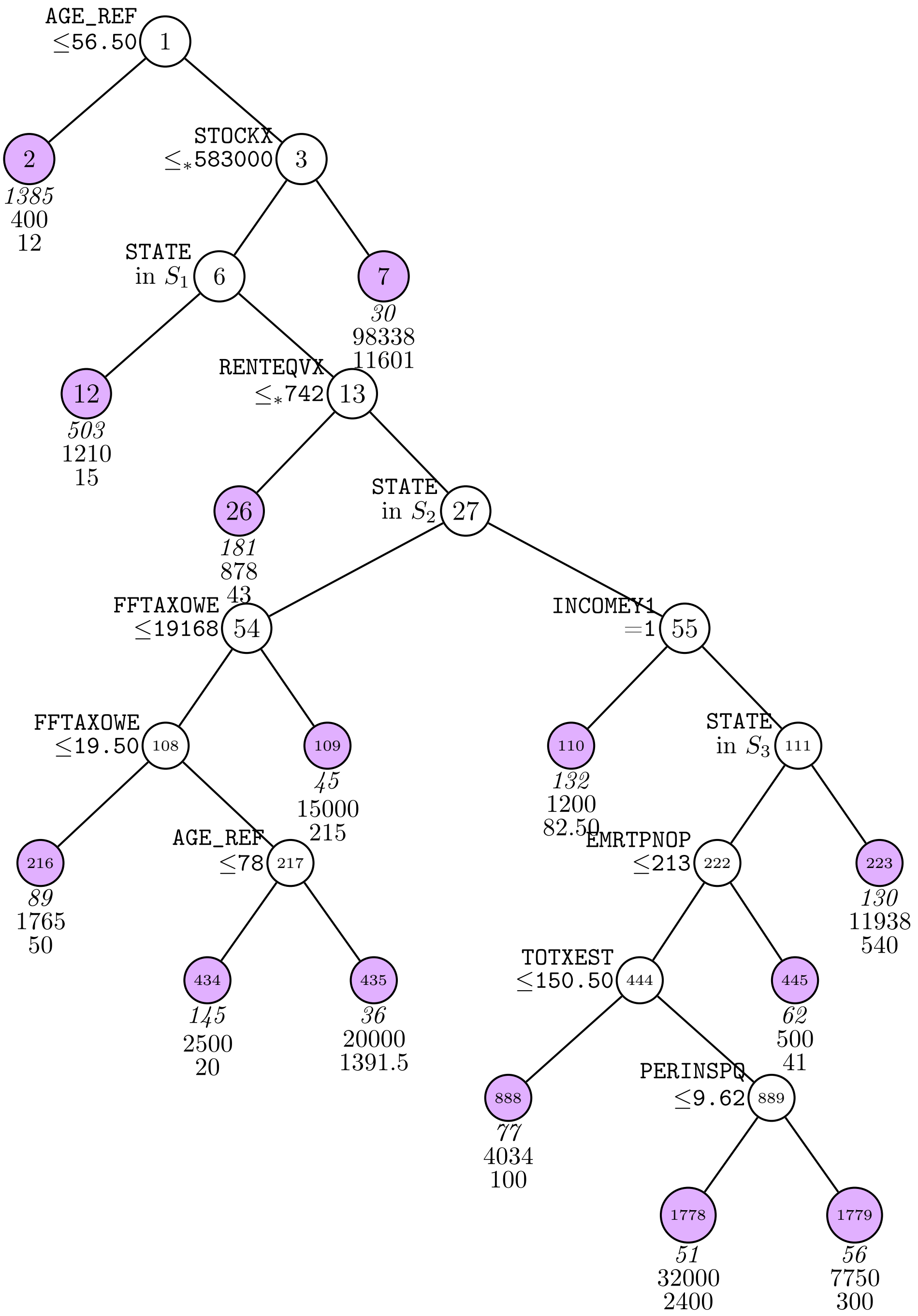

Figure 17: GUIDE v.38.0 0.25-SE piecewise constant 0.250 and 0.750-quantile regression tree for predicting `INTRDVX`. Sample size (*in italics*) and sample 0.750 and 0.250-quantiles of `INTRDVX` printed below nodes.

9 Poisson regression: solder data

We use a data set on printed circuit board soldering to show how GUIDE fits Poisson regression models. The data were analyzed in Chambers and Hastie (1992) and are given in `solder.dat`. The description file `solder.dsc` uses the `b` descriptor for the 5 categorical variables:

```
solder.dat
"?"
1
1, skips, d
2, opening, b
3, solder, b
4, mask, b
5, padtype, b
6, panel, b
```

9.1 Piecewise constant

9.1.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: 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
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
 Input choice ([1:7], <cr>=1): 3
 Choose complexity of model to use at each node:
 Choose 1 for multiple regression (not recommended if there are missing values)
 Choose 2 for simple polynomial in one N or F variable
 Choose 3 to fit a constant (recommended if there are missing values or an R variable)
 1: multiple linear, 2: simple polynomial, 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: solder.dsc
 Reading data description file ...
```

```
Training sample file: solder.dat
Missing value code: NA
Records in data file start on line 1
Warning: B variables changed to C
D variable is skips
Reading data file ...
Number of records in data file: 720
Length of longest entry in data file: 6
Checking for missing values ...
Finished checking
Assigning integer codes to values of 5 categorical variables
Re-checking data ...
Assigning codes to missing values if any ...
Data checks complete
Number of cases with positive D values: 478
Rereading data ...
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       720          0          0        0        0        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        5        0
No offset variable in data file.
Number of cases used for training: 720
Number of split variables: 5
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): cons.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: cons.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: cons.r
Input rank of top variable to split root node ([1:5], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < cons.in
```

The tree is shown in Figure 18, which is rather large. One way to reduce the size of the tree is to fit a more complex Poisson regression model in each node.

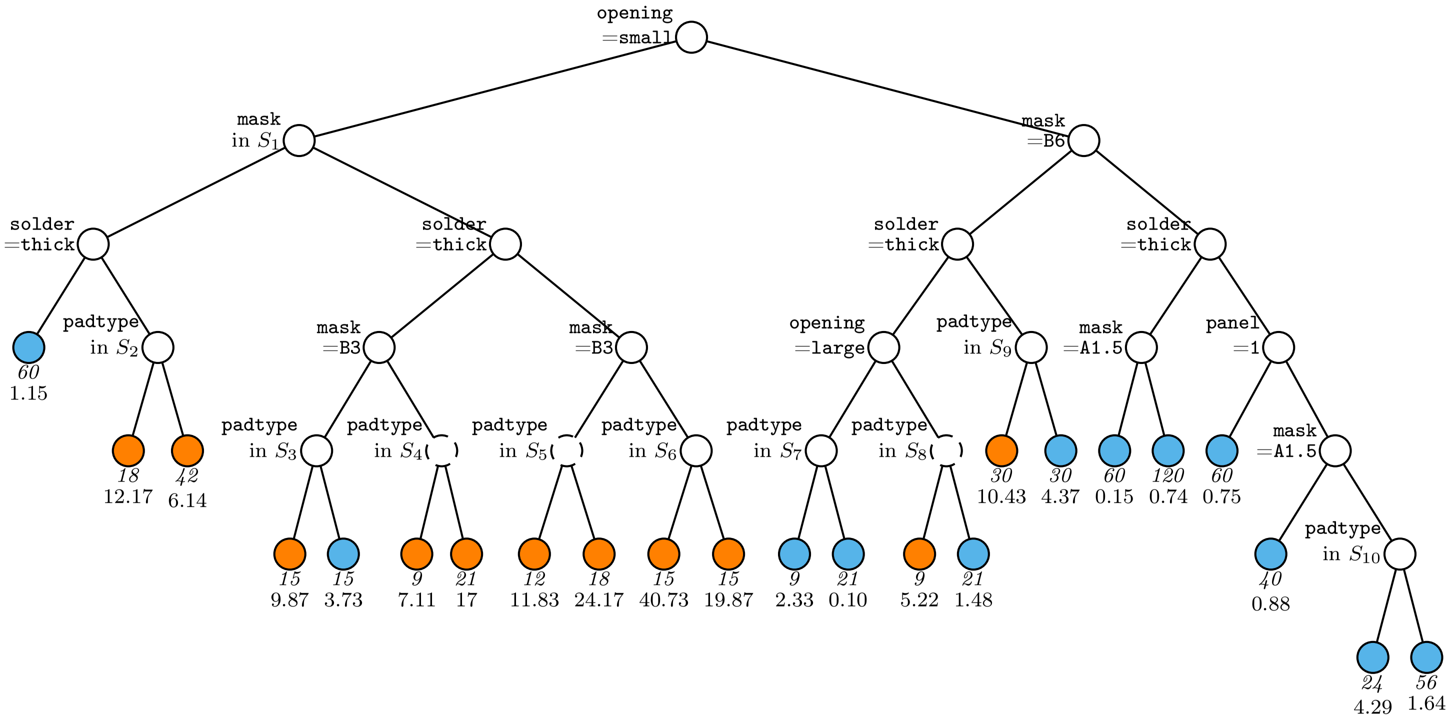

Figure 18: GUIDE v.38.0 0.25-SE piecewise constant Poisson regression tree for predicting `skips`. Tree constructed with 720 observations. Maximum number of split levels is 10 and minimum node sample size is 7. At each split, an observation goes to the left branch if and only if the condition is satisfied. Set $S_1 = \{$`A1.5`, `A3`$\}$. Set $S_2 = \{$`D4`, `D7`, `L4`$\}$. Set $S_3 = \{$`D4`, `D7`, `L4`, `L7`, `L8`$\}$. Set $S_4 = \{$`L6`, `L9`, `W9`$\}$. Set $S_5 = \{$`L6`, `L7`, `L9`, `W9`$\}$. Set $S_6 = \{$`D4`, `D6`, `D7`, `L4`, `W4`$\}$. Set $S_7 = \{$`D4`, `W4`, `W9`$\}$. Set $S_8 = \{$`D7`, `L4`, `L8`$\}$. Set $S_9 = \{$`D4`, `D7`, `L4`, `L8`, `W4`$\}$. Set $S_{10} = \{$`D4`, `D7`, `L4`$\}$. Circles with dashed lines denote nodes with no significant splits. Sample size (*in italics*) and mean of `skips` printed below nodes. Terminal nodes with means above and below value of 4.97 at root node are colored orange and skyblue, respectively. Second best split variable at root node is `mask`.

9.2 Multiple linear

Now we construct a tree where each node is fitted with a Poisson model containing only the main effects. This is where the "B" descriptor in solder.dsc is for.

9.2.1 Input file creation

```
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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 3
Choose complexity of model to use at each node:
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple polynomial in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
1: multiple linear, 2: simple polynomial, 3: constant ([1:3], <cr>=3): 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: solder.dsc
Reading data description file ...
Training sample file: solder.dat
Missing value code: NA
Records in data file start on line 1
D variable is skips
Reading data file ...
Number of records in data file: 720
Length of longest entry in data file: 6
Checking for missing values ...
Finished checking
Assigning integer codes to values of 5 categorical variables
Re-checking data ...
Assigning codes to missing values if any ...
Data checks complete
Number of cases with positive D values: 478
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: 17
Creating dummy variables ...
Rereading data ...
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       720          0          0        0        0        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        5        0        0
No offset variable in data file.
Number of cases used for training: 720
Number of split variables: 5
Number of dummy variables created: 17
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
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): 2
Input file name: mul.r
Input rank of top variable to split root node ([1:22], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < mul.in
```

9.2.2 Contents of mul.out

```
Poisson regression tree
No truncation of predicted values
Pruning by cross-validation
Data description file: solder.dsc
Training sample file: solder.dat
Missing value code: NA
Records in data file start on line 1
D variable is skips
Piecewise linear model
Number of records in data file: 720
Length of longest entry in data file: 6
Number of cases with positive D values: 478
Number of dummy variables created: 17

Summary information for training sample of size 720
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
```

```
                                                      #Codes/
                                                      Levels/
 Column  Name                 Minimum      Maximum    Periods   #Missing
      1  skips           d    0.000        48.00
      2  opening         b                                  3
      3  solder          b                                  2
      4  mask            b                                  4
      5  padtype         b                                 10
      6  panel           b                                  3
  =================== Constructed variables ===================
      7  opening.medium  f    0.000        1.000
      8  opening.small   f    0.000        1.000
      9  solder.thin     f    0.000        1.000
     10  mask.A3         f    0.000        1.000
     11  mask.B3         f    0.000        1.000
     12  mask.B6         f    0.000        1.000
     13  padtype.D6      f    0.000        1.000
     14  padtype.D7      f    0.000        1.000
     15  padtype.L4      f    0.000        1.000
     16  padtype.L6      f    0.000        1.000
     17  padtype.L7      f    0.000        1.000
     18  padtype.L8      f    0.000        1.000
     19  padtype.L9      f    0.000        1.000
     20  padtype.W4      f    0.000        1.000
     21  padtype.W9      f    0.000        1.000
     22  panel.2         f    0.000        1.000
     23  panel.3         f    0.000        1.000

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals    #X-var   #N-var   #F-var   #S-var
       720          0          0         0        0        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        5        0        0
No offset variable in data file.
Number of cases used for training: 720
Number of split variables: 5
Number of dummy variables created: 17

Missing values imputed with node means for fitting regression models in nodes
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 10
```

```
Minimum node sample size: 7
Top-ranked variables and chi-squared values at root node
     1  0.1782E+02   solder
     2  0.3481E+01   opening
     3  0.3357E+01   mask
     4  0.2453E+00   panel
     5  0.1361E+00   padtype

Size and CV Loss and SE of subtrees:
 Tree    #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       55    2.939E+00   1.916E-01   1.950E-01   2.852E+00   2.525E-01
   2       53    2.939E+00   1.916E-01   1.950E-01   2.852E+00   2.525E-01
   :
  36        4    1.488E+00   8.070E-02   8.672E-02   1.449E+00   7.036E-02
  37**      3    1.457E+00   7.447E-02   9.380E-02   1.343E+00   7.680E-02
  38        2    1.527E+00   7.949E-02   9.597E-02   1.455E+00   6.790E-02
  39        1    1.660E+00   8.239E-02   7.060E-02   1.651E+00   7.689E-02

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.

D-mean is mean of skips in the node
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      Other
 label    cases       fit  rank     D-mean     deviance  variable   variables
    1      720       720    18   4.965E+00  1.610E+00  solder
    2T     360       360    17   2.481E+00  1.279E+00  mask
    3      360       360    17   7.450E+00  1.628E+00  opening :mask  <- interaction
    6T     120       120    15   1.636E+01  1.367E+00  padtype
    7T     240       240    16   2.996E+00  1.403E+00  mask

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 opening

Regression tree:
For categorical variable splits, values not in training data go to the right
```

```
 Node 1: solder = "thick"
   Node 2: skips sample mean = 2.4805556
 Node 1: solder /= "thick"
   Node 3: opening = "small"
     Node 6: skips sample mean = 16.358333
   Node 3: opening /= "small"
     Node 7: skips sample mean = 2.9958333

 ***************************************************************


 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if solder = "thick"
 solder mode = "thick"
 Coefficients of loglinear regression function:
 Regressor         Coefficient  t-stat      p-value      Minimum          Mean       Maximum
 Constant          -1.220       -12.81       0.8882E-15
 mask.A3            0.4282        5.674       0.2043E-07    0.000       0.2500        1.000
 mask.B3            1.202        17.95        0.7772E-15    0.000       0.2500        1.000
 mask.B6            1.866        29.58         0.000        0.000       0.2500        1.000
 opening.medium     0.2585        3.884       0.1126E-03    0.000       0.3333        1.000
 opening.small      1.893        35.31        0.8882E-15    0.000       0.3333        1.000
 padtype.D6        -0.3687       -5.164       0.3144E-06    0.000       0.1000        1.000
 padtype.D7        -0.9844E-01   -1.487       0.1374        0.000       0.1000        1.000
 padtype.L4         0.2624        4.321       0.1774E-04    0.000       0.1000        1.000
 padtype.L6        -0.6685       -8.525        0.000        0.000       0.1000        1.000
 padtype.L7        -0.4902       -6.619       0.7177E-10    0.000       0.1000        1.000
 padtype.L8        -0.2712       -3.907       0.1023E-03    0.000       0.1000        1.000
 padtype.L9        -0.6365       -8.203       0.2220E-15    0.000       0.1000        1.000
 padtype.W4        -0.1100       -1.657       0.9804E-01    0.000       0.1000        1.000
 padtype.W9        -1.438        -13.80       0.4441E-15    0.000       0.1000        1.000
 panel.2            0.3335        7.929       0.9881E-14    0.000       0.3333        1.000
 panel.3            0.2544        5.947       0.4318E-08    0.000       0.3333        1.000
 solder.thin        1.100        28.46         0.000        0.000       0.5000        1.000
 ----------------------------
Node 2: Terminal node
 Coefficients of loglinear regression function:
 Regressor          Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant           -2.431       -10.68        0.000
 mask.A3            0.4670        2.373      0.1820E-01    0.000        0.2500         1.000
 mask.B3             1.831        11.01        0.000       0.000        0.2500         1.000
 mask.B6             2.520        15.71        0.000       0.000        0.2500         1.000
 opening.medium     0.8641        5.567      0.5228E-07    0.000        0.3333         1.000
 opening.small       2.465        18.18        0.000       0.000        0.3333         1.000
 padtype.D6        -0.3238       -2.034      0.4274E-01    0.000        0.1000         1.000
 padtype.D7         0.1201       0.8480      0.3970        0.000        0.1000         1.000
 padtype.L4         0.6985        5.534      0.6221E-07    0.000        0.1000         1.000
 padtype.L6        -0.4002       -2.458      0.1448E-01    0.000        0.1000         1.000
 padtype.L7         0.4167E-01   0.2887      0.7730        0.000        0.1000         1.000
 padtype.L8         0.1481        1.052      0.2936        0.000        0.1000         1.000
 padtype.L9        -0.5921       -3.426      0.6877E-03    0.000        0.1000         1.000
 padtype.W4        -0.5466E-01  -0.3696      0.7119        0.000        0.1000         1.000
 padtype.W9         -1.324       -5.886      0.9394E-08    0.000        0.1000         1.000
 panel.2            0.2224        2.718      0.6895E-02    0.000        0.3333         1.000
 panel.3            0.6825E-01   0.8049      0.4214        0.000        0.3333         1.000
 solder.thin         0.000        0.000        1.000       0.000         0.000         0.000
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if opening = "small"
 opening mode = "large"
 ----------------------------
Node 6: Terminal node
 Coefficients of loglinear regression function:
 Regressor          Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant            2.080        21.50        0.000
 mask.A3            0.3085        3.329      0.1202E-02    0.000        0.2500         1.000
 mask.B3             1.050        12.84        0.000       0.000        0.2500         1.000
 mask.B6             1.504        19.34        0.000       0.000        0.2500         1.000
 opening.medium      0.000        0.000        1.000       0.000         0.000         0.000
 opening.small       0.000        0.000        1.000       1.000         1.000         1.000
 padtype.D6        -0.2534       -2.788      0.6302E-02    0.000        0.1000         1.000
 padtype.D7        -0.1476       -1.671      0.9763E-01    0.000        0.1000         1.000
 padtype.L4         0.8309E-01   0.9980      0.3206        0.000        0.1000         1.000
 padtype.L6        -0.7187       -6.847      0.4730E-09    0.000        0.1000         1.000
 padtype.L7        -0.6473       -6.315      0.6560E-08    0.000        0.1000         1.000
 padtype.L8        -0.4255       -4.452      0.2127E-04    0.000        0.1000         1.000
 padtype.L9        -0.6404       -6.262      0.8418E-08    0.000        0.1000         1.000
 padtype.W4        -0.8668E-01  -0.9978      0.3207        0.000        0.1000         1.000
 padtype.W9         -1.376       -10.29        0.000       0.000        0.1000         1.000
 panel.2            0.3070        5.470      0.3070E-06    0.000        0.3333         1.000
 panel.3           0.1850        3.210       0.1762E-02    0.000       0.3333        1.000
 solder.thin        0.000        0.000        1.000        1.000        1.000        1.000
 ----------------------------
Node 7: Terminal node
 Coefficients of loglinear regression function:
 Regressor         Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant         -0.3711       -1.947       0.5284E-01
 mask.A3           0.8061        4.546       0.8965E-05    0.000       0.2500        1.000
 mask.B3            1.008        5.849       0.1735E-07    0.000       0.2500        1.000
 mask.B6            2.267        14.64       0.2220E-15    0.000       0.2500        1.000
 opening.medium    0.1030        1.379       0.1692        0.000       0.5000        1.000
 opening.small      0.000        0.000        1.000        0.000        0.000        0.000
 padtype.D6       -0.7995       -4.649       0.5709E-05    0.000       0.1000        1.000
 padtype.D7       -0.1915       -1.345       0.1800        0.000       0.1000        1.000
 padtype.L4        0.2065        1.601       0.1108        0.000       0.1000        1.000
 padtype.L6       -0.8201       -4.735       0.3894E-05    0.000       0.1000        1.000
 padtype.L7       -0.7595       -4.477       0.1206E-04    0.000       0.1000        1.000
 padtype.L8       -0.3606       -2.413       0.1662E-01    0.000       0.1000        1.000
 padtype.L9       -0.6660       -4.051       0.7039E-04    0.000       0.1000        1.000
 padtype.W4       -0.2254       -1.568       0.1183        0.000       0.1000        1.000
 padtype.W9        -1.747       -7.027       0.2514E-10    0.000       0.1000        1.000
 panel.2           0.5841        5.732       0.3190E-07    0.000       0.3333        1.000
 panel.3           0.6931        6.931       0.4388E-10    0.000       0.3333        1.000
 solder.thin        0.000        0.000        1.000        1.000        1.000        1.000
 ----------------------------
 Observed and fitted values are stored in mul.fit
 LaTeX code for tree is in mul.tex
 R code is stored in mul.r
```

Figure 19 shows the tree, which is much shorter than that in Figure 18. Note that node 3 has a different color (wheat) to indicate that the split there is due to an interaction between two variables (opening and mask); this is indicated by the blue comment <− interaction in the contents of mul.out above.

9.3 With offset variable: 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). The 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 an an offset variable and the former is excluded (x) from the analysis. For the purpose of illustration, we specify sex as

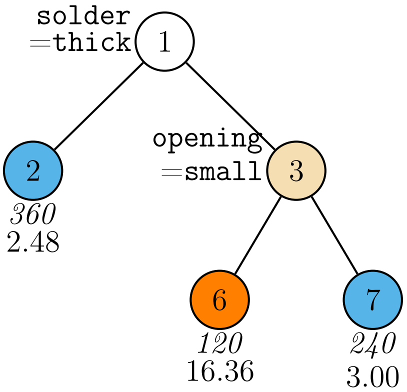

Figure 19: GUIDE v.38.0 0.25-SE multiple linear Poisson regression tree for predicting `skips`. Tree constructed with 720 observations. Maximum number of split levels is 10 and minimum node sample size is 7. At each split, an observation goes to the left branch if and only if the condition is satisfied. Intermediate nodes with splits due to interaction are in wheat color. Sample size (*in italics*) and mean of `skips` printed below nodes. Terminal nodes with means above and below value of 4.97 at root node are colored orange and skyblue respectively. Second best split variable at root node is `opening`.

`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:

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

9.3.1 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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 3
Choose complexity of model to use at each node:
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple polynomial in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
1: multiple linear, 2: simple polynomial, 3: constant ([1:3], <cr>=3): 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
D 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 ...
Finished checking
Assigning integer codes to values of 3 categorical variables
Re-checking data ...
Assigning codes to missing values if any ...
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 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
 Number of cases used for training: 920
 Number of split variables: 3
 Number of dummy variables created: 1
 Finished reading data file
 Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
 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): 2
 Input file name: poi.r
 Input rank of top variable to split root node ([1:4], <cr>=1):
 Input file is created!
 Run GUIDE with the command: guide < poi.in
```

9.3.2 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
 D 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 for training sample of size 920
 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
                                                   #Codes/
                                                   Levels/
  Column  Name           Minimum      Maximum    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
Number of cases used for training: 920
Number of split variables: 3
Number of dummy variables created: 1

Missing values imputed with node means for fitting regression models in nodes
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 10
Minimum node sample size: 7
Top-ranked variables and chi-squared values at root node
     1  0.2986E+03   agegp
     2  0.1574E+02   sex
     3  0.7551E-02   county

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       53   3.415E+00   4.129E-01   2.823E-01   3.274E+00   5.015E-01
   2       52   3.415E+00   4.129E-01   2.823E-01   3.274E+00   5.015E-01
   :
  36        4   2.282E+00   3.054E-01   2.517E-01   1.953E+00   3.691E-01
  37**      3   2.230E+00   3.278E-01   2.720E-01   1.910E+00   2.934E-01
  38        2   4.702E+00   8.054E-01   4.866E-01   4.153E+00   6.629E-01
  39        1   9.431E+00   1.420E+00   9.674E-01   9.043E+00   9.329E-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 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    Total    Cases Matrix    Node       Node    Split          Other
       label   cases      fit  rank     rate     deviance  variable       variables
          1      920      920     2  1.382E-02  9.179E+00  agegp
          2T     230      230     2  5.493E-03  1.863E+00  county
          3      690      690     2  1.763E-02  4.357E+00  agegp
          6T     230      230     2  1.339E-02  3.003E+00  county
          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:
For categorical variable splits, values not in training data go to the right

Node 1: agegp = "45-54"
  Node 2: deaths sample rate = 0.54928582E-2
Node 1: agegp /= "45-54"
  Node 3: agegp = "55-64"
    Node 6: deaths sample rate = 0.13389777E-1
  Node 3: agegp /= "55-64"
    Node 7: deaths sample rate = 0.20932715E-1

***************************************************************


WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.
```

```
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     -5.172       -366.9        0.000
 sex.M         1.437        89.64        0.000        0.000       0.5000        1.000
 Node mean for offset variable =    6.727
 ----------------------------
Node 2: Terminal node
 Coefficients of loglinear regression function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -5.834       -161.5       0.3331E-15
 sex.M         1.038        24.44       0.2220E-15    0.000       0.5000        1.000
 Node mean for offset variable =    6.857
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if agegp = "55-64"
 agegp mode = "55-64"
 ----------------------------
Node 6: Terminal node
 Coefficients of loglinear regression function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -5.117       -199.8        0.000
 sex.M         1.285        43.87        0.000        0.000       0.5000        1.000
 Node mean for offset variable =    6.920
 ----------------------------
Node 7: Terminal node
 Coefficients of loglinear regression function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -4.907       -256.9        0.000
 sex.M         1.714        79.68       0.2220E-15    0.000       0.5000        1.000
 Node mean for offset variable =    6.567
 ----------------------------
 Observed and fitted values are stored in poi.fit
 LaTeX code for tree is in poi.tex
 R code is stored in poi.r
```

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 20.

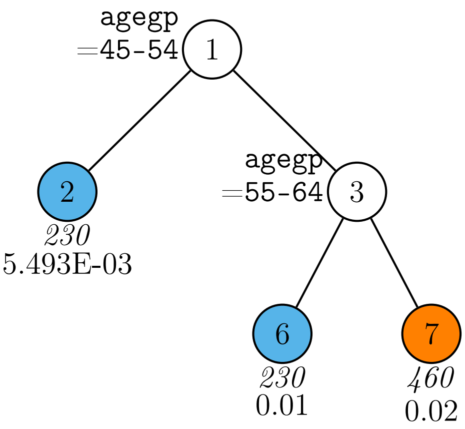

Figure 20: GUIDE v.38.0 0.25-SE multiple linear Poisson regression tree for predicting rate of `deaths`. Tree constructed with 920 observations. Maximum number of split levels is 10 and minimum node sample size is 7. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (*in italics*) and sample rate printed below nodes. Terminal nodes with rates above and below value of 0.01 at root node are colored orange and skyblue respectively. Second best split variable at root node is `sex`.

10 Censored response

Section 4 saw the modeling of right heart catheterization (RHC) in terms of the other variables. The data include a time-to-death variable survtime and a variable death that equals 1 if the subject died (uncensored) and equals 0 otherwise (censored). GUIDE can fit a proportional hazards model to the censored survival time if the event indicator death is specified as "D" and survtime as "T". The description file is rhcdsc2.txt whose contents follow.

```
rhcdata.txt
NA
2
1 X x
2 cat1 c
3 cat2 c
4 ca c
5 sadmdte x
6 dschdte x
7 dthdte x
8 lstctdte x
9 death d
10 cardiohx c
11 chfhx c
12 dementhx c
13 psychhx c
14 chrpulhx c
15 renalhx c
16 liverhx c
17 gibledhx c
18 malighx c
19 immunhx c
20 transhx c
21 amihx c
22 age n
23 sex c
24 edu n
25 surv2md1 n
26 das2d3pc n
27 t3d30 x
28 dth30 x
29 aps1 n
30 scoma1 n
31 meanbp1 n
32 wblc1 n
33 hrt1 n
34 resp1 n
```

```
35 temp1 n
36 pafi1 n
37 alb1 n
38 hema1 n
39 bili1 n
40 crea1 n
41 sod1 n
42 pot1 n
43 paco21 n
44 ph1 n
45 swang1 c
46 wtkilo1 n
47 dnr1 c
48 ninsclas c
49 resp c
50 card c
51 neuro c
52 gastr c
53 renal c
54 meta c
55 hema c
56 seps c
57 trauma c
58 ortho c
59 adld3p n
60 urin1 n
61 race c
62 income c
63 ptid x
64 survtime t
```

10.1 Proportional hazards

GUIDE has two options for modeling censored response data. The first is a piecewise Cox proportional hazards model.

Let the survival time of a subject be U with probability density $f(u)$ and distribution function $F(u)$. The survival probability function is $S(u) = P(U > u) = 1-F(u)$ and the hazard rate (instantaneous rate of death) at time u is $\lambda(u) = f(u)/S(u)$. Let U_i and C_i be survival and censoring times of subject i. Let $Y_i = \min(U_i, C_i)$ be the observed censored survival time and let $\delta_i = I(U_i < C_i)$ denote the event indicator. The proportional hazards model assumes that $\lambda(u, \mathbf{x}) = \lambda_0(u) \exp(\beta'\mathbf{x})$, where $\lambda_0(u)$ is an unknown baseline hazard function. Unlike other regression tree methods for survival data, $\lambda_0(u)$ is the same for all terminal nodes of a GUIDE tree.

10.1.1 Input file generation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: censored.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: censored.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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple linear in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
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: rhcdsc2.txt
Reading data description file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
20 N variables changed to S
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Re-checking data ...
Allocating missing value information ...
```

```
Assigning codes to missing values if any ...
Data checks complete
Smallest uncensored survtime: 2.0000
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime with positive weight: 1943.0000 1351.0000
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       31        0
Survival time variable in column: 64
Event indicator variable in column: 9
Proportion uncensored among nonmissing T and D variables: 0.649
Number of cases used for training: 5735
Number of split variables: 54
Number of cases excluded due to 0 weight or missing D or T: 0
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): censored.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: censored.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: censored.r
Input rank of top variable to split root node ([1:51], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < censored.in
```

10.1.2 Output file

```
Regression tree for censored response
Pruning by cross-validation
Data description file: rhcdsc2.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
23 N variables changed to S
D variable is death
Piecewise constant model
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
```

```
Missing values found among non-categorical variables
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
Smallest uncensored survtime: 2.0000
Largest uncensored and censored survtime with positive weight: 1943.0000 1351.0000

Summary information for training sample of size 5735
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
                                                 #Codes/
                                                 Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      2  cat1       c                                  9
      3  cat2       c                                  6     4535
      4  ca         c                                  3
      9  death      d    0.000        1.000
     10  cardiohx   c                                  2
     11  chfhx      c                                  2
      :
     45  swang1     c                                  2
      :
     64  survtime   t    2.000        1943.
  =================== Constructed variables ===================
     65  lnbasehaz  z   -3.818        2.038

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       31        0
Survival time variable in column: 64
Event indicator variable in column: 9
Proportion uncensored among nonmissing T and D variables: 0.649
Number of cases used for training: 5735
Number of split variables: 54
Number of cases excluded due to 0 weight or missing D or T: 0

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
```

```
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 15
Minimum node sample size: 57
Number of iterations for fitting: 20
Top-ranked variables and chi-squared values at root node
     1  0.7573E+03   surv2md1
     2  0.3288E+03   adld3p
     3  0.2341E+03   cat1
     4  0.2263E+03   aps1
     5  0.2170E+03   ca
     6  0.2119E+03   dnr1
     :
    51  0.1094E-01   chrpulhx
    52  0.8247E-02   cardiohx

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       74   1.284E+00   1.996E-02   1.216E-02   1.282E+00   1.261E-02
   2       73   1.284E+00   1.996E-02   1.228E-02   1.282E+00   1.262E-02
   :
  43       11   1.251E+00   1.800E-02   1.319E-02   1.251E+00   1.993E-02
  44**     10   1.246E+00   1.776E-02   1.259E-02   1.237E+00   1.786E-02
  45++      8   1.254E+00   1.718E-02   1.245E-02   1.241E+00   1.868E-02
  46        7   1.259E+00   1.717E-02   1.177E-02   1.249E+00   2.188E-02
  47        6   1.273E+00   1.723E-02   1.130E-02   1.270E+00   1.882E-02
  48        5   1.289E+00   1.744E-02   1.194E-02   1.284E+00   1.923E-02
  49        3   1.296E+00   1.714E-02   1.295E-02   1.297E+00   2.324E-02
  50        2   1.337E+00   1.699E-02   1.161E-02   1.331E+00   1.397E-02
  51        1   1.459E+00   1.629E-02   6.178E-03   1.454E+00   9.978E-03

0-SE tree based on mean is marked with * and has 10 terminal nodes
0-SE tree based on median is marked with + and has 10 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
```

```
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix   Median      Node    Split          Other
       label   cases      fit  rank  survtime  deviance  variable       variables
          1     5735     5735     1  1.920E+02  1.459E+00  surv2md1
          2     2164     2164     1  2.300E+01  1.499E+00  adld3p
          4     1930     1930     1  1.800E+01  1.530E+00  surv2md1
          8T     709      709     1  1.100E+01  1.429E+00  cat1
          9     1221     1221     1  2.800E+01  1.498E+00  dnr1
         18T    1027     1027     1  3.700E+01  1.434E+00  surv2md1
         19T     194      194     1  8.000E+00  1.431E+00  aps1
          5T     234      234     1  1.950E+02  9.294E-01  ca
          3     3571     3571     1  3.290E+02  1.223E+00  surv2md1
          6     1805     1805     1  2.270E+02  1.347E+00  adld3p
         12     1364     1364     1  1.290E+02  1.457E+00  dnr1
         24T    1214     1214     1  1.710E+02  1.412E+00  das2d3pc
         25T     150      150     1  2.550E+01  1.600E+00  hema1
         13T     441      441     1  3.750E+02  8.602E-01  das2d3pc
          7     1766     1766     1  4.030E+02  1.019E+00  chfhx
         14     1276     1276     1  4.410E+02  1.036E+00  das2d3pc
         28T     815      815     1  3.640E+02  1.065E+00  wtkilo1
         29T     461      461     1  6.720E+02  9.083E-01  surv2md1
         15T     490      490     1  3.730E+02  9.322E-01  surv2md1

Number of terminal nodes of final tree: 10
Total number of nodes of final tree: 19
Second best split variable (based on curvature test) at root node is adld3p

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: surv2md1 <= 0.56447053
  Node 2: adld3p = NA
    Node 4: surv2md1 <= 0.35847378
      Node 8: Median survival time = 11.000000
    Node 4: surv2md1 > 0.35847378 or NA
      Node 9: dnr1 = "No"
        Node 18: Median survival time = 37.000000
      Node 9: dnr1 /= "No"
        Node 19: Median survival time = 8.0000000
  Node 2: adld3p /= NA
    Node 5: Median survival time = 195.00000
Node 1: surv2md1 > 0.56447053 or NA
  Node 3: surv2md1 <= 0.71744752
    Node 6: adld3p = NA
      Node 12: dnr1 = "No"
        Node 24: Median survival time = 171.00000
       Node 12: dnr1 /= "No"
         Node 25: Median survival time = 26.000000
     Node 6: adld3p /= NA
       Node 13: Median survival time = 375.00000
   Node 3: surv2md1 > 0.71744752 or NA
     Node 7: chfhx = "0"
       Node 14: das2d3pc <= 23.857420
         Node 28: Median survival time = 364.00000
       Node 14: das2d3pc > 23.857420 or NA
         Node 29: Median survival time = 672.00000
     Node 7: chfhx /= "0"
       Node 15: Median survival time = 373.00000

 ***************************************************************

 Predictor means below are means of cases with no missing values.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if surv2md1 <= 0.56447053
 surv2md1 mean = 0.59245008
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value
 Constant      0.000
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if adld3p = NA
 adld3p mean = 1.3589744
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if surv2md1 <= 0.35847378
 surv2md1 mean = 0.38175857
 ----------------------------
Node 8: Terminal node
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value
 Constant      1.015
 ----------------------------
Node 9: Intermediate node
 A case goes into Node 18 if dnr1 = "No"
 dnr1 mode = "No"
 ----------------------------
Node 18: Terminal node
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.3649
 ----------------------------
Node 19: Terminal node
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value
 Constant      1.172
 ----------------------------
Node 5: Terminal node
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.1060E-02
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if surv2md1 <= 0.71744752
 surv2md1 mean = 0.71787079
 ----------------------------
Node 6: Intermediate node
 A case goes into Node 12 if adld3p = NA
 adld3p mean = 1.1609977
 ----------------------------
Node 12: Intermediate node
 A case goes into Node 24 if dnr1 = "No"
 dnr1 mode = "No"
 ----------------------------
Node 24: Terminal node
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value
 Constant    -0.1619E-01
 ----------------------------
Node 25: Terminal node
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.5691
 ----------------------------
Node 13: Terminal node
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value
 Constant    -0.5149
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if chfhx = "0"
 chfhx mode = "0"
 ----------------------------
Node 14: Intermediate node
 A case goes into Node 28 if das2d3pc <= 23.857420
 das2d3pc mean = 21.937035
 ----------------------------
Node 28: Terminal node
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value
 Constant    -0.5792
 ----------------------------
Node 29: Terminal node
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value
 Constant     -1.216
 ----------------------------
Node 15: Terminal node
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value
 Constant    -0.4135
 ----------------------------
 Observed and fitted values are stored in censored.fit
 LaTeX code for tree is in censored.tex
 R code is stored in censored.r
```

The top few lines of the file `censored.fit` are:

```
 train node obs_survtime  logbasecumhaz  survivalprob  mediansurvtime
     y  13  2.40000E+02+  -2.611849E-01  6.311581E-01   3.750000E+02
     y  15  4.50000E+01   -8.043843E-01  7.439029E-01   3.730000E+02
     y   8  3.17000E+02+  -5.002437E-02  7.254454E-02   1.100000E+01
     y  18  3.70000E+01   -8.890035E-01  5.531804E-01   3.700000E+01
     y  19  2.00000E+00   -4.010552E+00  9.431441E-01   8.000000E+00
```

The columns are:

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

node: terminal node label of observation.

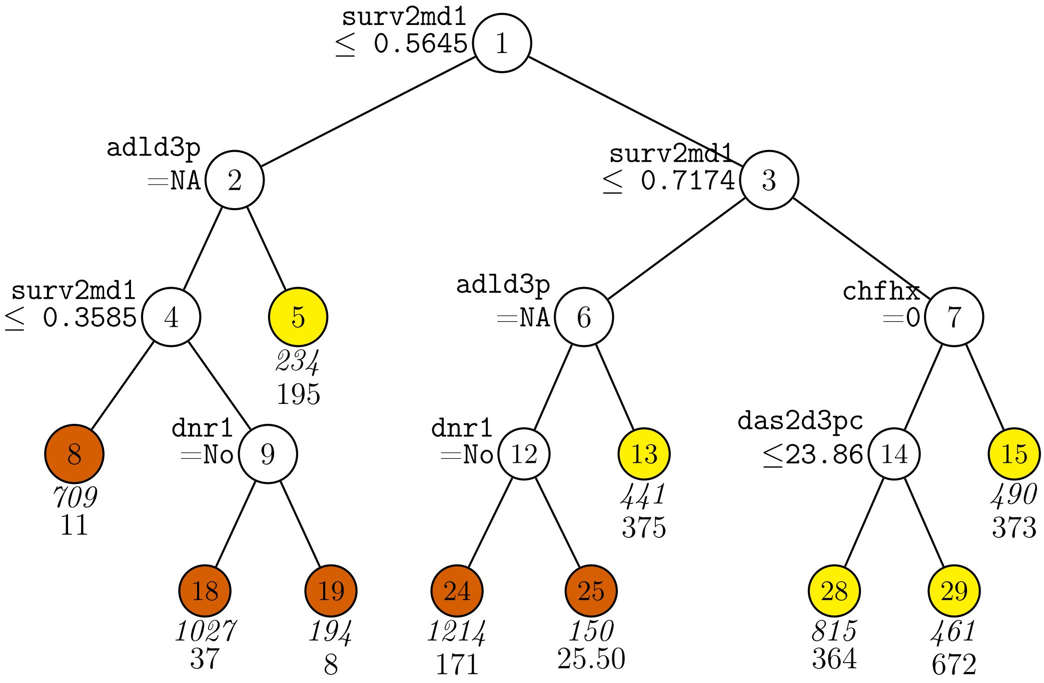

Figure 21: GUIDE v.38.0 0.25-SE piecewise constant proportional hazards regression tree for `survtime`. Tree constructed with 5735 observations. Maximum number of split levels is 15 and minimum node sample size is 57. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*) and median survival time printed below nodes. Terminal nodes with median survival times above and below 192 (median at root node) are colored yellow and vermillion respectively. Second best split variable at root node is `adld3p`.

obs_survtime: observed survival time t. Censored values have trailing plus (+) signs.

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.

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

$$\begin{aligned} \exp\{-\Lambda_0(t) \exp(\boldsymbol{\beta}'\mathbf{x})\} &= \exp\{-\exp(\beta_0 + \texttt{logbasecumhaz})\} \\ &= \exp(-\exp(-0.514911594896 - 0.2611849)) \\ &= 0.6311581 \end{aligned}$$

where $t = 240$ and $\beta_0 = -0.514911594896$ is the constant term in the node (`censored.r` gives β_0 to higher precision than `censored.out`).

mediansurvtime: median survival time among observations in node estimated from Kaplan-Meier survival function. A trailing plus (+) sign indicates estimate is censored.

Figure 22 plots the estimated survival curves in the terminal nodes of the tree. The plot is produced by the following R code.

```
library(survival)
z0 <- read.table("rhcdata.txt",header=TRUE)
z1 <- read.table("censored.fit",header=TRUE)
nodenum <- unique(sort(z1$node))
leg.txt <- paste("Node",nodenum)
leg.col <- c("green","magenta","blue","cyan","red")
leg.lty <- rep(c(1,2),c(5,5))
fit <- survfit(Surv(z0$survtime,z0$death) ~ z1$node, conf.type="none")
plot(fit,mark.time=FALSE,xlab="Survival time",ylab="Survival probability",
     col=leg.col,lwd=2,lty=leg.lty)
title("Kaplan-Meier survival curves")
legend("topright",legend=leg.txt,lty=leg.lty,col=leg.col,lwd=2,ncol=2)
```

10.2 Restricted mean event time

The mean survival time is not estimable if there is censoring. But given a prespecified time point τ, the restricted mean survival time $\mu(X) = E(Y|X)$ is estimable, where $Y = \min(U, C, \tau)$ and X is a covariate vector (Andersen et al., 2004; Chen and Tsiatis, 2001; Tian et al., 2014). GUIDE has an option to fit a *restricted event time model* to each node of the tree such that $\mu(X)$ is linear in the covariates.

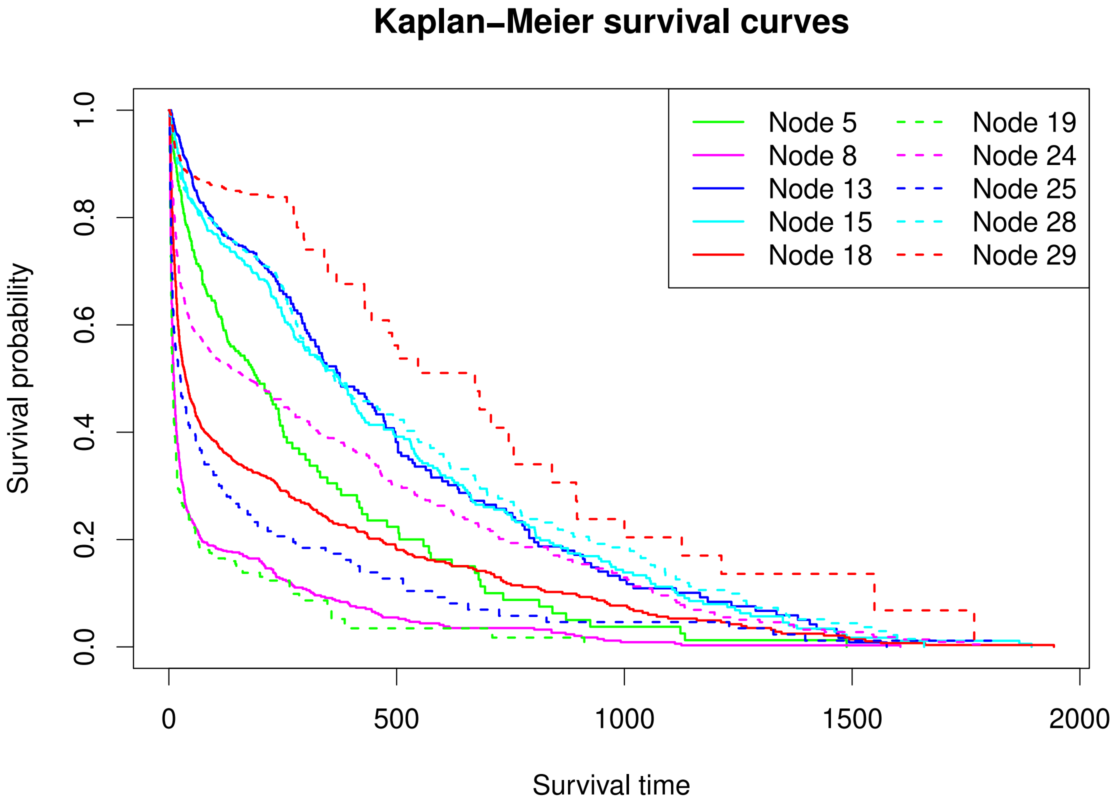

Figure 22: Kaplan-Meier survival curves for data in terminal nodes of Figure 21

10.2.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: rest.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: rest.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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended if there are no missing values)
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple polynomial in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
0: stepwise linear, 1: multiple linear, 2: simple polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <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: rhcdsc2.txt
Reading data description file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
20 N variables changed to S
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
```

```
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime with positive weight: 1943.0000 1351.0000
Smallest observed uncensored time is 2.0000
Largest observed censored or uncensored time is 1943.0000
Input restriction on event time ([2.00:1943.00], <cr>=972.00):

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       31        0
No weight variable in data file
Number of cases used for training: 3732
Number of split variables: 54
Number of cases excluded due to 0 weight or missing D: 2003
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): rest.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: rest.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: rest.r
Input rank of top variable to split root node ([1:51], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < rest.in
```

10.2.2 Contents of rest.out

```
Restricted mean event time regression tree
Pruning by cross-validation
Data description file: rhcdsc2.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
23 N variables changed to S
D variable is death
Piecewise constant model
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
```

```
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Smallest uncensored survtime: 2.0000
Largest uncensored and censored survtime with positive weight: 1943.0000 1351.0000
Interval for restricted mean event time is from 0 to 972.

Summary information for training sample of size 3732 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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
      2  cat1       c                                  9
      3  cat2       c                                  6     2807
      4  ca         c                                  3
      9  death      d    0.000        1.000
      :
     45  swang1     c                                  2
      :
     62  income     c                                  4
     64  survtime   t    2.000        1943.

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       31        0
No weight variable in data file
Number of cases used for training: 3732
Number of split variables: 54
Number of cases excluded due to 0 weight or missing D: 2003

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 13
Minimum node sample size: 37
Top-ranked variables and chi-squared values at root node
     1  0.1868E+03   adld3p
     2  0.1629E+03   surv2md1
     3  0.1122E+03   cat1
     4  0.6234E+02   aps1
     :
    51  0.1196E+00   amihx
    52  0.6209E-01   income

Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       75   1.121E+05   3.376E+03   2.477E+03   1.120E+05   2.101E+03
   2       74   1.121E+05   3.376E+03   2.477E+03   1.119E+05   2.107E+03
   :
  43+       8   1.086E+05   3.212E+03   2.008E+03   1.082E+05   3.190E+03
  44        7   1.086E+05   3.184E+03   2.177E+03   1.086E+05   3.279E+03
  45**      6   1.067E+05   3.063E+03   1.467E+03   1.084E+05   2.196E+03
  46        4   1.091E+05   3.044E+03   1.503E+03   1.090E+05   2.580E+03
  47        3   1.097E+05   3.045E+03   1.425E+03   1.090E+05   1.927E+03
  48        2   1.102E+05   3.062E+03   1.527E+03   1.102E+05   2.279E+03
  49        1   1.225E+05   3.100E+03   2.805E+02   1.225E+05   4.687E+02

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 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
* 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 weighted mean of death 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    Total    Cases Matrix    Node      Node      Split          Interacting
       label   cases      fit  rank    D-mean     MSE       variable       variable
          1     3732     3732     1  3.144E+02  1.800E+05   adld3p
          2      664      664     1  4.685E+02  2.273E+05   surv2md1
          4T      168      168     1  3.244E+02  1.404E+05   immunhx
          5      496      496     1  5.040E+02  2.427E+05   urin1
         10T      314      314     1  5.756E+02  2.829E+05   sod1
         11T      182      182     1  3.515E+02  1.074E+05   race
```

```
          3     3068     3068     1  2.647E+02  1.556E+05   surv2md1
          6T    1262     1262     1  1.607E+02  8.878E+04   dnr1
          7     1806     1806     1  3.225E+02  1.880E+05   urin1
         14T    1000     1000     1  4.001E+02  2.482E+05   surv2md1
         15T     806      806     1  2.057E+02  8.243E+04   swang1 :immunhx

 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 surv2md1

 Regression tree:

 Node 1: adld3p <= 5.5000000
   Node 2: surv2md1 <= 0.58646870
     Node 4: survtime-mean = 324.40508
   Node 2: surv2md1 > 0.58646870 or NA
     Node 5: urin1 = NA
       Node 10: survtime-mean = 575.62515
     Node 5: urin1 /= NA
       Node 11: survtime-mean = 351.45397
 Node 1: adld3p > 5.5000000 or NA
   Node 3: surv2md1 <= 0.49098337
     Node 6: survtime-mean = 160.70095
   Node 3: surv2md1 > 0.49098337 or NA
     Node 7: urin1 = NA
       Node 14: survtime-mean = 400.06348
     Node 7: urin1 /= NA
       Node 15: survtime-mean = 205.70770

 ***************************************************************

 Predictor means below are means of cases with no missing values.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if adld3p <= 5.5000000
 adld3p mean = 1.2733830
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant      314.4        45.27        0.000
 survtime mean = 314.380
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if surv2md1 <= 0.58646870
 surv2md1 mean = 0.68493485
 ----------------------------
Node 4: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      324.4        11.22        0.000
 survtime mean = 324.405
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if urin1 = NA
 urin1 mean = 2420.9321
 ----------------------------
Node 10: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      575.6        19.18        0.000
 survtime mean = 575.625
 ----------------------------
Node 11: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      351.5        14.47        0.000
 survtime mean = 351.454
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if surv2md1 <= 0.49098337
 surv2md1 mean = 0.54259828
 ----------------------------
Node 6: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      160.7        19.16        0.000
 survtime mean = 160.701
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if urin1 = NA
 urin1 mean = 1998.7301
 ----------------------------
```

```
Node 14: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      400.1        25.39        0.000
 survtime mean = 400.063
 ----------------------------
Node 15: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      205.7        20.34        0.000
 survtime mean = 205.708
 ----------------------------
 Observed and fitted values are stored in rest.fit
 LaTeX code for tree is in rest.tex
 R code is stored in rest.r
```

Figure 23 shows the restricted mean event time tree.

11 Randomized trials

Causal effects of treatments are best studied in a randomized trial where the treatments are assigned randomly to subjects. The goal is to show that one treatment is more efficacious than another across all subjects. If this determination is not achieved, a secondary goal may be to search for subgroups of subjects with differential treatment effects.

There are two types of covariates for identification of subgroups with differential treatment effects. 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 one that provides information on the likely benefit from the treatment. Predictive variables can be used to identify subgroups of patients who are most likely to benefit from a given therapy. In general, 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 options, 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).

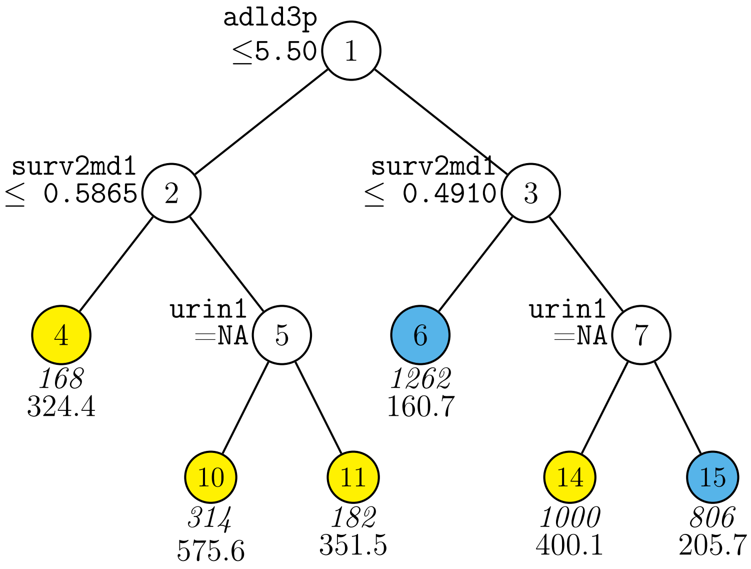

Figure 23: GUIDE v.38.0 0.25-SE piecewise constant regression tree for mean `survtime` restricted to less than 972.000. Tree constructed with 3732 observations (excluding observations with non-positive weight or missing values in `D`, `T`, `R` or `Z` variables). Maximum number of split levels is 13 and minimum node sample size is 37. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (*in italics*) and restricted mean of `survtime` printed below nodes. Terminal nodes with means above and below value of 314.4 at root node are colored yellow and skyblue respectively. Second best split variable at root node is `surv2md1`.

11.1 Three treatment arms

We first demonstrate this on a data set from a three-armed randomized controlled experiment to find out whether two interventions (DVD or Phone) are more efficacious than a control at promoting mammography screening. The relevant data and description files are `cape.dat` and `cape.dsc`. Note that the three treatment levels (contained in the treatment (`R`) variable `group`) are assumed to be categorical (i.e., nominal valued). See Loh et al. (2016) for more information on the data.

Because the response variable (`resp6`) is 0-1 (0=no, 1=yes), we use least-squares regression with `resp6` designated as the dependent variable `D` or `d` in the description file. The treatment variable (`group`) is designated as `R` or `r` (for “Rx”).

11.1.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: gi.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: gi.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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <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 (recommended if there are no missing values)
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple polynomial in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
0: stepwise linear, 1: multiple linear, 2: simple polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <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: cape.dsc
Reading data description file ...
Training sample file: cape.dat
Missing value code: NA
```

```
Records in data file start on line 1
R variable present
21 N variables changed to S
Warning: model changed to linear in treatment
D variable is resp6
Reading data file ...
Number of records in data file: 1681
Length of longest entry in data file: 25
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 18 categorical variables
Finished assigning codes to 10 categorical variables
Treatment (R) variable is group with values "Control", "DVD", and "Phone"
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
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: 2
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):

Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
Proportion of training sample for each level of group
"Control"    0.3278
    "DVD"    0.3309
  "Phone"    0.3413
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      1681         43         84        1        0        0       21
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       17        0        1
No weight variable in data file
Number of cases used for training: 1638
Number of split variables: 38
Number of dummy variables created: 2
Number of cases excluded due to 0 weight or missing D or R: 43
```

```
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): gi.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: gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: gi.r
Input rank of top variable to split root node ([1:41], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < gi.in
```

11.1.2 Contents of gi.out

```
Least squares regression tree
Pruning by cross-validation
Data description file: cape.dsc
Training sample file: cape.dat
Missing value code: NA
Records in data file start on line 1
R variable present
21 N variables changed to S
Warning: model changed to linear in treatment
D variable is resp6
Piecewise linear model
Number of records in data file: 1681
Length of longest entry in data file: 25
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Treatment (R) variable is group with values "Control", "DVD", and "Phone"
Number of dummy variables created: 2
Proportion of training sample for each level of group
"Control"    0.3278
    "DVD"    0.3309
  "Phone"    0.3413

Summary information for training sample of size 1638 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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  resp6         d    0.000        1.000
      3  group         r                                     3
      4  age           s    41.00        75.00                       1
      5  educyrs       s    2.000        20.00
      6  collegeormore  c                                     2
      :
     37  susc          s    5.000        25.00
     38  fear          s    8.000        40.00
     39  fatal         s    11.00        42.00
     40  know          s    1.000        7.000
     41  stage         c                                     4
  =================== Constructed variables ===================
     42  group.DVD     f    0.000        1.000
     43  group.Phone   f    0.000        1.000

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      1681         43         84        1        0        0       21
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       17        0        1
 No weight variable in data file
 Number of cases used for training: 1638
 Number of split variables: 38
 Number of dummy variables created: 2
 Number of cases excluded due to 0 weight or missing D or R: 43

 Missing values imputed with node means for fitting regression models in nodes
 Predictive priority (Gi)
 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: 0.2500

 No nodewise interaction tests
 Split values for N and S variables based on exhaustive search
 Maximum number of split levels: 11
 Minimum node sample size: 8
 Minimum fraction of cases per treatment at each node: 0.066
 Top-ranked variables and chi-squared values at root node
      1  0.6775E+01   sf12gh
      2  0.5072E+01   know
      3  0.3940E+01   incle75k
      :
     30  0.1110E-03   sf12pf
     31  0.1774E-07   sf12mh

 Size and CV MSE and SE of subtrees:
```

```
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1      125   3.439E-01   9.506E-03   1.064E-02   3.585E-01   1.561E-02
   2      124   3.439E-01   9.506E-03   1.064E-02   3.585E-01   1.561E-02
   :
  77       12   2.491E-01   4.721E-03   6.754E-03   2.462E-01   6.768E-03
  78**      5   2.390E-01   3.240E-03   2.264E-03   2.410E-01   3.959E-03
  79++      1   2.414E-01   2.372E-03   5.044E-04   2.410E-01   6.719E-04

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 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.

D-mean is mean of resp6 in the node
Cases fit give the number of cases used to fit node
MSE and R^2 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
          1     1638     1638     3  4.035E-01  2.410E-01  0.0006 sf12gh
          2      903      903     3  3.732E-01  2.336E-01  0.0046 know
          4      703      703     3  3.898E-01  2.384E-01  0.0018 educyrs
          8      543      543     3  3.720E-01  2.324E-01  0.0105 yearmam
         16T     427      427     3  2.998E-01  2.091E-01  0.0107 educyrs
         17T     116      116     3  6.379E-01  2.248E-01  0.0518 sf12rp
          9T     160      160     3  4.500E-01  2.387E-01  0.0535 know
          5T     200      200     3  3.150E-01  2.039E-01  0.0693 fear
          3T     735      735     3  4.408E-01  2.455E-01  0.0081 sf12sf

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 know

Regression tree:

Node 1: sf12gh <= 72.500000
  Node 2: know <= 6.5000000
    Node 4: educyrs <= 15.500000
```

```
       Node 8: yearmam <= 3.5000000
         Node 16: resp6-mean = 0.29976581
       Node 8: yearmam > 3.5000000 or NA
         Node 17: resp6-mean = 0.63793103
     Node 4: educyrs > 15.500000 or NA
       Node 9: resp6-mean = 0.45000000
   Node 2: know > 6.5000000 or NA
     Node 5: resp6-mean = 0.31500000
 Node 1: sf12gh > 72.500000 or NA
   Node 3: resp6-mean = 0.44081633

 ***************************************************************

 Predictor means below are means of cases with no missing values.
 Regression coefficients are computed from the complete cases.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if sf12gh <= 72.500000
 sf12gh mean = 65.921856
 Coefficients of least squares regression function:
 Regressor      Coefficient  t-stat      p-value      Minimum         Mean      Maximum
 Constant       0.3985        18.81         0.000
 group.DVD     -0.7366E-02  -0.2465       0.8054        0.000       0.3309        1.000
 group.Phone    0.2188E-01   0.7378       0.4608        0.000       0.3413        1.000
 resp6 mean = 0.403541
 No truncation of predicted values
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if know <= 6.5000000
 know mean = 5.6087154
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if educyrs <= 15.500000
 educyrs mean = 13.800853
 ----------------------------
```

```
Node 8: Intermediate node
 A case goes into Node 16 if yearmam <= 3.5000000
 yearmam mean = 2.0055249
 ----------------------------
Node 16: Terminal node
 Coefficients of least squares regression functions:
 Regressor      Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       0.3333        8.279       0.2776E-14
 group.DVD     -0.9843E-01   -1.790       0.7419E-01    0.000       0.3489        1.000
 group.Phone    0.2237E-02   0.4068E-01   0.9676        0.000       0.3489        1.000
 resp6 mean = 0.299766
 No truncation of predicted values
 ----------------------------
Node 17: Terminal node
 Coefficients of least squares regression functions:
 Regressor      Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       0.5000        6.149       0.1204E-07
 group.DVD      0.1154        1.037       0.3019        0.000       0.3362        1.000
 group.Phone    0.2674        2.458       0.1550E-01    0.000       0.3707        1.000
 resp6 mean = 0.637931
 No truncation of predicted values
 ----------------------------
Node 9: Terminal node
 Coefficients of least squares regression functions:
 Regressor      Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       0.3788        6.298       0.2840E-08
 group.DVD      0.2366        2.611       0.9889E-02    0.000       0.3250        1.000
 group.Phone   -0.2165E-01  -0.2244       0.8227        0.000       0.2625        1.000
 resp6 mean = 0.450000
 No truncation of predicted values
 ----------------------------
Node 5: Terminal node
 Coefficients of least squares regression functions:
 Regressor      Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       0.1831        3.417       0.7695E-03
 group.DVD      0.2883        3.791       0.1993E-03    0.000       0.3500        1.000
 group.Phone    0.1050        1.321       0.1882        0.000       0.2950        1.000
 resp6 mean = 0.315000
 No truncation of predicted values
 ----------------------------
Node 3: Terminal node
 Coefficients of least squares regression functions:
 Regressor      Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       0.4895        15.21        0.000
 group.DVD     -0.1101       -2.407       0.1634E-01    0.000       0.3156        1.000
 group.Phone   -0.3832E-01  -0.8659       0.3868        0.000       0.3619        1.000
```

```
resp6 mean = 0.440816
No truncation of predicted values
----------------------------
Number of times Li-Martin approximation used = 157
Proportion of variance (R-squared) explained by tree model: 0.0579

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

The tree has 5 terminal nodes (subgroups) and the results for each terminal node give the treatment effects of `DVD` and `Phone` versus `Control`, which is the first treatment level in alphabetical order. Figure 24 shows the tree diagram.

11.2 Censored response: proportional hazards

We now consider a randomized controlled breast cancer trial where the response variable is a censored survival time (Schmoor et al., 1996). The data are in the file `cancerdata.txt`; they are included in the `TH.data` R package (Hothorn, 2017) as well. 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 the event indicator `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 are automatically changed to `s` when the program executes). See Loh et al. (2019a, 2016, 2015, 2019c) and Loh and Zhou (2020) for further analysis of the data.

```
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
```

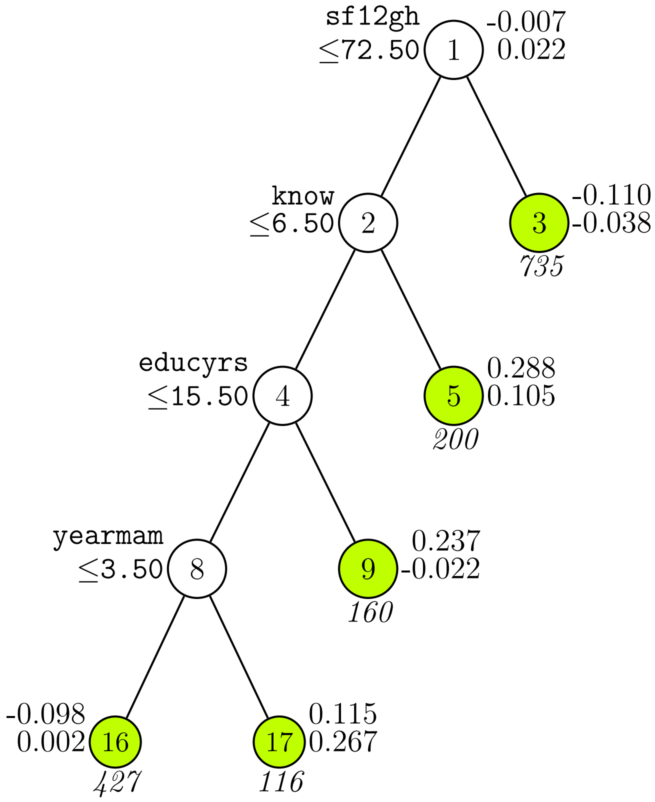

Figure 24: GUIDE v.38.0 0.25-SE least-squares regression tree using Gi option for dependent variable `resp6` without linear prognostic effects. Tree constructed with 1638 observations (excluding observations with non-positive weight or missing values in `D`, `T`, `R` or `Z` variables). Maximum number of split levels is 11, minimum node sample size is 8 and minimum treatment fraction is 0.066. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size *(in italics)* printed below nodes. Treatment `group` effects (relative to level `Control`) of `DVD`, `Phone`, respectively, beside nodes. Second best split variable at root node is `know`.

11.2.1 Without linear prognostic control

The simplest model only uses the covariates to split the intermediate nodes; terminal nodes are fitted with treatment means.

Input file generation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: ph-gi.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: ph-gi.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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple linear in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
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: 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
6 N variables changed to S
Warning: model changed to linear in treatment
D 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 ...
Finished checking
Assigning integer codes to values of 2 categorical variables
Treatment (R) variable is horTh with values "no" and "yes"
```

```
Re-checking data ...
Assigning codes to missing values if any ...
Data checks complete
Smallest uncensored time: 72.0000
Number of cases dropped due to missing D or T or censored T < smallest uncensored T: 14
Number of complete cases excluding censored T < smallest uncensored T: 672
Number of cases used to compute baseline hazard: 672
Number of 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 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):

Creating dummy variables ...
Rereading data ...
Largest uncensored and censored time by horTh
 "no"      2456.0000    2563.0000
"yes"      2372.0000    2659.0000
Proportion of training sample for each level of horTh
 "no"    0.6399
"yes"    0.3601
     Total  #cases w/   #missing
    #cases    miss. D  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
Number of cases used for training: 672
Number of split variables: 7
Number of dummy variables created: 1
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): ph-gi.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: ph-gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: ph-gi.r
Input rank of top variable to split root node ([1:9], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < ph-gi.in
```

Results The contents of ph-gi.out follow.

```
Regression tree for censored response
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
6 N variables changed to S
Warning: model changed to linear in treatment
D 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"
Number of cases dropped due to missing D or T or censored T < smallest uncensored T: 14
Number of complete cases excluding censored T < smallest uncensored T: 672
Number of cases used to compute baseline hazard: 672
Number of cases with D=1 and T >= smallest uncensored: 299
Number of dummy variables created: 1
Smallest uncensored time: 72.0000
Largest uncensored and censored time by horTh
horTh     Uncensored     Censored
 "no"      2456.0000    2563.0000
"yes"      2372.0000    2659.0000
Proportion of training sample for each level of horTh
 "no"    0.6399
"yes"    0.3601

Summary information for training sample of size 672 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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
                                                 #Codes/
                                                 Levels/
 Column  Name            Minimum      Maximum    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  lnbasehaz  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        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
 Number of cases used for training: 672
 Number of split variables: 7
 Number of dummy variables created: 1

 Missing values imputed with node means for fitting regression models in nodes
 Predictive priority (Gi)
 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: 0.2500

 No nodewise interaction tests
 Fraction of cases used for splitting each node: 1.0000
 Maximum number of split levels: 10
 Minimum node sample size: 6
 Minimum fraction of cases per treatment at each node: 0.072
 Number of iterations for fitting: 20
 Top-ranked variables and chi-squared values at root node
      1  0.2101E+01   progrec
      2  0.1669E+01   estrec
      3  0.1108E+01   tsize
      4  0.3557E+00   pnodes
      5  0.2413E+00   tgrade
      6  0.2057E-01   menostat
      7  0.1879E-02   age

 Size and CV Loss and SE of subtrees:
  Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
    1       48   1.739E+00   8.406E-02   6.834E-02   1.706E+00   7.329E-02
    2       47   1.737E+00   8.408E-02   6.866E-02   1.697E+00   7.379E-02
    :
   30**      2   1.398E+00   5.064E-02   1.949E-02   1.400E+00   2.803E-02
   31       1   1.435E+00   5.100E-02   1.066E-02   1.446E+00   1.482E-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.

 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   Median      Node    Split
        label   cases      fit  rank   survtime   deviance  variable
           1      672      672     1  1.807E+03  1.431E+00  progrec
           2T     274      274     1  1.140E+03  1.601E+00  estrec
           3T     398      398     1  2.286E+03  1.188E+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: Median survival time = 1140.0000
 Node 1: progrec > 21.500000 or NA
   Node 3: Median survival time = 2286.0000

 ***************************************************************


 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.
```

```
Node 1: Intermediate node
 A case goes into Node 2 if progrec <= 21.500000
 progrec mean = 110.91518
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.000
 horTh.yes   -0.3654       -2.933       0.3471E-02    0.000        0.3601        1.000
 ----------------------------
Node 2: Terminal node
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.3729
 horTh.yes   -0.1140      -0.6871       0.4926        0.000        0.3613        1.000
 ----------------------------
Node 3: Terminal node
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant    -0.2596
 horTh.yes   -0.6453       -3.375       0.8098E-03    0.000        0.3593        1.000
 ----------------------------
 Observed and fitted values are stored in ph-gi.fit
 LaTeX code for tree is in ph-gi.tex
 R code is stored in ph-gi.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 `ph-gi.out` show that the fitted proportional hazards model is

$$\begin{aligned}\lambda(u, \mathbf{x}) &= \lambda_0(u)[\exp\{\hat\beta_1 + \hat\gamma_1 I(\texttt{horTh} = \texttt{yes})\} I(\texttt{progrec} \leq 21.5) \\ &\quad + \exp\{\hat\beta_2 + \hat\gamma_2 I(\texttt{horTh} = \texttt{yes})\} I(\texttt{progrec} > 21.5)]\end{aligned}$$

with $\hat\beta_1 = 0.37292$, $\hat\gamma_1 = -0.11404$, $\hat\beta_2 = -0.25964$, and $\hat\gamma_2 = -0.64531$.

Figure 25 shows the tree diagram. The numbers beside each terminal node are relative hazards of `horTh` = `yes` verus `no`, namely, $\exp(\hat\gamma_1) = \exp(-0.11404) = 0.8922223$ for node 2 and $\exp(\hat\gamma_2) = \exp(-0.64531) = 0.5244999$ for node 3. Figure 26 shows Kaplan-Meier survival functions of the data in the terminal nodes. The plots are produced by the following R code.

```
library(survival)
z <- read.table("cancerdata.txt",header=TRUE)
leg.txt <- c("horTh = yes","horTh = no")
leg.col <- c("red","blue")
leg.lty <- 1:2
```

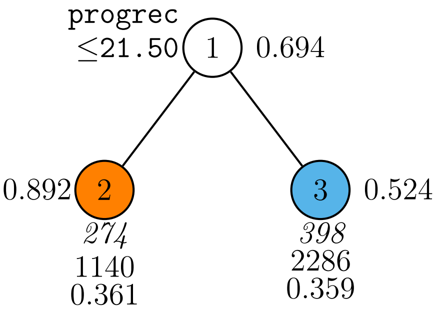

Figure 25: GUIDE v.38.0 0.25-SE proportional hazards regression tree using Gi option for `time` and event indicator `death` without linear prognostic effects. Tree constructed with 672 observations (excluding observations with non-positive weight or missing values in `D`, `T`, `R` or `Z` variables). Maximum number of split levels is 10, minimum node sample size is 6 and minimum treatment fraction is 0.072. At each split, an observation goes to the left branch if and only if the condition is satisfied. Treatment `horTh` hazard ratio of level `yes` to `no` beside nodes. Sample size (in *italics*), median survival time, and proportion of `horTh` = `yes` printed below nodes. Terminal nodes with treatment hazard ratio above and below 0.694 (ratio at root node) are colored orange and skyblue respectively. Second best split variable at root node is `estrec`.

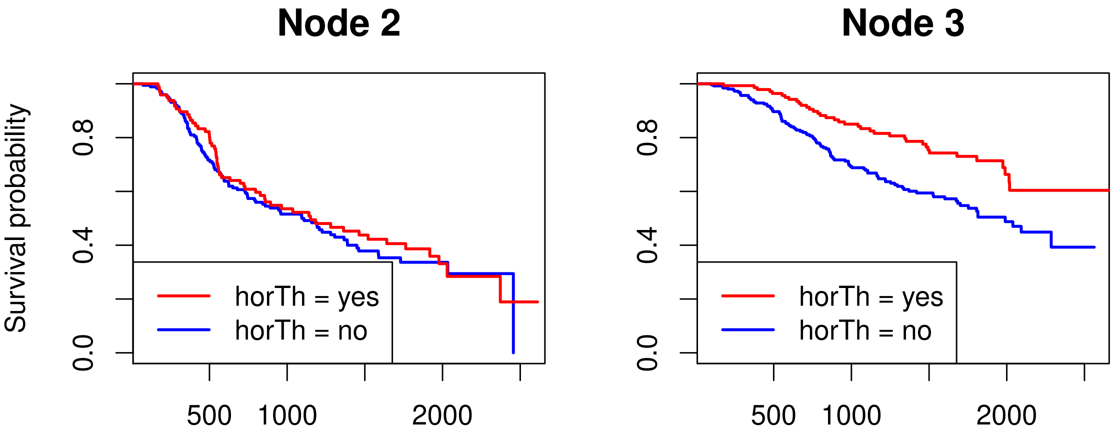

Figure 26: Estimated survival probability functions for breast cancer data

```
xr <- range(z$time)
zg <- read.table("ph-gi.fit",header=TRUE)
nodes <- zg$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 ph-gi.fit gives the terminal node number, observed survival time, log baseline cumulative hazard, 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 trailing plus (+) sign in values of the observed and median survival times indicates censoring.

```
train  node   obs_survtime  logbasecumhaz   survivalprob  mediansurvtime     horTh.yes
  y       3   1.81400E+03   -3.356226E-01   5.761313E-01    2.286000E+03   -6.453111E-01
  y       3   2.01800E+03   -2.103084E-01   7.204845E-01    2.286000E+03   -6.453111E-01
  y       3   7.12000E+02   -1.284520E+00   8.940654E-01    2.286000E+03   -6.453111E-01
  y       3   1.80700E+03   -3.581910E-01   7.536968E-01    2.286000E+03   -6.453111E-01
  y       3   7.72000E+02   -1.162320E+00   7.856518E-01    2.286000E+03   -6.453111E-01
  y       2   4.48000E+02   -2.083218E+00   8.345918E-01    1.140000E+03   -1.140416E-01
  y       3   2.17200E+03+  -1.218663E-01   6.989709E-01    2.286000E+03   -6.453111E-01
```

11.2.2 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., 2019c). 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-gi.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-gi.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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple linear in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
1: multiple linear, 2: simple linear, 3: constant ([1:3], <cr>=3): 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: 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
D 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 ...
Finished checking
Assigning integer codes to values of 2 categorical variables
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
```

```
Assigning codes to missing values if any ...
Data checks complete
Smallest uncensored time: 72.0000
Number of cases dropped due to missing D or T or censored T < smallest uncensored T: 14
Number of complete cases excluding censored T < smallest uncensored T: 672
Number of cases used to compute baseline hazard: 672
Number of 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 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):

Creating dummy variables ...
Rereading data ...
Largest uncensored and censored time by horTh
 "no"      2456.0000    2563.0000
"yes"      2372.0000    2659.0000
Proportion of training sample for each level of horTh
 "no"    0.6399
"yes"    0.3601
     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: .445
Number of cases used for training: 672
Number of split variables: 7
Number of dummy variables created: 1
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): lin-gi.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-gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: lin-gi.r
Input rank of top variable to split root node ([1:9], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < lin-gi.in
```

Contents of `lin-gi.out` The results in the following output file `lin-gi.out` show that the default 0.25-SE mean-based ($*$) tree has no splits. When this happens, GUIDE automatically selects the next smallest nontrivial tree from the 0-SE mean-based ($**$), 0.25-SE median-based ($++$), or 0-SE median-based ($+$) tree. In this example, the $++$ tree is selected; see Figure 27.

```
Regression tree for censored response
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
D 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"
Number of cases dropped due to missing D or T or censored T < smallest uncensored T: 14
Number of complete cases excluding censored T < smallest uncensored T: 672
Number of cases used to compute baseline hazard: 672
Number of cases with D=1 and T >= smallest uncensored: 299
Number of dummy variables created: 1
Smallest uncensored time: 72.0000
Largest uncensored and censored time by horTh
horTh     Uncensored     Censored
 "no"      2456.0000    2563.0000
"yes"      2372.0000    2659.0000
Proportion of training sample for each level of horTh
 "no"    0.6399
"yes"    0.3601

Summary information for training sample of size 672 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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
                                                 #Codes/
                                                 Levels/
 Column  Name            Minimum      Maximum    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  lnbasehaz  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
Number of cases used for training: 672
Number of split variables: 7
Number of dummy variables created: 1

Missing values imputed with node means for fitting regression models in nodes
Predictive priority (Gi)
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 10
Minimum node sample size: 7
Minimum fraction of cases per treatment at each node: 0.072
Number of iterations for fitting: 20
Top-ranked variables and chi-squared values at root node
     1  0.3130E+01   estrec
     2  0.1672E+01   progrec
     3  0.1137E+01   tsize
     4  0.3983E+00   pnodes
     5  0.1718E+00   tgrade
     6  0.9820E-01   menostat
     7  0.2054E-04   age
```

```
Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       43   1.247E+07   1.219E+07   1.214E+07   7.263E+00   3.919E+06
   2       42   1.247E+07   1.219E+07   1.214E+07   7.266E+00   3.919E+06
   :
  20        6   2.741E+05   2.739E+05   2.591E+05   1.542E+00   2.450E-01
  21++      2   1.370E+00   7.295E-02   5.276E-02   1.320E+00   3.197E-02
  22**      1   1.355E+00   5.363E-02   2.719E-02   1.330E+00   2.698E-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 2 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 same as -- tree
+ tree same as ++ tree
* tree same as ** tree
* tree same as -- tree

WARNING: tree based on mean CV estimate of error has no splits
Following tree is based on median CV with bootstrap 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
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix   Median      Node    Split
       label   cases      fit  rank   survtime   deviance  variable
          1      672      672     3  1.807E+03  1.343E+00  estrec
          2T     137      137     3  1.093E+03  1.604E+00  estrec
          3T     535      535     3  1.990E+03  1.235E+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 progrec

Regression tree:

Node 1: estrec <= 4.5000000
  Node 2: Median survival time = 1093.0000
Node 1: estrec > 4.5000000 or NA
  Node 3: Median survival time = 1990.0000

***************************************************************
```

```
 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if estrec <= 4.5000000
 estrec mean = 97.474702
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.000
 pnodes       0.5630E-01    8.575        0.000        1.000        4.987        51.00
 horTh.yes   -0.3465       -2.778       0.5627E-02    0.000       0.3601        1.000
 ----------------------------
Node 2: Terminal node
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.1061
 pnodes       0.1058        4.542       0.1229E-04    1.000        5.328        24.00
 horTh.yes    0.9302E-02   0.3845E-01   0.9694        0.000       0.3358        1.000
 ----------------------------
Node 3: Terminal node
 Coefficients of log-relative hazard function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant    -0.9493E-01
 pnodes       0.5438E-01    7.430       0.4339E-12    1.000        4.899        51.00
 horTh.yes   -0.4141       -2.845       0.4617E-02    0.000       0.3664        1.000
 ----------------------------
 Observed and fitted values are stored in lin-gi.fit
 LaTeX code for tree is in lin-gi.tex
 R code is stored in lin-gi.r
```

11.3 Censored response: restricted mean

11.3.1 Without linear prognostic control

The piecewise-constant tree has no splits when the restricted mean option is chosen.

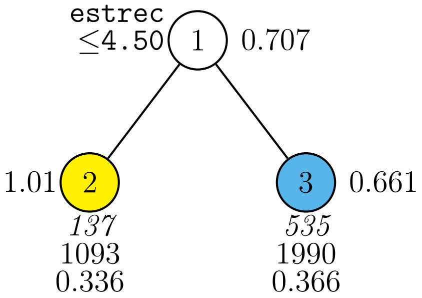

Figure 27: GUIDE v.38.0 0.25-SE (based on median CV estimate of error) proportional hazards regression tree using Gi option for `time` and event indicator `death` with simple linear prognostic effects. Tree constructed with 672 observations (excluding observations with non-positive weight or missing values in `D`, `T`, `R` or `Z` variables). Maximum number of split levels is 10, minimum node sample size is 7 and minimum treatment fraction is 0.072. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*), median survival time, and proportion of `horTh` = `yes` printed below nodes. Treatment `horTh` hazard ratio of level `yes` to `no` beside nodes. Terminal nodes with treatment hazard ratio above and below 0.707 (ratio at root node) are colored yellow and skyblue respectively. Second best split variable at root node is `progrec`.

Input file generation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: rest-gi.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: rest-gi.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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended if there are no missing values)
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple polynomial in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
0: stepwise linear, 1: multiple linear, 2: simple polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <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: 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
6 N variables changed to S
Warning: model changed to linear in treatment
D 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 ...
Finished checking
Assigning integer codes to values of 2 categorical variables
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to missing values if any ...
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: 1
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):

Creating dummy variables ...
Rereading data ...
Largest uncensored and censored time by horTh
 "no"      2456.0000    2563.0000
"yes"      2372.0000    2659.0000
Smallest observed uncensored time is 72.0000
Largest observed censored or uncensored time is 2659.0000
Input restriction on event time ([72.00:2659.00], <cr>=1222.00):

Proportion of training sample for each level of horTh
 "no"    0.6360
"yes"    0.3640
     Total  #cases w/   #missing
    #cases    miss. D  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
No weight variable in data file
Number of cases used for training: 533
Number of split variables: 7
Number of dummy variables created: 1
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): rest-gi.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: rest-gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: rest-gi.r
Input rank of top variable to split root node ([1:9], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < rest-gi.in
```

Results

```
Restricted mean event time regression tree
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
6 N variables changed to S
Warning: model changed to linear in treatment
D 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"
Number of dummy variables created: 1
Smallest uncensored time: 72.0000
Largest uncensored and censored time by horTh
horTh     Uncensored     Censored
 "no"      2456.0000    2563.0000
"yes"      2372.0000    2659.0000
Interval for restricted mean event time is from 0 to 1222.
Proportion of training sample for each level of horTh
 "no"    0.6360
"yes"    0.3640

Summary information for training sample of size 533 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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  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        36.00
      7  progrec    s    0.000        1490.
      8  estrec     s    0.000        1091.
      9  time       t    72.00        2659.
     10  death      d    0.000        1.000
  =================== Constructed variables ===================
     11  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        6
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        1        0        1
No weight variable in data file
Number of cases used for training: 533
Number of split variables: 7
Number of dummy variables created: 1

Missing values imputed with node means for fitting regression models in nodes
Predictive priority (Gi) using restricted mean event time
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 10
Minimum node sample size: 5
Minimum fraction of cases per treatment at each node: 0.073
Top-ranked variables and chi-squared values at root node
     1  0.1169E+02   estrec
     2  0.2062E+01   progrec
     3  0.1847E+01   tgrade
     4  0.4400E+00   age
     5  0.3773E+00   pnodes
     6  0.2634E+00   menostat
     7  0.1340E+00   tsize

Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       60   5.252E+05   2.825E+04   1.526E+04   5.295E+05   1.788E+04
   2       59   5.252E+05   2.825E+04   1.526E+04   5.295E+05   1.788E+04
   :
  38        2   4.437E+05   2.183E+04   1.070E+04   4.441E+05   1.700E+04
  39**      1   4.338E+05   1.732E+04   6.012E+03   4.385E+05   7.335E+03

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.
```

```
 D-mean is weighted mean of death in the node
 Cases fit give the number of cases used to fit node
 MSE and R^2 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     533      533     2  9.873E+02  1.519E+05  0.0106 estrec
 Best split at root node is estrec <= 8.5000

 Number of terminal nodes of final tree: 1
 Total number of nodes of final tree: 1
 Best split variable (based on curvature test) at root node is estrec

 Regression tree:

 Node 1: terminal

 ***************************************************************


Node 1: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      960.8        51.78        0.000
 horTh.yes     73.85        2.385       0.1744E-01    0.000       0.3591        1.000
 time mean = 987.273
 No truncation of predicted values
 ----------------------------
 Number of times Li-Martin approximation used = 1
 Observed and fitted values are stored in rest-gi.fit
 LaTeX code for tree is in rest-gi.tex
 R code is stored in rest-gi.r
```

11.3.2 With linear prognostic control

A trivial tree is obtained too if a linear regressor is included 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: rest-lin-gi.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: rest-lin-gi.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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended if there are no missing values)
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple polynomial in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
0: stepwise linear, 1: multiple linear, 2: simple polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <cr>=3): 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: 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
D 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 ...
Finished checking
Assigning integer codes to values of 2 categorical variables
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to missing values if any ...
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: 1
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):

Creating dummy variables ...
Rereading data ...
```

```
Largest uncensored and censored time by horTh
 "no"      2456.0000    2563.0000
"yes"      2372.0000    2659.0000
Smallest observed uncensored time is 72.0000
Largest observed censored or uncensored time is 2659.0000
Input restriction on event time ([72.00:2659.00], <cr>=1222.00):

Proportion of training sample for each level of horTh
 "no"    0.6360
"yes"    0.3640
     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
No weight variable in data file
Number of cases used for training: 533
Number of split variables: 7
Number of dummy variables created: 1
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): rest-lin-gi.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: rest-lin-gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: rest-lin-gi.r
Input rank of top variable to split root node ([1:9], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < rest-lin-gi.in
```

Results

```
Restricted mean event time regression tree
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
D variable is death
Piecewise simple linear or constant model
Powers are dropped if they are not significant at level 0.0500
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"
Number of dummy variables created: 1
Smallest uncensored time: 72.0000
Largest uncensored and censored time by horTh
horTh     Uncensored     Censored
 "no"      2456.0000    2563.0000
"yes"      2372.0000    2659.0000
Interval for restricted mean event time is from 0 to 1222.
Proportion of training sample for each level of horTh
 "no"    0.6360
"yes"    0.3640

Summary information for training sample of size 533 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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  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        36.00
      7  progrec    n    0.000        1490.
      8  estrec     n    0.000        1091.
      9  time       t    72.00        2659.
     10  death      d    0.000        1.000
  =================== Constructed variables ===================
     11  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
No weight variable in data file
Number of cases used for training: 533
Number of split variables: 7
Number of dummy variables created: 1

Missing values imputed with node means for fitting regression models in nodes
Predictive priority (Gi) using restricted mean event time
```

```
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 10
Minimum node sample size: 6
Minimum fraction of cases per treatment at each node: 0.073
Top-ranked variables and chi-squared values at root node
     1  0.1193E+02   estrec
     2  0.2708E+01   progrec
     3  0.2007E+01   tgrade
     4  0.1079E+01   age
     5  0.6277E+00   menostat
     6  0.2553E+00   pnodes
     7  0.8480E-02   tsize

Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       47   8.295E+05   1.115E+05   1.027E+05   7.154E+05   8.867E+04
   2       46   8.295E+05   1.115E+05   1.027E+05   7.154E+05   8.867E+04
   :
  28        2   6.944E+05   9.592E+04   9.797E+04   5.880E+05   5.445E+04
  29**      1   3.811E+05   1.674E+04   7.817E+03   3.778E+05   1.110E+04

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.

D-mean is weighted mean of death in the node
Cases fit give the number of cases used to fit node
MSE and R^2 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     533      533     3  9.873E+02  1.335E+05  0.1320 estrec  -pnodes
Best split at root node is estrec <= 7.5000

Number of terminal nodes of final tree: 1
```

```
 Total number of nodes of final tree: 1
 Best split variable (based on curvature test) at root node is estrec

 Regression tree:

 Node 1: terminal

 ***************************************************************


Node 1: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      1072.        49.43        0.000
 pnodes       -23.75       -8.612       0.1110E-15    1.000         4.848        36.00
 horTh.yes     83.61        2.878       0.4164E-02
 time mean = 987.273
 No truncation of predicted values
 ----------------------------
 Observed and fitted values are stored in rest-lin-gi.fit
 LaTeX code for tree is in rest-lin-gi.tex
 R code is stored in rest-lin-gi.r
```

12 Observational studies

A classification tree was built in Section 4 to predict the occurence of right heart catheterization (RHC), which is a treatment used to treat critically ill patients with heart problems. GUIDE can fit a tree model to find subgroups where the treatment (represented by variable `swang1`) is beneficial or not for survival. This is done by specifying the treatment variable as "`r`" and the event variable `death` (1=die, 0=not die) as "`d`" in the description file `rhcdsc3.txt` below.

```
rhcdata.txt
NA
2
1 X x
2 cat1 c
3 cat2 c
4 ca c
5 sadmdte x
6 dschdte x
7 dthdte x
8 lstctdte x
9 death d
```

```
10 cardiohx c
11 chfhx c
12 dementhx c
13 psychhx c
14 chrpulhx c
15 renalhx c
16 liverhx c
17 gibledhx c
18 malighx c
19 immunhx c
20 transhx c
21 amihx c
22 age n
23 sex c
24 edu n
25 surv2md1 n
26 das2d3pc n
27 t3d30 x
28 dth30 x
29 aps1 n
30 scoma1 n
31 meanbp1 n
32 wblc1 n
33 hrt1 n
34 resp1 n
35 temp1 n
36 pafi1 n
37 alb1 n
38 hema1 n
39 bili1 n
40 crea1 n
41 sod1 n
42 pot1 n
43 paco21 n
44 ph1 n
45 swang1 r
46 wtkilo1 n
47 dnr1 c
48 ninsclas c
49 resp c
50 card c
51 neuro c
52 gastr c
53 renal c
54 meta c
55 hema c
```

```
56 seps c
57 trauma c
58 ortho c
59 adld3p n
60 urin1 n
61 race c
62 income c
63 ptid x
64 survtime t
```

12.1 Censored response: proportional hazards

GUIDE can fit models with the Gi or Gs options. The Gi option is designed to be sensitive to detect *predictive* variables (variables that have interactions with the treatment variable) while Gs option is equally sensitive to such variables as well as *prognostic* variables (those thta have an effect on the outcome irrespective of the treatment). See Loh et al. (2015) for details.

12.1.1 Gi option

Gi input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: surv-gi.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: surv-gi.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
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
 Input choice ([1:7], <cr>=1): 4
 Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1):
 Choose complexity of model to use at each node:
 Choose 1 for multiple regression (not recommended if there are missing values)
 Choose 2 for simple linear in one N or F variable
 Choose 3 to fit a constant (recommended if there are missing values or an R variable)
 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: rhcdsc3.txt
Reading data description file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
20 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Smallest uncensored survtime: 2.0000
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
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 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):

Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime by swang1
"NoRHC"      1867.0000    1243.0000
  "RHC"      1943.0000    1351.0000
```

```
Proportion of training sample for each level of swang1
"NoRHC"    0.6192
  "RHC"    0.3808
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
Survival time variable in column: 64
Event indicator variable in column: 9
Proportion uncensored among nonmissing T and D variables: 0.649
Number of cases used for training: 5735
Number of split variables: 53
Number of dummy variables created: 1
Number of cases excluded due to 0 weight or missing D, T or R: 0
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): surv-gi.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: surv-gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: surv-gi.r
Input rank of top variable to split root node ([1:52], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < surv-gi.in
```

Contents of surv-gi.out

```
Regression tree for censored response
Pruning by cross-validation
Data description file: rhcdsc3.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
23 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Piecewise linear model
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
```

```
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
Number of dummy variables created: 1
Smallest uncensored survtime: 2.0000
Largest uncensored and censored survtime by swang1
 swang1     Uncensored     Censored
"NoRHC"      1867.0000    1243.0000
  "RHC"      1943.0000    1351.0000
Proportion of training sample for each level of swang1
"NoRHC"    0.6192
  "RHC"    0.3808

Summary information for training sample of size 5735
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
                                                  #Codes/
                                                  Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      2  cat1        c                                  9
      3  cat2        c                                  6     4535
      4  ca          c                                  3
      9  death       d    0.000        1.000
     10  cardiohx    c                                  2
      :
     44  ph1         s    6.579        7.770
     45  swang1      r                                  2
     46  wtkilo1     s    19.50        244.0                   515
      :
     62  income      c                                  4
     64  survtime    t    2.000        1943.
  =================== Constructed variables ===================
     65  lnbasehaz0  z   -3.818        2.038
     66  swang1.RHC  f    0.000        1.000

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
Survival time variable in column: 64
Event indicator variable in column: 9
Proportion uncensored among nonmissing T and D variables: 0.649
```

```
Number of cases used for training: 5735
Number of split variables: 53
Number of dummy variables created: 1
Number of cases excluded due to 0 weight or missing D, T or R: 0

Missing values imputed with node means for fitting regression models in nodes
Predictive priority (Gi)
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 15
Minimum node sample size: 7
Minimum fraction of cases per treatment at each node: 0.076
Number of iterations for fitting: 20
Top-ranked variables and chi-squared values at root node
     1  0.1323E+02   ph1
     2  0.1018E+02   resp1
     3  0.8324E+01   cat2
     4  0.7453E+01   pot1
     :
    35  0.1497E-01   sod1
    36  0.3221E-04   meanbp1

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1      499   2.105E+00   6.751E-02   5.520E-02   2.061E+00   5.624E-02
   2      498   2.105E+00   6.751E-02   5.520E-02   2.061E+00   5.624E-02
   :
 321       14   1.323E+00   1.610E-02   6.606E-03   1.334E+00   1.298E-02
 322**      5   1.322E+00   1.586E-02   7.111E-03   1.331E+00   1.190E-02
 323        1   1.367E+00   1.526E-02   6.317E-03   1.358E+00   9.980E-03

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
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix   Median      Node    Split
       label   cases      fit  rank   survtime   deviance  variable
          1     5735     5735     1  1.920E+02  1.367E+00  ph1
          2     1411     1411     1  1.150E+02  1.454E+00  cat2
          4T    1307     1307     1  1.570E+02  1.416E+00  paco21
          5T     104      104     1  1.400E+01  1.636E+00  malighx
          3     4324     4324     1  2.070E+02  1.334E+00  resp1
          6     3341     3341     1  2.200E+02  1.333E+00  paco21
         12T     687      687     1  6.900E+01  1.531E+00  income
         13T    2654     2654     1  2.390E+02  1.265E+00  paco21
          7T     983      983     1  1.640E+02  1.319E+00  hrt1

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 resp1

Regression tree:
For categorical variable splits, values not in training data go to the right

Node 1: ph1 <= 7.3344730
  Node 2: cat2 = "MOSF w/Sepsis", "NA"
    Node 4: Median survival time = 157.00000
  Node 2: cat2 /= "MOSF w/Sepsis", "NA"
    Node 5: Median survival time = 14.000000
Node 1: ph1 > 7.3344730 or NA
  Node 3: resp1 <= 38.500000 or NA
    Node 6: paco21 <= 29.498050
      Node 12: Median survival time = 69.000000
    Node 6: paco21 > 29.498050 or NA
      Node 13: Median survival time = 239.00000
  Node 3: resp1 > 38.500000
    Node 7: Median survival time = 164.00000

***************************************************************

Predictor means below are means of cases with no missing values.
Regression coefficients are computed from the complete cases.

WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
```

```
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if ph1 <= 7.3344730
 ph1 mean = 7.3884135
 Coefficients of log-relative hazard function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       0.000
 swang1.RHC    0.1504        4.494       0.7131E-05    0.000       0.3808        1.000
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if cat2 = "MOSF w/Sepsis", "NA"
 cat2 mode = "NA"
 ----------------------------
Node 4: Terminal node
 Coefficients of log-relative hazard function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -0.6181E-01
 swang1.RHC    0.4067        6.034       0.2086E-08    0.000       0.4499        1.000
 ----------------------------
Node 5: Terminal node
 Coefficients of log-relative hazard function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.8005
 swang1.RHC   -0.3295       -1.558       0.1223        0.000       0.3558        1.000
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if resp1 <= 38.500000 or NA
 resp1 mean = 28.418652
 ----------------------------
Node 6: Intermediate node
 A case goes into Node 12 if paco21 <= 29.498050
 paco21 mean = 36.054906
 ----------------------------
Node 12: Terminal node
 Coefficients of log-relative hazard function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.3006
 swang1.RHC   -0.3237E-01  -0.3424       0.7322        0.000       0.3916        1.000
 ----------------------------
Node 13: Terminal node
 Coefficients of log-relative hazard function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -0.7105E-01
 swang1.RHC    0.5937E-02   0.1159       0.9078        0.000       0.3632        1.000
 ----------------------------
Node 7: Terminal node
 Coefficients of log-relative hazard function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -0.1150E-01
 swang1.RHC    0.3555        4.329       0.1651E-04    0.000       0.3316        1.000
 ----------------------------
 Observed and fitted values are stored in surv-gi.fit
 LaTeX code for tree is in surv-gi.tex
 R code is stored in surv-gi.r
```

Figure 28 shows the tree diagram. The following contents of `surv-gi.r` give the R function for retrieving the node numbers and regression coefficients from the tree structure.

```
predicted <- function(){
 if(!is.na(ph1) & ph1 <= 7.33447300000 ){
   catvalues <- c("MOSF w/Sepsis","NA")
   catvalues <- c(catvalues,NA)
   if(is.na(cat2) | cat2 %in% catvalues){
     nodeid <- 4
     predict <- c(-0.618134773832E-1,0.406689682597)
   } else {
     nodeid <- 5
     predict <- c(0.800468154417,-0.329463311994)
   }
 } else {
   if(is.na(resp1) | resp1 <= 38.5000000000 ){
     if(!is.na(paco21) & paco21 <= 29.4980500000 ){
       nodeid <- 12
       predict <- c(0.300583118160,-0.323677803504E-1)
     } else {
       nodeid <- 13
       predict <- c(-0.710500703708E-1,0.593672033426E-2)
     }
   } else {
     nodeid <- 7
     predict <- c(-0.114966933127E-1,0.355516696179)
   }
 }
 return(c(nodeid,predict))
}
## end of function
##
```

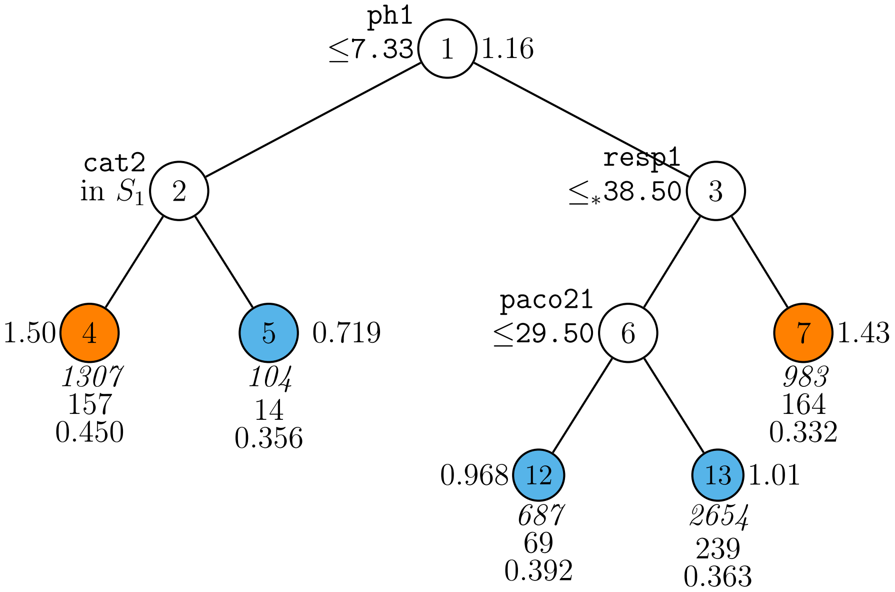

Figure 28: GUIDE v.38.0 0.25-SE proportional hazards regression tree using Gi option for `survtime` and event indicator `death` without linear prognostic effects. Tree constructed with 5735 observations. Maximum number of split levels is 15, minimum node sample size is 7 and minimum treatment fraction is 0.076. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '≤ or missing'. Set S_1 = {`MOSF w/Sepsis`, `NA`}. Treatment `swang1` hazard ratio of level `RHC` to `NoRHC` beside nodes. Sample size (in *italics*), median survival time, and proportion of `swang1` = `RHC` printed below nodes. Terminal nodes with treatment hazard ratio above and below 1.162 (ratio at root node) are colored orange and skyblue respectively. Second best split variable at root node is `resp1`.

```
swang1.values <- c("NoRHC","RHC")
##
## newdata.txt is the file containing the data to be predicted
## Missing value code is NA
newdata <- read.table("newdata.txt",header=TRUE,colClasses="character")
## node contains terminal node ID of each case
## coefs contain regression coefficients
node <- NULL
coefs <- NULL
for(i in 1:nrow(newdata)){
    cat2 <- as.character(newdata$cat2[i])
    resp1 <- as.numeric(newdata$resp1[i])
    paco21 <- as.numeric(newdata$paco21[i])
    ph1 <- as.numeric(newdata$ph1[i])
    swang1 <- as.character(newdata$swang1[i])
    if(swang1 %in% swang1.values){
        swang1.RHC <- if(swang1 == "RHC") 1 else 0
    } else {
        swang1.RHC <- NA
    }
    tmp <- predicted()
    node <- c(node,as.numeric(tmp[1]))
    coefs <- rbind(coefs,tmp[-1])
}
```

12.1.2 Gs option

Gs input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: surv-gs.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: surv-gs.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
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
 Input choice ([1:7], <cr>=1): 4
 Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1):
 Choose complexity of model to use at each node:
```

```
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple linear in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
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: rhcdsc3.txt
Reading data description file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
20 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Smallest uncensored survtime: 2.0000
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
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 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2): 1
Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
```

```
Largest uncensored and censored survtime by swang1
"NoRHC"      1867.0000    1243.0000
  "RHC"      1943.0000    1351.0000
Proportion of training sample for each level of swang1
"NoRHC"    0.6192
  "RHC"    0.3808
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
Survival time variable in column: 64
Event indicator variable in column: 9
Proportion uncensored among nonmissing T and D variables: 0.649
Number of cases used for training: 5735
Number of split variables: 53
Number of dummy variables created: 1
Number of cases excluded due to 0 weight or missing D, T or R: 0
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): surv-gs.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: surv-gs.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: surv-gs.r
Input rank of top variable to split root node ([1:52], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < surv-gs.in
```

Contents of surv-gs.out

```
Regression tree for censored response
Pruning by cross-validation
Data description file: rhcdsc3.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
23 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Piecewise linear model
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
```

```
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
Number of dummy variables created: 1
Smallest uncensored survtime: 2.0000
Largest uncensored and censored survtime by swang1
 swang1     Uncensored     Censored
"NoRHC"      1867.0000    1243.0000
  "RHC"      1943.0000    1351.0000
Proportion of training sample for each level of swang1
"NoRHC"    0.6192
  "RHC"    0.3808

Summary information for training sample of size 5735
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
                                                  #Codes/
                                                  Levels/
 Column  Name            Minimum      Maximum     Periods   #Missing
      2  cat1        c                                  9
      3  cat2        c                                  6     4535
      4  ca          c                                  3
      9  death       d   0.000        1.000
     10  cardiohx    c                                  2
      :
     44  ph1         s   6.579        7.770
     45  swang1      r                                  2
      :
     62  income      c                                  4
     64  survtime    t   2.000        1943.
  =================== Constructed variables ===================
     65  lnbasehaz0  z   -3.818       2.038
     66  swang1.RHC  f   0.000        1.000

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
Survival time variable in column: 64
```

```
Event indicator variable in column: 9
Proportion uncensored among nonmissing T and D variables: 0.649
Number of cases used for training: 5735
Number of split variables: 53
Number of dummy variables created: 1
Number of cases excluded due to 0 weight or missing D, T or R: 0

Missing values imputed with node means for fitting regression models in nodes
Prognostic priority (Gs)
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 15
Minimum node sample size: 7
Minimum fraction of cases per treatment at each node: 0.076
Number of iterations for fitting: 20
Top-ranked variables and chi-squared values at root node
     1  0.5838E+03   surv2md1
     2  0.2041E+03   ca
     3  0.1954E+03   das2d3pc
     4  0.1863E+03   malighx
     :
    35  0.8147E+00   sex
    36  0.3710E-02   race

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1      474   1.782E+00   4.039E-02   3.576E-02   1.773E+00   5.580E-02
   2      473   1.782E+00   4.039E-02   3.576E-02   1.773E+00   5.580E-02
   :
  294      10   1.267E+00   1.714E-02   1.099E-02   1.274E+00   2.001E-02
 295++      8   1.263E+00   1.702E-02   1.130E-02   1.269E+00   1.975E-02
 296**      6   1.266E+00   1.693E-02   1.030E-02   1.278E+00   1.552E-02
 297        5   1.274E+00   1.682E-02   1.134E-02   1.283E+00   2.106E-02
 298        4   1.289E+00   1.687E-02   1.531E-02   1.283E+00   1.530E-02
 299        2   1.325E+00   1.687E-02   1.141E-02   1.321E+00   1.525E-02
 300        1   1.438E+00   1.619E-02   6.455E-03   1.428E+00   1.061E-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
++ 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
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix   Median      Node    Split
       label   cases      fit  rank   survtime   deviance  variable
          1     5735     5735     1  1.920E+02  1.438E+00  surv2md1
          2     2164     2164     1  2.300E+01  1.474E+00  adld3p
          4     1930     1930     1  1.800E+01  1.503E+00  surv2md1
          8T     709      709     1  1.100E+01  1.400E+00  meanbp1
          9T    1221     1221     1  2.800E+01  1.473E+00  crea1
          5T     234      234     1  1.950E+02  8.937E-01  age
          3     3571     3571     1  3.290E+02  1.210E+00  das2d3pc
          6T    2730     2730     1  2.840E+02  1.218E+00  age
          7      841      841     1  5.030E+02  1.090E+00  surv2md1
         14T     455      455     1  4.160E+02  1.229E+00  urin1
         15T     386      386     1  7.460E+02  7.910E-01  pafi1

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 ca

Regression tree:

Node 1: surv2md1 <= 0.56447053
  Node 2: adld3p = NA
    Node 4: surv2md1 <= 0.35847378
      Node 8: Median survival time = 11.000000
    Node 4: surv2md1 > 0.35847378 or NA
      Node 9: Median survival time = 28.000000
  Node 2: adld3p /= NA
    Node 5: Median survival time = 195.00000
Node 1: surv2md1 > 0.56447053 or NA
  Node 3: das2d3pc <= 24.890625
    Node 6: Median survival time = 284.00000
  Node 3: das2d3pc > 24.890625 or NA
    Node 7: surv2md1 <= 0.74340820
      Node 14: Median survival time = 416.00000
```

```
     Node 7: surv2md1 > 0.74340820 or NA
       Node 15: Median survival time = 746.00000

 ***************************************************************

 Predictor means below are means of cases with no missing values.
 Regression coefficients are computed from the complete cases.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if surv2md1 <= 0.56447053
 surv2md1 mean = 0.59245008
 Coefficients of log-relative hazard function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       0.000
 swang1.RHC    0.1527        4.564       0.5127E-05    0.000       0.3808        1.000
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if adld3p = NA
 adld3p mean = 1.3589744
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if surv2md1 <= 0.35847378
 surv2md1 mean = 0.38175857
 ----------------------------
Node 8: Terminal node
 Coefficients of log-relative hazard function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       1.080
 swang1.RHC   -0.3028E-01  -0.3868       0.6990        0.000       0.4485        1.000
 ----------------------------
Node 9: Terminal node
 Coefficients of log-relative hazard function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.5441
 swang1.RHC   -0.4152E-01  -0.6510       0.5151        0.000       0.4496        1.000
 ----------------------------
Node 5: Terminal node
 Coefficients of log-relative hazard function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.1839
 swang1.RHC   -0.4255       -2.357       0.1923E-01    0.000       0.3162         1.000
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if das2d3pc <= 24.890625
 das2d3pc mean = 20.731454
 ----------------------------
Node 6: Terminal node
 Coefficients of log-relative hazard function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -0.2421
 swang1.RHC    0.1607        3.106       0.1918E-02    0.000       0.3407         1.000
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if surv2md1 <= 0.74340820
 surv2md1 mean = 0.72699431
 ----------------------------
Node 14: Terminal node
 Coefficients of log-relative hazard function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -0.4105
 swang1.RHC    0.1108       0.7854       0.4326        0.000       0.4000         1.000
 ----------------------------
Node 15: Terminal node
 Coefficients of log-relative hazard function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      -1.303
 swang1.RHC    0.7566E-01   0.3276       0.7434        0.000       0.3394         1.000
 ----------------------------
 Observed and fitted values are stored in surv-gs.fit
 LaTeX code for tree is in surv-gs.tex
 R code is stored in surv-gs.r
```

Figure 29 shows the tree. The estimated hazard ratios (`RHC` vs `NoRHC`) beside the terminal nodes indicate that RHC seldom reduces the hazard of death. Following are the top 3 lines of the file `surv-gs.fit`

```
train  node  obs_survtime  logbasecumhaz   survivalprob  mediansurvtime    swang1.RHC
    y     6  2.40000E+02+  -3.029494E-01   5.600147E-01    2.840000E+02   1.607011E-01
    y     6  4.50000E+01   -8.381673E-01   6.712070E-01    2.840000E+02   1.607011E-01
    y     8  3.17000E+02+  -9.838655E-02   7.500497E-02    1.100000E+01  -3.027736E-02
```

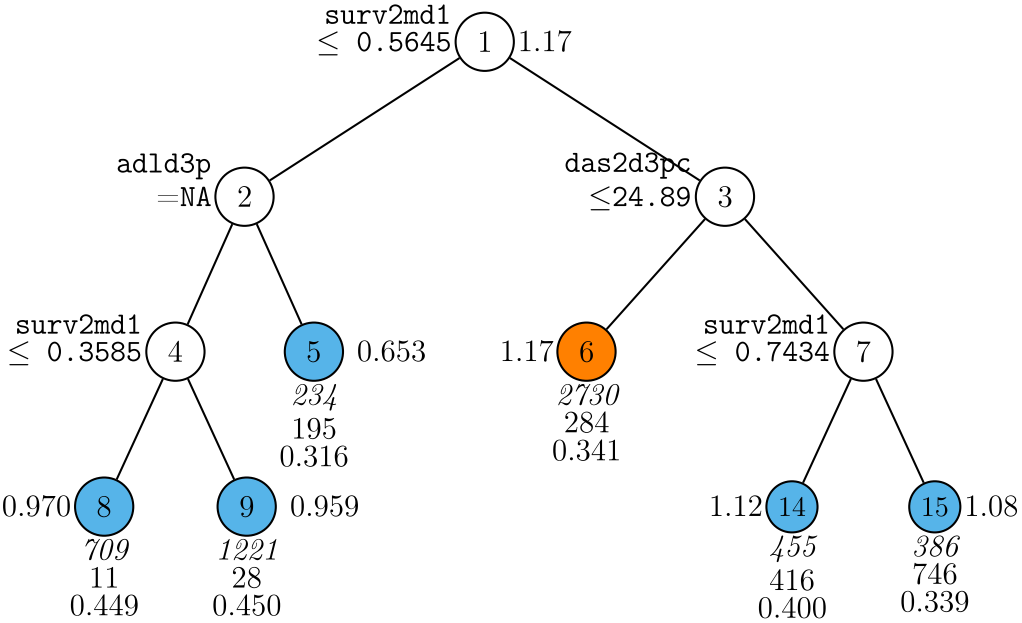

Figure 29: GUIDE v.38.0 0.25-SE proportional hazards regression tree using Gs option for `survtime` and event indicator `death` without linear prognostic effects. Tree constructed with 5735 observations. Maximum number of split levels is 15, minimum node sample size is 7 and minimum treatment fraction is 0.076. At each split, an observation goes to the left branch if and only if the condition is satisfied. Treatment `swang1` hazard ratio of level `RHC` to `NoRHC` beside nodes. Sample size (in *italics*), median survival time, and proportion of `swang1` = `RHC` printed below nodes. Terminal nodes with treatment hazard ratio above and below 1.165 (ratio at root node) are colored orange and skyblue respectively. Second best split variable at root node is `ca`.

The column definitions are

`train`: "y" if the observation is used for model fitting, "n" if not.

`node`: terminal node label of observation.

`obs_survtime`: observed survival time t, with trailing plus $(+)$ sign if censored.

`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.

`relativehaz`: $\exp(\boldsymbol{\beta}'\mathbf{x})$, risk of death relative to the average for the sample, where $\mathbf{x}$ is the covariate vector of the observation and $\boldsymbol{\beta}$ is the estimated regression coefficient vector in the node. For example, the first observation has `swang1` = "`NoRHC`" and is in terminal node 6 with $\beta_0 = -0.242135921383$ and $\beta_1 = 0.160701079661$ (see `surv-gs.r` or the abbreviated values in `surv-gs.out`). Thus its `relativehaz` $= \exp(\boldsymbol{\beta}'\mathbf{x}) = \exp(-0.242135921383 + 0.160701079661 \times I(\texttt{swang1} = \texttt{RHC})) = 0.7849495$.

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

$$\begin{aligned}\exp\{-\Lambda_0(t)\exp(\boldsymbol{\beta}'\mathbf{x})\} &= \exp\{-\exp(\beta_0 + \texttt{logbasecumhaz})\} \\ &= \exp(-\exp(-0.242135921383 - 0.3029494)) \\ &= 0.5600147\end{aligned}$$

where $t = 240$ and $\beta_0 = -0.242135921383$ is the constant term in the node (`surv-gs.r` gives β_0 to higher precision than `surv-gs.out`).

`mediansurvtime`: median survival time among observations in node estimated from Kaplan-Meier survival function. A trailing plus (+) sign indicates estimate is censored.

`swang1.RHC`: estimated treatment effect β_1 for level `RHC` of `swang1`. For the first obsservation, because $\beta_1 = 0.160701079661$, its relative hazard of `RHC` vs `NoRHC` is $\exp(\beta_1) = \exp(0.160701079661) = 1.174334$ (value is printed beside node 6 in Figure 29).

12.2 Censored response: restricted mean

GUIDE can also construct a tree model such that a restricted mean event time (Chen and Tsiatis, 2001; Tian et al., 2014) is fitted in each node of the tree.

12.2.1 Gi option

Gi input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: rest-gi.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: rest-gi.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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended if there are no missing values)
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple polynomial in one N or F variable
Choose 3 to fit a constant (recommended if there are missing values or an R variable)
0: stepwise linear, 1: multiple linear, 2: simple polynomial, 3: constant,
4: stepwise simple ANCOVA ([0:4], <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: rhcdsc3.txt
Reading data description file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
20 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
```

```
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
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: 1
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):

Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime by swang1
"NoRHC"      1867.0000    1243.0000
  "RHC"      1943.0000    1351.0000
Smallest observed uncensored time is 2.0000
Largest observed censored or uncensored time is 1943.0000
Input restriction on event time ([2.00:1943.00], <cr>=622.00):

Proportion of training sample for each level of swang1
"NoRHC"    0.5993
  "RHC"    0.4007
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
No weight variable in data file
Number of cases used for training: 3763
Number of split variables: 53
Number of dummy variables created: 1
Number of cases excluded due to 0 weight or missing D or R: 1972
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): rest-gi.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: rest-gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1):
Input rank of top variable to split root node ([1:52], <cr>=1):
```

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

Contents of rest-gi.out

```
Restricted mean event time regression tree
Pruning by cross-validation
Data description file: rhcdsc3.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
23 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Piecewise linear model
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Number of dummy variables created: 1
Smallest uncensored survtime: 2.0000
Largest uncensored and censored survtime by swang1
 swang1     Uncensored     Censored
"NoRHC"      1867.0000    1243.0000
  "RHC"      1943.0000    1351.0000
Interval for restricted mean event time is from 0 to 622.
Proportion of training sample for each level of swang1
"NoRHC"    0.5993
  "RHC"    0.4007

Summary information for training sample of size 3763 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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
      2  cat1        c                                   9
      3  cat2        c                                   6     2836
      4  ca          c                                   3
```

```
     9  death       d    0.000       1.000
    10  cardiohx    c                                2
     :
    45  swang1      r                                2
    46  wtkilo1     s    24.10       200.8                   315
     :
    62  income      c                                4
    64  survtime    t    2.000       1943.
  ================== Constructed variables ===================
    65  swang1.RHC  f    0.000       1.000

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
No weight variable in data file
Number of cases used for training: 3763
Number of split variables: 53
Number of dummy variables created: 1
Number of cases excluded due to 0 weight or missing D or R: 1972

Missing values imputed with node means for fitting regression models in nodes
Predictive priority (Gi) using restricted mean event time
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 13
Minimum node sample size: 7
Minimum fraction of cases per treatment at each node: 0.080
Top-ranked variables and chi-squared values at root node
     1  0.9407E+01   scoma1
     2  0.7887E+01   ph1
     3  0.7551E+01   pafi1
     :
    37  0.1688E-01   meanbp1
    38  0.4169E-02   cat1

Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1      325   1.644E+05   5.598E+03   3.914E+03   1.652E+05   6.405E+03
   2      324   1.644E+05   5.598E+03   3.914E+03   1.652E+05   6.405E+03
   :
```

```
 217        3   1.295E+05   4.444E+03   4.786E+03   1.294E+05   6.909E+03
 218**      2   1.157E+05   3.411E+03   2.378E+03   1.141E+05   3.229E+03
 219        1   1.198E+05   3.143E+03   9.972E+02   1.190E+05   1.421E+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.

D-mean is weighted mean of death in the node
Cases fit give the number of cases used to fit node
MSE and R^2 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
          1     3763     3763     2  2.583E+02  9.489E+04  0.0043 scoma1
          2T    3124     3124     2  2.781E+02  9.938E+04  0.0075 pafi1
          3T     639      639     2  1.333E+02  4.975E+04  0.0016 sod1

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 ph1

Regression tree:

Node 1: scoma1 <= 49.500000
  Node 2: terminal
Node 1: scoma1 > 49.500000 or NA
  Node 3: terminal

***************************************************************

Predictor means below are means of cases with no missing values.
Regression coefficients are computed from the complete cases.

WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
```

```
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if scoma1 <= 49.500000
 scoma1 mean = 20.462797
 Coefficients of least squares regression function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       271.2        52.27        0.000
 swang1.RHC    -33.80       -4.020       0.5926E-04    0.000       0.3808        1.000
 survtime mean = 258.284
 No truncation of predicted values
 ----------------------------
Node 2: Terminal node
 Coefficients of least squares regression functions:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       295.7        51.17        0.000
 swang1.RHC    -44.75       -4.866       0.1195E-05    0.000       0.3949        1.000
 survtime mean = 278.051
 No truncation of predicted values
 ----------------------------
Node 3: Terminal node
 Coefficients of least squares regression functions:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       138.4        14.56        0.000
 swang1.RHC    -17.66       -1.003       0.3161        0.000       0.2916        1.000
 survtime mean = 133.272
 No truncation of predicted values
 ----------------------------
 Number of times Li-Martin approximation used = 394
 Observed and fitted values are stored in rest-gi.fit
 LaTeX code for tree is in rest-gi.tex
```

Figure 30 shows the Gi restricted mean event time tree.

12.2.2 Gs option

The results for the Gs follow.

Contents of `rest-gs.out`

```
 Restricted mean event time regression tree
 Pruning by cross-validation
```

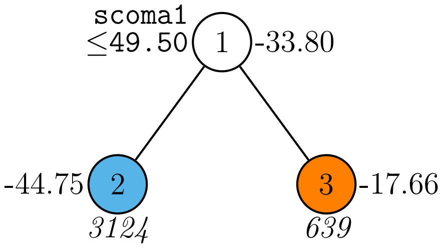

Figure 30: GUIDE v.38.0 0.25-SE regression tree using Gi option for mean `survtime` restricted to less than 622.00 without linear prognostic effects. Tree constructed with 3763 observations (excluding observations with non-positive weight or missing values in `D`, `T`, `R` or `Z` variables). Maximum number of split levels is 13, minimum node sample size is 7 and minimum treatment fraction is 0.080. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size *(in italics)* printed below nodes. Treatment `swang1` effect for level `RHC` (relative to `NoRHC`) beside nodes. Terminal nodes with treatment effect above and below -33.80 (effect at root node) are colored orange and skyblue respectively. Second best split variable at root node is `ph1`.

```
Data description file: rhcdsc3.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
23 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Piecewise linear model
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Number of dummy variables created: 1
Smallest uncensored survtime: 2.0000
Largest uncensored and censored survtime by swang1
 swang1     Uncensored     Censored
"NoRHC"      1867.0000    1243.0000
  "RHC"      1943.0000    1351.0000
Interval for restricted mean event time is from 0 to 622.
Proportion of training sample for each level of swang1
"NoRHC"    0.5993
```

```
  "RHC"    0.4007

Summary information for training sample of size 3763 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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
      2  cat1        c                                  9
      3  cat2        c                                  6     2836
      4  ca          c                                  3
      9  death       d    0.000        1.000
      :
     44  ph1         s    6.579        7.770
     45  swang1      r                                  2
      :
     62  income      c                                  4
     64  survtime    t    2.000        1943.
  =================== Constructed variables ===================
     65  swang1.RHC  f    0.000        1.000

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
No weight variable in data file
Number of cases used for training: 3763
Number of split variables: 53
Number of dummy variables created: 1
Number of cases excluded due to 0 weight or missing D or R: 1972

Missing values imputed with node means for fitting regression models in nodes
Prognostic priority (Gs) using restricted mean event time
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 13
Minimum node sample size: 7
Minimum fraction of cases per treatment at each node: 0.080
```

```
Top-ranked variables and chi-squared values at root node
     1  0.2951E+03   adld3p
     2  0.2487E+03   surv2md1
     3  0.1741E+03   cat1
     :
    37  0.7371E+00   age
    38  0.2904E-01   race

Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1      335   1.441E+05   5.066E+03   4.356E+03   1.410E+05   5.174E+03
   2      334   1.441E+05   5.066E+03   4.356E+03   1.410E+05   5.174E+03
   :
 208        4   1.192E+05   4.107E+03   3.838E+03   1.178E+05   3.646E+03
 209**      3   1.073E+05   3.239E+03   2.787E+03   1.062E+05   4.219E+03
 210        2   1.088E+05   3.224E+03   1.786E+03   1.089E+05   2.183E+03
 211        1   1.198E+05   3.143E+03   9.972E+02   1.190E+05   1.421E+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 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.

D-mean is weighted mean of death in the node
Cases fit give the number of cases used to fit node
MSE and R^2 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
          1     3763     3763     2  2.583E+02  9.489E+04  0.0043 adld3p
          2T     676      676     2  3.853E+02  1.042E+05  0.0002 surv2md1
          3     3087     3087     2  2.172E+02  8.359E+04  0.0020 surv2md1
          6T    1206     1206     2  1.323E+02  5.057E+04  0.0023 temp1
          7T    1881     1881     2  2.605E+02  9.620E+04  0.0043 surv2md1

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 surv2md1

Regression tree:
 Node 1: adld3p <= 5.5000000
   Node 2: terminal
 Node 1: adld3p > 5.5000000 or NA
   Node 3: surv2md1 <= 0.47897136
     Node 6: terminal
   Node 3: surv2md1 > 0.47897136 or NA
     Node 7: terminal

 ***************************************************************

 Predictor means below are means of cases with no missing values.
 Regression coefficients are computed from the complete cases.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if adld3p <= 5.5000000
 adld3p mean = 1.2690165
 Coefficients of least squares regression function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       271.2        52.27        0.000
 swang1.RHC    -33.80       -4.020       0.5926E-04    0.000       0.3808        1.000
 survtime mean = 258.284
 No truncation of predicted values
 ----------------------------
Node 2: Terminal node
 Coefficients of least squares regression functions:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       383.5        37.66        0.000
 swang1.RHC     6.412       0.3323       0.7398        0.000       0.2786        1.000
 survtime mean = 385.335
 No truncation of predicted values
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if surv2md1 <= 0.47897136
 surv2md1 mean = 0.54299449
 ----------------------------
```

```
Node 6: Terminal node
 Coefficients of least squares regression functions:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       123.3        15.47        0.000
 swang1.RHC     19.79        1.671       0.9505E-01    0.000       0.4529        1.000
 survtime mean = 132.300
 No truncation of predicted values
 ----------------------------
Node 7: Terminal node
 Coefficients of least squares regression functions:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       273.9        36.68        0.000
 swang1.RHC    -33.99       -2.857       0.4326E-02    0.000       0.3939        1.000
 survtime mean = 260.514
 No truncation of predicted values
 ----------------------------
 Observed and fitted values are stored in rest-gs.fit
 LaTeX code for tree is in rest-gs.tex
```

Figure 31 shows the Gs restricted mean event time tree.

13 Multi-response: health service data

GUIDE has two options for fitting a piecewise-constant regression model to predict two or more dependent variables simultaneously (Loh and Zheng, 2013). The first (named `multiresponse` or option `5` in the input file) requires the number of dependent variables to be the same for each observation. Observations with missing values in one or more dependent variables are excluded. The second (named `longitudinal data (with T variables)` or option `6` in the input file) requires each dependent variable to be associated with an observation time variable. It fits a model to all observations, including those with missing values in some dependent variables. The observation times are not required to be the same for all subjects, i.e., they may vary from subject to subject, but observations with missing times are excluded from model fitting. We demonstrate the first option in this section. The second option is used in Section 14.

The data file `nmes.txt` contains observations on 4406 subjects from a National Medical Expenditure Survey (NMES) conducted in 1987 and 1988. Table 14 gives the names of the variables and their definitions. The data were previously analyzed in Deb and Trivedi (1997), Cameron and Trivedi (1998, chap. 6), and Zeileis (2006). Here we construct a regression tree to predict the outcomes for the first 6 variables

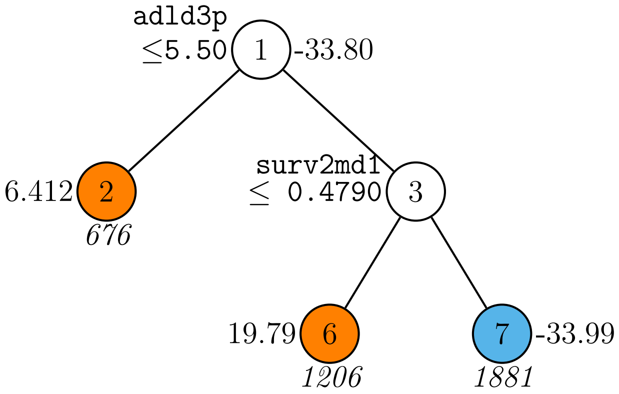

Figure 31: GUIDE v.38.0 0.25-SE regression tree using Gs option for mean `survtime` restricted to less than 622.00 without linear prognostic effects. Tree constructed with 3763 observations (excluding observations with non-positive weight or missing values in `D`, `T`, `R` or `Z` variables). Maximum number of split levels is 13, minimum node sample size is 7 and minimum treatment fraction is 0.080. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size *(in italics)* printed below nodes. Treatment `swang1` effect for level `RHC` (relative to `NoRHC`) beside nodes. Terminal nodes with treatment effect above and below -33.80 (effect at root node) are colored orange and skyblue respectively. Second best split variable at root node is `surv2md1`.

Table 14: Definitions of variables in NMES data

ofp	number of physician office visits
ofnp	number of nonphysician office visits
opp	number of physician outpatient visits
opnp	number of nonphysician outpatient visits
emer	number of emergency room visits
hosp	number of hospitalizations
health	self-perceived health (poor, average, or excellent)
numchron	number of chronic conditions
adldiff	has condition that limits daily living (no, yes)
region	region of U.S. (midwest, noreast, west, other)
age	age in years
black	African American (no, yes)
gender	sex (female, male)
married	married (no, yes)
school	number of years of education
faminc	family income in $10,000
employed	employed (no, yes)
privins	covered by private insurance (no, yes)
medicaid	covered by Medicaid (no, yes)

(`ofp`, `ofnp`, `opp`, `opnp`, `emer`, and `hosp`). The contents of the description file `nmes.dsc` follow.

```
nmes.txt
NA
1
1 ofp d
2 ofnp d
3 opp d
4 opnp d
5 emer d
6 hosp d
7 health c
8 numchron n
9 adldiff c
10 region c
11 age n
12 black c
13 gender c
```

```
14 married c
15 school n
16 faminc n
17 employed c
18 privins c
19 medicaid c
```

13.1 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
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
 Input choice ([1:7], <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: nmes.dsc
 Reading data description file ...
 Training sample file: nmes.txt
 Missing value code: NA
 Records in data file start on line 1
 4 N variables changed to S
 Number of D variables; 6
 D variables are:
 ofp
 ofnp
 opp
 opnp
 emer
 hosp
 Multivariate or univariate split variable selection:
 Choose multivariate if there is an order among the D variables;
 choose univariate otherwise or if item response
 Input 1 for multivariate, 2 for univariate ([1:2], <cr>=1): 2
```

```
D variables can be normalized to have unit variance,
e.g., if they have different scales or units
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: 4406
Length of longest entry in data file: 9
Checking for missing values ...
Finished checking
Assigning integer codes to values of 9 categorical variables
Re-checking data ...
Assigning codes to missing values if any ...
Data checks complete
Normalizing data
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
      4406          0          0        0        0        0        4
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        9        0
Number of cases used for training: 4406
Number of split variables: 13
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): mult.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: mult.nid
Input 2 to save fitted values at each terminal node; 1 otherwise ([1:2], <cr>=2):
Input name of file to store node fitted values: mult.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: mult.r
Input rank of top variable to split root node ([1:13], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < mult.in
```

13.2 Contents of mult.out

```
Multi-response or longitudinal data without T variables
Pruning by cross-validation
```

```
Data description file: nmes.dsc
Training sample file: nmes.txt
Missing value code: NA
Records in data file start on line 1
4 N variables changed to S
Number of D variables: 6
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: 4406
Length of longest entry in data file: 9
Model fitted to subset of observations with complete D values
Neither LDA nor PCA used

Summary information for training sample of size 4406
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  ofp        d    0.000        89.00
      2  ofnp       d    0.000        104.0
      3  opp        d    0.000        141.0
      4  opnp       d    0.000        155.0
      5  emer       d    0.000        12.00
      6  hosp       d    0.000        8.000
      7  health     c                                  3
      8  numchron   s    0.000        8.000
      9  adldiff    c                                  2
     10  region     c                                  4
     11  age        s    6.600        10.90
     12  black      c                                  2
     13  gender     c                                  2
     14  married    c                                  2
     15  school     s    0.000        18.00
     16  faminc     s   -1.012        54.84
     17  employed   c                                  2
     18  privins    c                                  2
     19  medicaid   c                                  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
```

```
      4406          0          0        0        0        0        4
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        9        0
Number of cases used for training: 4406
Number of split variables: 13

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 14
Minimum node sample size: 220
Top-ranked variables and chi-squared values at root node
     1  0.6017E+03   numchron
     2  0.3823E+03   health
     3  0.2025E+03   adldiff
     4  0.9838E+02   privins
     5  0.6583E+02   region
     6  0.5639E+02   age
     7  0.5257E+02   medicaid
     8  0.5218E+02   school
     9  0.3187E+02   gender
    10  0.3126E+02   black
    11  0.1892E+02   faminc
    12  0.1172E+02   married
    13  0.6155E+01   employed

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       15   1.944E+01   2.073E-01   6.220E-01   1.895E+01   7.603E-01
   2       14   1.894E+01   2.107E-01   5.038E-01   1.886E+01   3.985E-01
   3       13   1.894E+01   2.107E-01   5.038E-01   1.886E+01   3.985E-01
   4       12   1.894E+01   2.107E-01   5.038E-01   1.886E+01   3.985E-01
   5       11   1.774E+01   2.173E-01   6.896E-01   1.865E+01   8.712E-01
   6       10   1.736E+01   2.188E-01   1.013E+00   1.865E+01   8.712E-01
   7        9   1.372E+01   2.272E-01   1.614E+00   1.287E+01   3.674E+00
   8        8   1.217E+01   2.271E-01   1.466E+00   9.094E+00   2.333E+00
   9        7   8.592E+00   2.152E-01   1.195E+00   9.000E+00   7.831E-01
  10        6   7.465E+00   2.084E-01   9.112E-01   9.000E+00   1.293E+00
  11        5   3.612E+00   1.710E-01   3.239E-01   3.166E+00   5.668E-01
  12        4   1.185E+00   1.303E-01   1.470E-01   9.699E-01   1.007E-01
  13**      3   1.185E+00   1.303E-01   1.470E-01   9.699E-01   1.007E-01
  14        2   1.259E+00   1.296E-01   1.461E-01   1.068E+00   9.920E-02
  15        1   1.635E+00   1.308E-01   1.448E-01   1.421E+00   1.078E-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 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.

 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    Total    Cases       Node    Split
       label    cases      fit        MSE    variable
           1     4406     4406  1.000E+00    numchron
           2T    2523     2523  5.688E-01    numchron
           3     1883     1883  1.528E+00    health
           6T     426      426  2.282E+00    -
           7T    1457     1457  1.277E+00    privins

 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 health

 Regression tree for multi-response data:
 For categorical variable splits, values not in training data go to the right

 Node 1: numchron <= 1.5000000
   Node 2: Mean cost = 0.56857139
 Node 1: numchron > 1.5000000 or NA
   Node 3: health = "poor"
     Node 6: Mean cost = 2.2768607
   Node 3: health /= "poor"
     Node 7: Mean cost = 1.2765754

 ***************************************************************


Node 1: Intermediate node
 A case goes into Node 2 if numchron <= 1.5000000
 numchron mean = 1.5419882
 Means of ofp, ofnp, opp, opnp, emer, and hosp
   5.7744E+00   1.6180E+00   7.5079E-01   5.3609E-01   2.6350E-01
   2.9596E-01
 ----------------------------
Node 2: Terminal node
 Means of ofp, ofnp, opp, opnp, emer, and hosp
   4.4392E+00   1.4491E+00   4.6968E-01   3.9516E-01   1.6488E-01
   1.6647E-01
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if health = "poor"
 health mode = "average"
 ----------------------------
Node 6: Terminal node
 Means of ofp, ofnp, opp, opnp, emer, and hosp
   9.4319E+00   1.5000E+00   1.5282E+00   6.8310E-01   7.2066E-01
   7.9108E-01
 ----------------------------
Node 7: Terminal node
 Means of ofp, ofnp, opp, opnp, emer, and hosp
   7.0172E+00   1.9451E+00   1.0103E+00   7.3713E-01   3.0062E-01
   3.7543E-01
 ----------------------------
 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
 R code is stored in mult.r
```

The tree is shown in Figure 32. The file `mult.fit` saves the mean values of the dependent variables in each terminal node:

```
  node   ofp          ofnp         opp          opnp         emer         hosp
     2  0.44392E+01  0.14491E+01  0.46968E+00  0.39516E+00  0.16488E+00  0.16647E+00
     6  0.94319E+01  0.15000E+01  0.15282E+01  0.68310E+00  0.72066E+00  0.79108E+00
     7  0.70172E+01  0.19451E+01  0.10103E+01  0.73713E+00  0.30062E+00  0.37543E+00
```

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).

14 Longitudinal response with varying times

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;

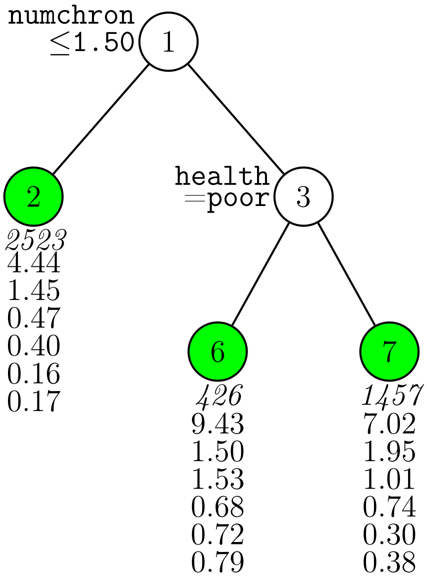

Figure 32: GUIDE v.38.0 0.25-SE regression tree for predicting response variables `ofp`, `ofnp`, `opp`, `opnp`, `emer`, and `hosp`, without using PCA at each node. Tree constructed with 4406 observations. Maximum number of split levels is 14 and minimum node sample size is 220. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (*in italics*) and predicted values of `ofp`, `ofnp`, `opp`, `opnp`, `emer`, and `hosp` printed below nodes. Second best split variable at root node is `health`.

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. Observation time points are indicated by `t`. The `d` and `t` variable columns may appear anywhere in the data, but the first `d` must be associated with the first `t`, second `d` with the second `t`, and so on. The number of `d` and `t` variables must be the same. Missing `d` values are permitted to allow for observations with unequal numbers of observation times. Observations with missing values in one or more `t` variable are excluded from model fitting.

```
wagedat.txt
NA
1
1 id x
2 hgc n
3 exper1 t
4 exper2 t
5 exper3 t
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
```

14.1 Input file creation

In the dialog below, we choose the 0-SE pruning rule because the default produces no splits.

```
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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <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 0 to get tree with fixed no. of nodes, 1 to prune by CV, 2 for no pruning ([0:2], <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
One N variable changed to S
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 ...
Finished checking
Missing values found in D variables
Assigning integer codes to values of 1 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
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
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        1        0
Number of cases used for training: 888
Number of split variables: 2
Number of 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>=0.25): 0
```

```
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 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): wage.tex
Choose a color for the terminal nodes:
(1) white
(2) lightgray
(3) aqua
(4) skyblue
(5) lime
(6) yellow
(7) red
(8) mauve
(9) green
(10) orange
(11) cyan
Input your choice ([1:11], <cr>=9):
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>=2):
Input name of file to store node fitted values: wage.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: wage.r
Input rank of top variable to split root node ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < wage.in
```

14.2 Contents of wage.out

```
Longitudinal data with T variables
Lowess smoothing
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
One N variable 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
Missing values found in D variables
Model fitted to subset of observations with complete D values

Summary information for training sample of size 888
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
      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/   #missing
    #cases    miss. 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
Number of cases used for training: 888
Number of split variables: 2
Number of cases excluded due to 0 weight or missing D: 0

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.000

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 10
```

```
Minimum node sample size: 44
Top-ranked variables and chi-squared values at root node
     1  0.1235E+02   hgc
     2  0.6915E+01   race

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1        9   1.262E+02   1.042E+01   9.660E+00   1.244E+02   1.005E+01
   2        7   1.262E+02   1.042E+01   9.660E+00   1.244E+02   1.005E+01
   3        5   1.243E+02   1.054E+01   9.934E+00   1.206E+02   1.029E+01
   4**      3   1.235E+02   1.051E+01   9.863E+00   1.205E+02   1.077E+01
   5++      2   1.237E+02   1.060E+01   1.006E+01   1.204E+02   1.102E+01
   6        1   1.244E+02   1.065E+01   1.011E+01   1.210E+02   1.171E+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 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

Following tree has minimum mean CV (*)

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    Total    Cases      Node    Split
      label    cases      fit      MSE     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
Second best split variable (based on curvature test) at root node is race

Regression tree for longitudinal data:
For categorical variable splits, 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
 R code is stored in wage.r
 Split and fit variable names are stored in wage.var
```

Figure 33 shows the tree and Figure 34 plots lowess-smoothed curves of mean wage in the two terminal nodes. The figure is produced by the following R code.

```
z <- read.table("wagedat.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,
```

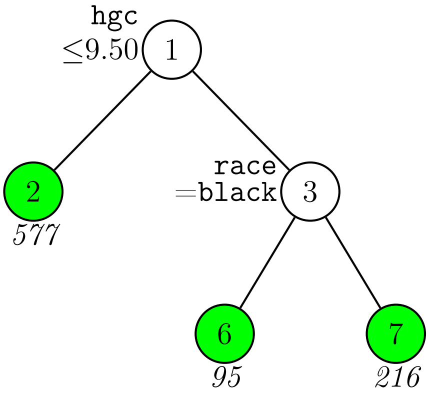

Figure 33: GUIDE v.38.0 0-SE regression tree for predicting longitudinal variables wage1, wage2, etc. Tree constructed with 888 observations. Maximum number of split levels is 10 and minimum node sample size is 44. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size *(in italics)* printed below nodes. Second best split variable at root node is race.

```
           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]
```

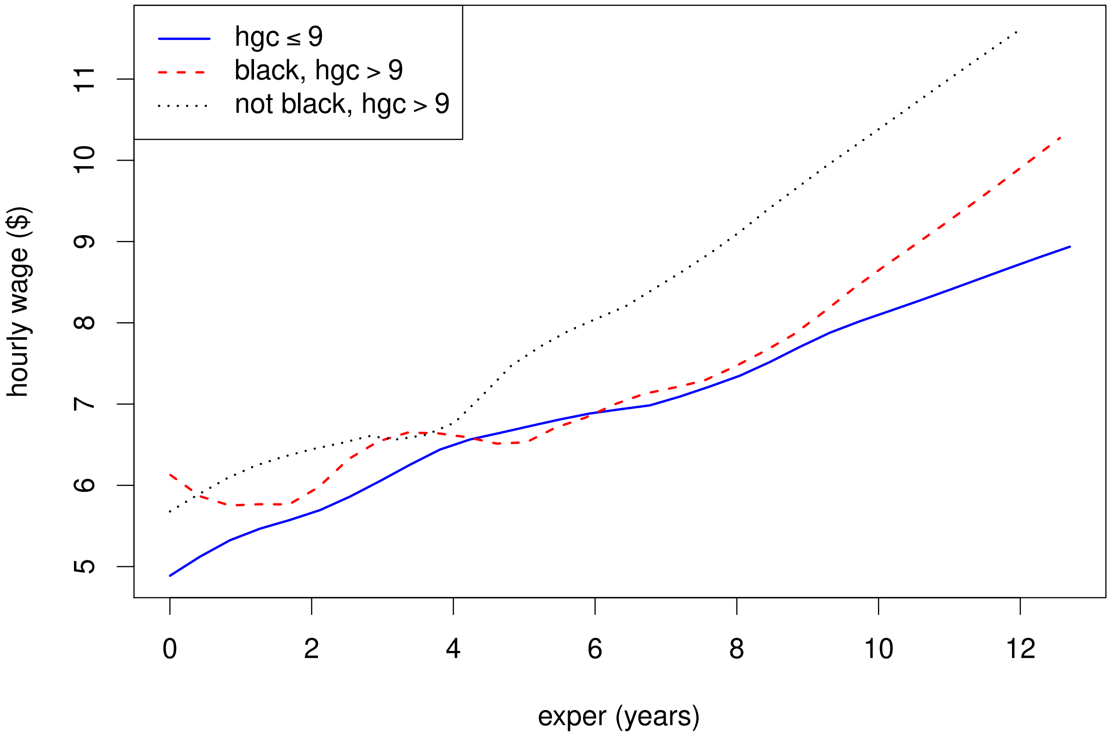

Figure 34: Lowess-smoothed mean wage curves in the terminal nodes of Figure 33.

```
    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])
}
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 fitted1
    2   0.10000E-02  0.12700E+02  0.48875E+01  0.51221E+01  0.53241E+01  0.54668E+01  0.55738E+01
    6   0.80000E-02  0.12558E+02  0.61270E+01  0.58648E+01  0.57522E+01  0.57674E+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
```

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.

```
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
```

15 Logistic regression

If the dependent variable Y takes values 0 and 1 and a preliminary estimate of $p = P(Y = 1)$ is available, GUIDE can construct a tree model such that a simple or multiple linear logistic regression model is fitted in each node. The preliminary estimate of p may be obtained by fitting a GUIDE forest or kernel discriminant model to the data. Missing values in the predictor variables used in the logistic models are imputed with node means. See Loh (2021) for more details.

We demonstrate the simple linear logistic feature by revisiting the NHTSA data introduced in Sec. 6. The data and description files are `withest.dat` and `withest.dsc`, where `withest.dat` is the same as `nhtsaclass.csv` except for an added last column containing the predicted values from GUIDE forest. This variable is denoted by the letter "E" or "e" in the description file `withest.dsc` (see Section 3.1). The "d" variable is `HIC2` which must take values 0 or 1.

15.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: logits.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: logits.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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 7
Choose complexity of model to use at each node:
Choose 1 for multiple regression (not recommended if there are missing values)
Choose 2 for simple polynomial in one N or F variable
1: multiple linear, 2: simple polynomial ([1:2], <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: withest.dsc
Reading data description file ...
Training sample file: withest.dat
Missing value code: NA
Records in data file start on line 2
D variable is HIC2
Reading data file ...
Number of records in data file: 3310
Length of longest entry in data file: 19
Checking for missing values ...
```

```
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 48 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: C variable RST5PT takes only 1 value
Warning: C variable RSTABT takes only 1 value
Warning: C variable RSTBSS takes only 1 value
Warning: C variable RSTCSR takes only 1 value
Warning: C variable RSTFSS takes only 1 value
Warning: C variable RSTISS takes only 1 value
Warning: C variable RSTOT takes only 1 value
Warning: C variable RSTSBK takes only 1 value
Warning: C variable RSTSHE takes only 1 value
Warning: C variable RSTVES takes only 1 value
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310         34       2891       57       31        0        5
    #P-var   #M-var   #B-var   #C-var   #I-var
         6        0        0       48        0
Number of cases used for training: 3276
Number of split variables: 84
Number of cases excluded due to 0 weight or missing D: 34
Proportion of ones in HIC2 variable:   8.4554334554334559E-002
Finished reading data file
Minimum number of D=0 and D=1 in each node:           9
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): logits.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: logits.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: logits.r
Input rank of top variable to split root node ([1:90], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < logits.in
```

15.2 Contents of `logits.out`

```
Binary logistic regression tree
Pruning by cross-validation
Data description file: withest.dsc
Training sample file: withest.dat
Missing value code: NA
Records in data file start on line 2
D variable is HIC2
Piecewise simple linear logistic model
Number of records in data file: 3310
Length of longest entry in data file: 19
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Warning: C variable RST5PT takes only 1 value
Warning: C variable RSTABT takes only 1 value
Warning: C variable RSTBSS takes only 1 value
Warning: C variable RSTCSR takes only 1 value
Warning: C variable RSTFSS takes only 1 value
Warning: C variable RSTISS takes only 1 value
Warning: C variable RSTOT takes only 1 value
Warning: C variable RSTSBK takes only 1 value
Warning: C variable RSTSHE takes only 1 value
Warning: C variable RSTVES takes only 1 value

Summary information for training sample of size 3276 (excluding observations with
non-positive weight or missing values in d, e, t, r or z 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,
e=estimated success probability
                                                 #Codes/
                                                 Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      2  BARRIG     c                                  3
      3  BARSHP     c                                 21
      4  BARANG     p    0.000        330.0          360        14
      7  OCCAGE     s    0.000        99.00                   1242
      8  OCCSEX     c                                  4
      :
```

```
     145  RSTUNK     c                             3
     146  RSTVES     c                             1
     147  HIC2       d    0.000        1.000
     149  estHIC2    e    0.000       0.7240

      Total  #cases w/   #missing
     #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       3310         34       2891       57       31        0        5
     #P-var   #M-var   #B-var   #C-var   #I-var
          6        0        0       48        0
 Number of cases used for training: 3276
 Number of split variables: 84
 Number of cases excluded due to 0 weight or missing D: 34
 Proportion of ones in HIC2 variable: 0.084554

 Missing values imputed with node means for fitting regression models in nodes
 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: 0.2500

 Nodewise interaction tests on all variables
 Fraction of cases used for splitting each node: 1.0000
 Maximum number of split levels: 13
 Minimum node sample size: 65
 Minimum number of D=0 and D=1 in each node: 9
 Top-ranked variables and chi-squared values at root node
      1  0.5235E+03   COLMEC
      2  0.4301E+03   BMPENG
      3  0.2659E+03   BARSHP
      4  0.2285E+03   IMPANG
      :
     64  0.7170E+00   VEHSPD
     65  0.4370E+00   CURBWT
     66  0.1921E+00   DUMSIZ

 Size and CV Loss and SE of subtrees:
  Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
    1*      13   4.353E-01   2.182E-02   1.387E-02   4.351E-01   1.390E-02
    2++     12   4.371E-01   2.187E-02   1.386E-02   4.375E-01   1.445E-02
    3**     10   4.391E-01   2.212E-02   1.291E-02   4.396E-01   1.036E-02
    4        9   4.413E-01   2.232E-02   1.297E-02   4.396E-01   1.069E-02
    5        8   4.506E-01   2.314E-02   1.301E-02   4.502E-01   1.163E-02
    6        6   4.506E-01   2.314E-02   1.301E-02   4.502E-01   1.163E-02
    7        5   4.541E-01   2.323E-02   1.007E-02   4.536E-01   1.180E-02
    8        4   4.592E-01   2.081E-02   8.197E-03   4.557E-01   8.981E-03
    9        3   4.581E-01   1.995E-02   9.476E-03   4.492E-01   1.153E-02
  10        2   4.548E-01   1.941E-02   7.804E-03   4.492E-01   1.098E-02
  11        1   4.548E-01   1.941E-02   7.804E-03   4.492E-01   1.098E-02

 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 13 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 HIC2 in the node
 Cases fit give the number of cases used to fit node
 Node deviance is residual deviance divided by residual degrees of freedom
        Node    Total    Cases Matrix    Node       Node    Split          Other
        label   cases      fit  rank    D-mean    deviance  variable       variables
           1     3276     3276     2  8.455E-02  4.546E-01  COLMEC   -YEAR
           2      662      662     2  3.051E-01  1.211E+00  BX2   -BX17
           4      305      305     2  2.689E-01  1.101E+00  BX5   +BX5
           8      229      229     2  2.271E-01  1.063E+00  VEHTWT   +BX18
          16T      89       89     2  1.236E-01  6.612E-01 -   -WHLBAS
          17      140      140     2  2.929E-01  1.134E+00  VEHWID   -VEHWID
          34T      70       70     2  4.429E-01  1.330E+00 -   -ENGDSP
          35T      70       70     2  1.429E-01  7.540E-01 -   -YEAR
           9T      76       76     2  3.947E-01  1.169E+00 -   +BX5
           5      357      357     2  3.361E-01  1.192E+00  TRANSM   +VEHSPD
          10T     100      100     2  2.100E-01  1.004E+00 -   -YEAR
          11T     257      257     2  3.852E-01  1.221E+00 VEHSPD   +VEHSPD
           3     2614     2614     2  2.869E-02  2.344E-01  BARSHP   -YEAR
           6     1581     1581     2  4.175E-02  2.853E-01  IMPANG   -YEAR
          12T      67       67     2  2.388E-01  1.033E+00 -   -YEAR
          13     1514     1514     2  3.303E-02  2.160E-01  BARSHP   -YEAR
          26T    1150     1150     2  3.565E-02  2.068E-01 BODY   -YEAR
          27T     364      364     2  2.473E-02  1.992E-01 -   -YEAR
           7T    1033     1033     2  8.712E-03  9.261E-02 -   -YEAR

 Number of terminal nodes of final tree: 10
 Total number of nodes of final tree: 19
 Second best split variable (based on curvature test) at root node is BMPENG

 Regression tree:
 For categorical variable splits, values not in training data go to the right
```

```
Node 1: COLMEC = "BWU", "EMB", "EXA", "NON", "OTH"
  Node 2: BX2 <= 3496.5000 or NA
    Node 4: BX5 <= 82.500000 or NA
      Node 8: VEHTWT <= 1368.5000
        Node 16: HIC2 proportion of 1s = 0.12359551
      Node 8: VEHTWT > 1368.5000 or NA
        Node 17: VEHWID <= 1847.0000
          Node 34: HIC2 proportion of 1s = 0.44285714
        Node 17: VEHWID > 1847.0000 or NA
          Node 35: HIC2 proportion of 1s = 0.14285714
    Node 4: BX5 > 82.500000
      Node 9: HIC2 proportion of 1s = 0.39473684
  Node 2: BX2 > 3496.5000
    Node 5: TRANSM = "A4", "AF"
      Node 10: HIC2 proportion of 1s = 0.21000000
    Node 5: TRANSM /= "A4", "AF"
      Node 11: HIC2 proportion of 1s = 0.38521401
Node 1: COLMEC /= "BWU", "EMB", "EXA", "NON", "OTH"
  Node 3: BARSHP = "LCB", "POL", "US2", "US3"
    Node 6: IMPANG in (284, 286)
      Node 12: HIC2 proportion of 1s = 0.23880597
    Node 6: IMPANG not in (284, 286) or NA
      Node 13: BARSHP = "LCB"
        Node 26: HIC2 proportion of 1s = 0.35652174E-1
      Node 13: BARSHP /= "LCB"
        Node 27: HIC2 proportion of 1s = 0.24725275E-1
  Node 3: BARSHP /= "LCB", "POL", "US2", "US3"
    Node 7: HIC2 proportion of 1s = 0.87124879E-2

***************************************************************

Predictor means below are means of cases with no missing values.
Regression coefficients are computed from the complete cases.

WARNING: p-values below not adjusted for split search. For a bootstrap solution see:

1. Loh et al. (2016), "Identification of subgroups with differential treatment effects
for longitudinal and multiresponse variables", Statistics in Medicine, v.35, 4837-4855.

2. Loh et al. (2019), "Subgroups from regression trees with adjustment for prognostic
effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
in "Design and Analysis of Subgroups with Biopharmaceutical Applications", Springer, pp.147-165.
```

```
Node 1: Intermediate node
 A case goes into Node 2 if COLMEC = "BWU", "EMB", "EXA", "NON", "OTH"
 COLMEC mode = "UNK"
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      258.0        17.26       0.6661E-15
 YEAR        -0.1306       -17.38        0.000        1972.        2000.        2017.
 Proportion of ones in variable HIC2 = 0.845543E-1
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if BX2 <= 3496.5000 or NA
 BX2 mean = 3695.6483
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if BX5 <= 82.500000 or NA
 BX5 mean = 1890.6456
 ----------------------------
Node 8: Intermediate node
 A case goes into Node 16 if VEHTWT <= 1368.5000
 VEHTWT mean = 1572.1150
 ----------------------------
Node 16: Terminal node
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      13.04        2.251       0.2689E-01
 WHLBAS      -0.6388E-02   -2.566       0.1199E-01    1656.        2391.        2944.
 Proportion of ones in variable HIC2 = 0.123596
 ----------------------------
Node 17: Intermediate node
 A case goes into Node 34 if VEHWID <= 1847.0000
 VEHWID mean = 1821.0809
 ----------------------------
Node 34: Terminal node
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      1.500        1.834       0.7103E-01
 ENGDSP      -0.5767       -2.164       0.3397E-01    1.300        3.066        6.600
 Proportion of ones in variable HIC2 = 0.442857
 ----------------------------
Node 35: Terminal node
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      905.5        2.143       0.3569E-01
 YEAR        -0.4585       -2.147       0.3538E-01    1975.        1980.        2016.
 Proportion of ones in variable HIC2 = 0.142857
 ----------------------------
Node 9: Terminal node
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -2.229       -3.447       0.9370E-03
 BX5          0.8431E-03    3.304       0.1472E-02    85.00        1962.        4870.
 Proportion of ones in variable HIC2 = 0.394737
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if TRANSM = "A4", "AF"
 TRANSM mode = "MF"
 ----------------------------
Node 10: Terminal node
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      181.6        1.715       0.8959E-01
 YEAR        -0.9200E-01   -1.726       0.8747E-01    1979.        1989.        2012.
 Proportion of ones in variable HIC2 = 0.210000
 ----------------------------
Node 11: Terminal node
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -57.50       -2.901       0.4041E-02
 VEHSPD        1.015        2.886       0.4237E-02    39.60        55.49        57.10
 Proportion of ones in variable HIC2 = 0.385214
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if BARSHP = "LCB", "POL", "US2", "US3"
 BARSHP mode = "LCB"
 ----------------------------
Node 6: Intermediate node
 A case goes into Node 12 if IMPANG in [284, 286]
 IMPANG mean = 67.425680
 ----------------------------
Node 12: Terminal node
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      487.9        2.343       0.2222E-01
 YEAR        -0.2439       -2.348       0.2195E-01    1999.        2005.        2012.
 Proportion of ones in variable HIC2 = 0.238806
 ----------------------------
Node 13: Intermediate node
 A case goes into Node 26 if BARSHP = "LCB"
 BARSHP mode = "LCB"
 ----------------------------
Node 26: Terminal node
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      614.6        7.711       0.2742E-13
 YEAR        -0.3093       -7.737       0.2276E-13    1982.        2004.        2017.
 Proportion of ones in variable HIC2 = 0.356522E-1
 ----------------------------
Node 27: Terminal node
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      308.0        3.707       0.2422E-03
 YEAR        -0.1552       -3.745       0.2097E-03    1986.        2011.        2017.
 Proportion of ones in variable HIC2 = 0.247253E-1
 ----------------------------
Node 7: Terminal node
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      278.4        2.744       0.6167E-02
 YEAR        -0.1418       -2.787       0.5417E-02    1974.         2000.        2017.
 Proportion of ones in variable HIC2 = 0.871249E-2
 ----------------------------
 Observed and fitted values are stored in logits.fit
 LaTeX code for tree is in logits.tex
 R code is stored in logits.r
```

Figure 35 shows the logistic regression tree and Figure 36 shows the fitted logistic curves in the terminal nodes (see Table 13 for the meanings of the variables). The R code for the plots is given in Figure 37.

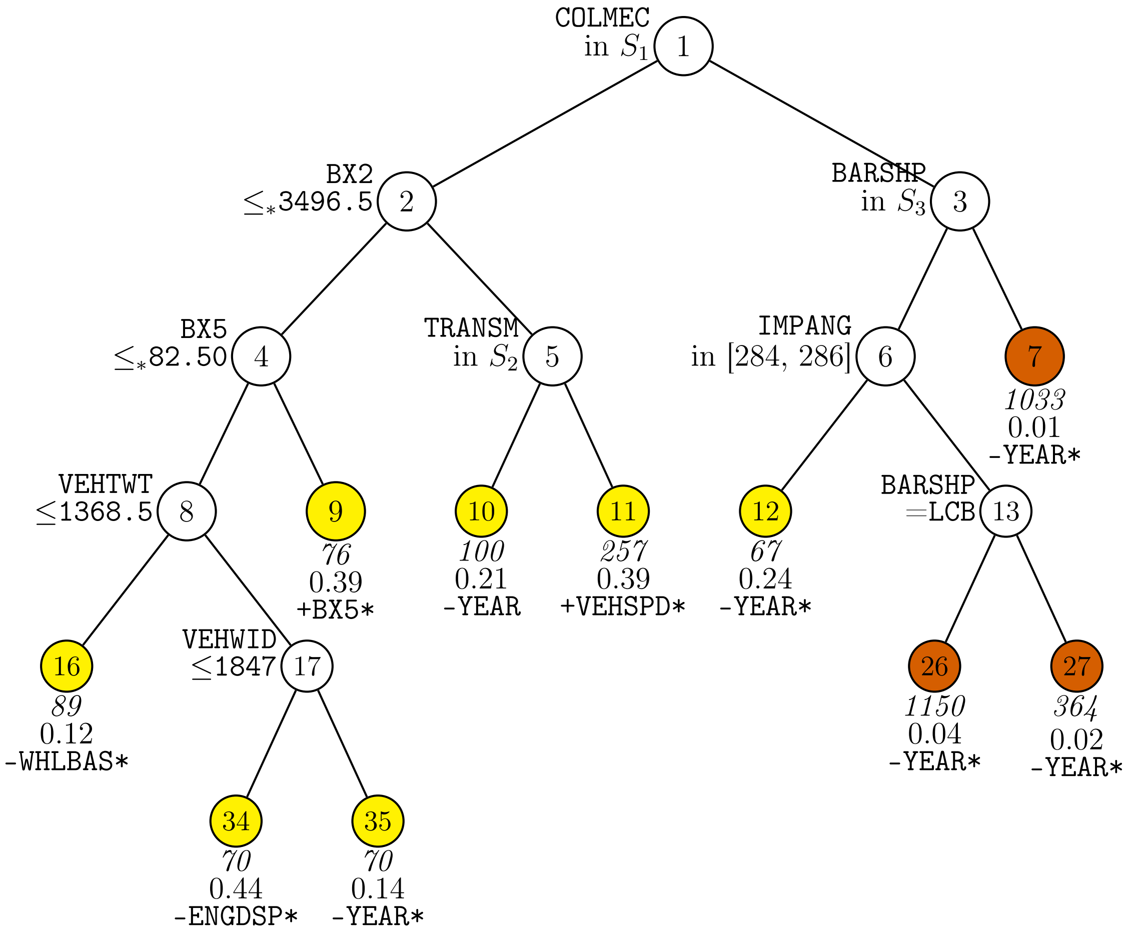

Figure 35: GUIDE v.38.0 0.25-SE piecewise simple linear logistic regression tree for predicting HIC2. Tree constructed with 3276 observations (excluding observations with non-positive weight or missing values in D, T, R or Z variables). Maximum number of split levels is 13 and minimum node sample size is 65. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. Set S_1 = {BWU, EMB, EXA, NON, OTH}. Set S_2 = {A4, AF}. Set S_3 = {LCB, POL, US2, US3}. Sample size (in *italics*), proportion of 1s in HIC2, and signed name of regressor variable printed below nodes. Terminal nodes with proportions of 1s above and below value of 0.08 at root node are colored yellow and vermillion respectively. Asterisk appended to regressor name indicates its slope is significant at the 0.05 level (unadjusted for multiplicity and model fitting). Second best split variable at root node is BMPENG.

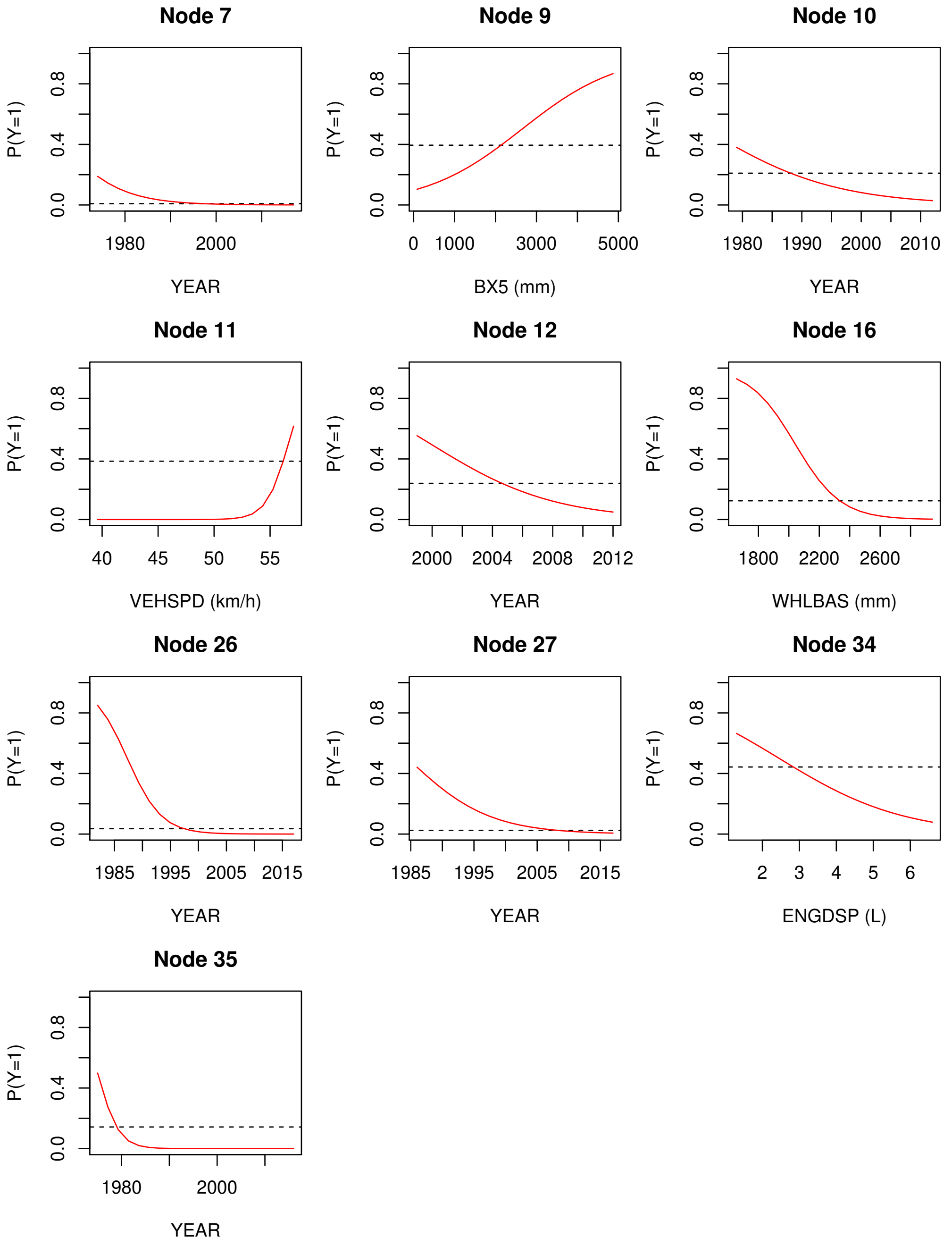

Figure 36: Estimated logistic regression curves in terminal nodes of tree in Figure 35. Horizontal dashed line marks proportion of head injury in node.

```
z1 <- read.csv("nhtsadata.csv",header=TRUE)
z2 <- read.table("logits.fit",header=TRUE)
par(mfrow=c(4,3),mar=c(4,4,3,1),cex=0.9)
nvarid <- 1:dim(z1)[2]
nodes <- unique(sort(z2$node))
xnames <- c("YEAR","BX5","YEAR","VEHSPD","YEAR","WHLBAS","YEAR","YEAR",
            "ENGDSP","YEAR")
xlabs <- c("YEAR","BX5␣(mm)","YEAR","VEHSPD␣(km/h)","YEAR","WHLBAS␣(mm)",
           "YEAR","YEAR","ENGDSP␣(L)","YEAR")
titles.txt <- paste("Node",nodes)
i <- 0
for(node in nodes){
    i <- i+1
    tmp <- names(z1) %in% xnames[i]
    xid <- nvarid[tmp]
    gp <- z2$node == node & z2$train == "y" & !is.na(z1[,xid])
    x <- z1[,xid][gp]
    y <- z1$HIC2[gp]
    plot(y ~ x,xlab=xlabs[i],ylab="P(Y=1)",type="n")
    title(main=titles.txt[i])
    y1 <- z1$HIC2[z2$node == node & z2$train == "y"]
    abline(h=mean(y1),lty=2)
    model <- glm(y ~ x, family=’binomial’)
    xgrid <- seq(from=min(x),to=max(x),length.out=20)
    fitted <- model$coef[1]+model$coef[2]*xgrid
    fitted <- 1/(1+exp(-fitted))
    lines(fitted ~ xgrid,col="red")
}
```

Figure 37: R code for Figure 36

16 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.

16.1 Classification: RHC data

We show here how to obtain the importance scores for predicting `swang1`, the variable that takes values `RHC` and `NoRHC`; see Section 4.

16.1.1 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
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: rhcdsc1.txt
Reading data description file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
22 N variables changed to S
D variable is swang1
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 2
```

```
Assigning integer codes to values of 30 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Class  #Cases     Proportion
NoRHC    3551     0.61918047
RHC      2184     0.38081953
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157       11        0        0       22
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       30        0
Number of cases used for training: 5735
Number of split variables: 52
Number of cases excluded due to 0 weight or missing D: 0
Finished reading data file
Choose 1 for estimated priors, 2 for equal priors, 3 to input 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 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): imp.tex
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 rank of top variable to split root node ([1:50], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < imp.in
```

16.1.2 Contents of `imp.out`

The most interesting part of the output file is at the end, as shown below. The variables, sorted according to their importance scores, are divided into three groups. Those with scores above and below 1.0 are considered "important" and "unimportant", respectively. The division is such that if all the variables are independent of the response variable, the probability is 0.05 that any is found important. The group of important variables is further divided between the “highly important” (99% confidence) and the “likely important” (95% confidence). If all the variables are independent of the response variable, the probability is 0.01 that any is found to be highly important.

```
 Scaled importance scores of predictor variables
      Score     Rank  Variable
  2.367E+01     1.00  aps1
  2.344E+01     2.00  cat1
  2.144E+01     3.00  crea1
  2.104E+01     4.00  pafi1
  1.930E+01     5.00  meanbp1
  1.273E+01     6.00  neuro
  1.137E+01     7.00  alb1
  1.068E+01     8.00  cat2
  1.067E+01     9.00  card
  1.019E+01    10.00  hema1
  9.686E+00    11.00  wtkilo1
  8.175E+00    12.00  seps
  8.055E+00    13.00  adld3p
  6.756E+00    14.00  dnr1
  6.514E+00    15.00  bili1
  6.198E+00    16.00  resp
  5.644E+00    17.00  paco21
  4.100E+00    18.00  transhx
  4.059E+00    19.00  hrt1
  4.022E+00    20.00  resp1
  3.989E+00    21.00  chrpulhx
  3.717E+00    22.00  ph1
  3.449E+00    23.00  ninsclas
  3.383E+00    24.00  dementhx
  2.413E+00    25.00  das2d3pc
  2.247E+00    26.00  psychhx
  2.168E+00    27.00  renal
  2.043E+00    28.00  gastr
  1.867E+00    29.00  cardiohx
  1.759E+00    30.00  income
  1.462E+00    31.00  urin1
  1.320E+00    32.00  trauma
 --------- variables above this line are highly important -----
  1.228E+00    33.00  age
  1.196E+00    34.00  sex
  1.181E+00    35.00  sod1
  1.172E+00    36.00  edu
  1.057E+00    37.00  wblc1
 --------- variables below this line are unimportant ----------
  9.432E-01    38.00  immunhx
  9.272E-01    39.00  malighx
  9.202E-01    40.00  ca
  8.261E-01    41.00  scoma1
  7.870E-01    42.00  amihx
  6.956E-01    43.00  chfhx
  6.486E-01    44.00  gibledhx
  4.369E-01    45.00  pot1
  4.157E-01    46.00  ortho
  3.943E-01    47.00  renalhx
  3.642E-01    48.00  hema
  3.518E-01    49.00  liverhx
  3.200E-01    50.00  meta
  2.827E-01    51.00  temp1
  1.251E-01    52.00  race

 Variables with scores above 1.27 are highly important
 Variables with scores between 1.0 and 1.27 are likely important
 Variables with scores below 1.0 are unimportant
 No. highly important, likely important, and unimportant split variables: 32, 5, 15
 LaTeX code for tree is in imp.tex
 Importance scores are stored in imp.scr
```

The scores are also printed in the file `imp.scr`, whose contents follow. The file has three columns, labeled `Type`, `Score`, and `Variable`. The first column entries are "`H`" (for high importance, 99% confidence), "`L`" (for low importance, 95% confidence), and "U" (for unimportant).

```
Type      Score       Variable
 H      2.367E+01     aps1
 H      2.344E+01     cat1
 H      2.144E+01     crea1
 H      2.104E+01     pafi1
 H      1.930E+01     meanbp1
 H      1.273E+01     neuro
 H      1.137E+01     alb1
 H      1.068E+01     cat2
 H      1.067E+01     card
 H      1.019E+01     hema1
 H      9.686E+00     wtkilo1
 H      8.175E+00     seps
 H      8.055E+00     adld3p
 H      6.756E+00     dnr1
 H      6.514E+00     bili1
```

```
  H      6.198E+00     resp
  H      5.644E+00     paco21
  H      4.100E+00     transhx
  H      4.059E+00     hrt1
  H      4.022E+00     resp1
  H      3.989E+00     chrpulhx
  H      3.717E+00     ph1
  H      3.449E+00     ninsclas
  H      3.383E+00     dementhx
  H      2.413E+00     das2d3pc
  H      2.247E+00     psychhx
  H      2.168E+00     renal
  H      2.043E+00     gastr
  H      1.867E+00     cardiohx
  H      1.759E+00     income
  H      1.462E+00     urin1
  H      1.320E+00     trauma
  L      1.228E+00     age
  L      1.196E+00     sex
  L      1.181E+00     sod1
  L      1.172E+00     edu
  L      1.057E+00     wblc1
  U      9.432E-01     immunhx
  U      9.272E-01     malighx
  U      9.202E-01     ca
  U      8.261E-01     scoma1
  U      7.870E-01     amihx
  U      6.956E-01     chfhx
  U      6.486E-01     gibledhx
  U      4.369E-01     pot1
  U      4.157E-01     ortho
  U      3.943E-01     renalhx
  U      3.642E-01     hema
  U      3.518E-01     liverhx
  U      3.200E-01     meta
  U      2.827E-01     temp1
  U      1.251E-01     race
```

Figure 38 shows a barplot of the scores. It is made by the following R code.

```
leg.col <- c("orange","yellow")
leg.txt <- c("highly important","likely important")
par(las=1,mar=c(5,12,4,2))
x <- read.table("imp.scr",header=TRUE)
score <- x$Score
```

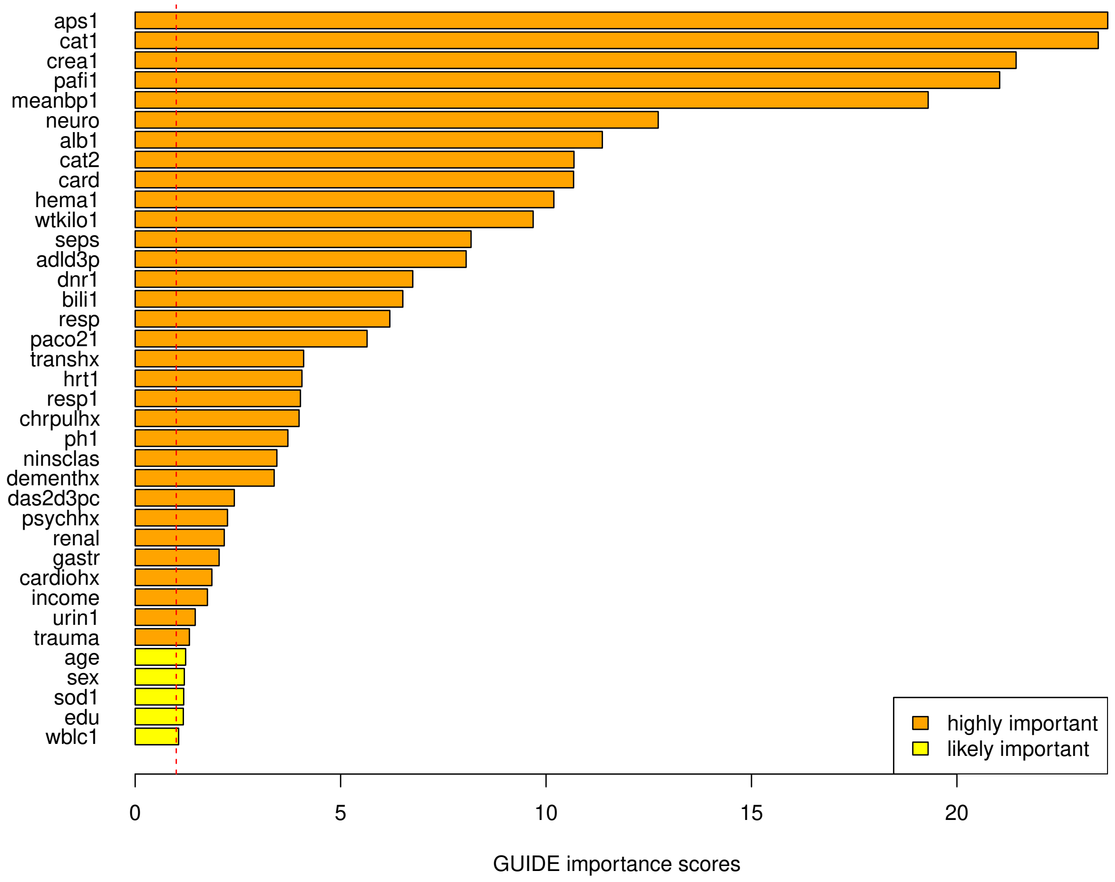

Figure 38: Scores of important variables for predicting `swang1`

```
vars <- x$Variable
type <- x$Type
barcol <- rep("orange",length(vars))
barcol[type == "L"] <- "yellow"
barcol[type == "U"] <- "cyan"
n <- sum(x$Type != "U")
barplot(rev(score[1:n]),names.arg=rev(vars[1:n]),col=rev(barcol[1:n]),horiz=TRUE,xlab="GUIDE importa
abline(v=1,col="red",lty=2)
legend("bottomright",legend=leg.txt,fill=leg.col)
```

Figure 39 shows the classification tree from `imp.tex` that produced the scores. It is an unpruned tree with four levels of splits.

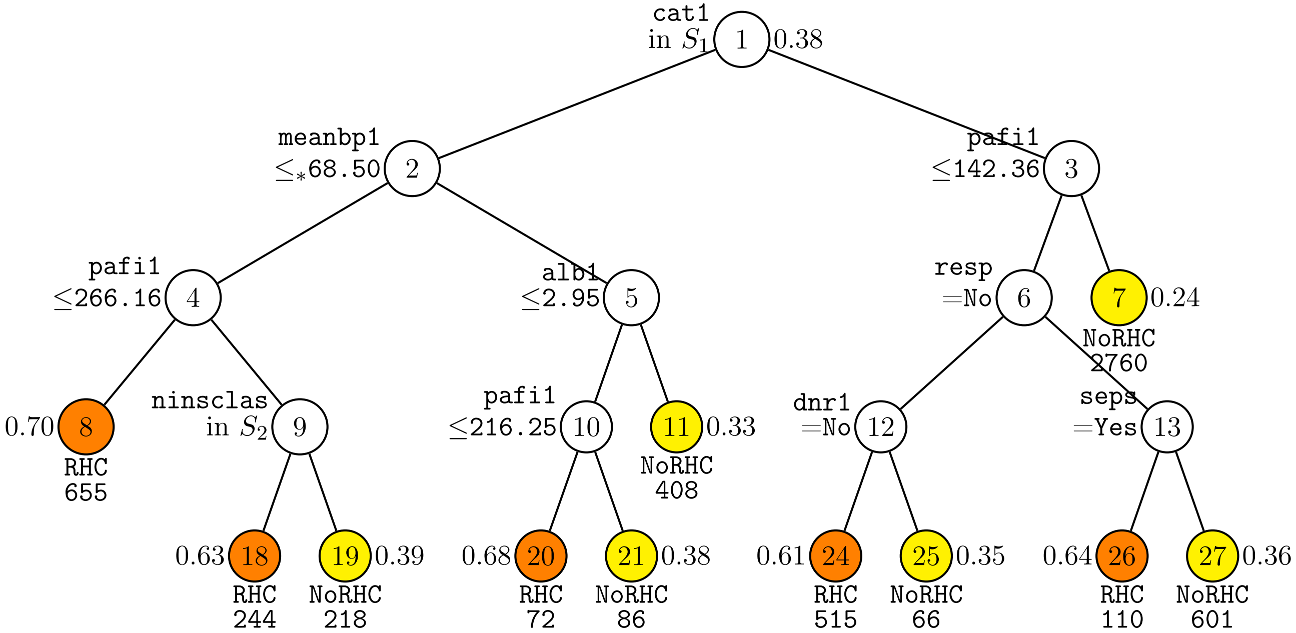

Figure 39: GUIDE v.38.0 importance scoring classification tree for predicting swang1 using estimated priors and unit misclassification costs. Tree constructed with 5735 observations. Maximum number of split levels is 4 and minimum node sample size is 57. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. Set S_1 = {CHF, MOSF w/Sepsis}. Set S_2 = {No insurance, Private, Private & Medicare}. Predicted classes and sample sizes printed below terminal nodes; class sample proportion for swang1 = RHC beside nodes. Second best split variable at root node is aps1.

16.2 Censored response with R variable

Following is the corresponding scoring procedure for a censored response with a treatment (R) variable (swang1). The R variable is not given a score because it acts as a linear predictor in the nodes of the tree.

16.2.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: imp_surv.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_surv.out
 Input 1 for classification, 2 for regression, 3 for propensity score grouping
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
 Input choice ([1:6], <cr>=1): 4
 Input 1 for proportional hazards, 2 for restricted mean event time ([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: rhcdsc3.tx
 Description file rhcdsc3.tx does not exist
 Input name of data description file (max 100 characters);
 enclose with matching quotes if it has spaces: rhcdsc3.txt
rhcdsc3.txt
 Reading data description file ...
 Training sample file: rhcdata.txt
 Missing value code: NA
 Records in data file start on line 2
 R variable present
 20 N variables changed to S
 Warning: model changed to linear in treatment
 D variable is death
 Reading data file ...
 Number of records in data file: 5735
 Length of longest entry in data file: 19
 Checking for missing values ...
 Finished checking
 Missing values found among categorical variables
 Separate categories will be created for missing categorical variables
 Missing values found among non-categorical variables
```

```
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Smallest uncensored survtime: 2.0000
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
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
Input 1 if randomized trial, 2 if observational study: ([1:2], <cr>=1): 2
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):

Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime by swang1
"NoRHC"      1867.0000    1243.0000
  "RHC"      1943.0000    1351.0000
Proportion of training sample for each level of swang1
"NoRHC"    0.6192
  "RHC"    0.3808
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
Survival time variable in column: 64
Event indicator variable in column: 9
Proportion uncensored among nonmissing T and D variables: 0.649
Number of cases used for training: 5735
Number of split variables: 53
Number of dummy variables created: 1
Number of cases excluded due to 0 weight or missing D, T or R: 0
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): imp_surv.tex
```

```
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_surv.scr
Input rank of top variable to split root node ([1:52], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < imp_surv.in
```

16.2.2 Partial contents of imp_surv.out

The output shows that there is only one important variable.

```
Scaled importance scores of predictor variables
 (F, I and R variables are excluded)
     Score     Rank  Variable
 1.061E+00     1.00  dnr1
--------- variables below this line are unimportant ----------
 9.386E-01     2.00  ph1
 8.397E-01     3.00  paco21
 8.293E-01     4.00  chrpulhx
 7.984E-01     5.00  resp1
 4.899E-01     6.00  liverhx
 4.500E-01     7.00  gastr
 4.444E-01     8.00  pot1
 4.229E-01     9.00  cat2
 3.911E-01    10.00  gibledhx
 3.909E-01    11.00  age
 3.645E-01    12.00  pafi1
 3.561E-01    13.00  aps1
 3.192E-01    14.00  amihx
 3.180E-01    15.00  malighx
 3.046E-01    16.00  hrt1
 2.881E-01    17.00  surv2md1
 2.719E-01    18.00  ninsclas
 2.540E-01    19.00  edu
 2.498E-01    20.00  das2d3pc
 2.394E-01    21.00  meanbp1
 2.259E-01    22.00  income
 2.022E-01    23.00  scoma1
 1.832E-01    24.00  ortho
 1.753E-01    25.00  crea1
 1.721E-01    26.00  temp1
 1.674E-01    27.00  hema1
 1.618E-01    28.00  ca
```

```
  1.587E-01    29.00  hema
  1.471E-01    30.00  psychhx
  1.470E-01    31.00  wtkilo1
  1.438E-01    32.00  trauma
  1.433E-01    33.00  renalhx
  1.412E-01    34.00  sex
  1.407E-01    35.00  neuro
  1.325E-01    36.00  urin1
  1.300E-01    37.00  alb1
  1.269E-01    38.00  chfhx
  1.254E-01    39.00  wblc1
  1.004E-01    40.00  dementhx
  9.603E-02    41.00  adld3p
  9.302E-02    42.00  race
  8.556E-02    43.00  seps
  8.360E-02    44.00  sod1
  8.273E-02    45.00  cat1
  7.711E-02    46.00  cardiohx
  7.603E-02    47.00  resp
  5.051E-02    48.00  card
  4.784E-02    49.00  renal
  4.577E-02    50.00  transhx
  4.444E-02    51.00  meta
  4.180E-02    52.00  bili1
  3.832E-02    53.00  immunhx

 Variables with scores above 1.46 are highly important
 Variables with scores between 1.0 and 1.46 are likely important
 Variables with scores below 1.0 are unimportant
 No. highly important, likely important, and unimportant split variables: 0, 1, 52
 LaTeX code for tree is in imp_surv.tex
 Importance scores are stored in imp_surv.scr
```

17 Propensity scores: RHC data

Propensity score matching is often used in causal inference to estimate average treatment effects. Given a treatment variable Z taking values 0 (no treatment) and 1 (treatment), the propensity score for a subject with covariate $X = x$ is $\pi(x) = P(Z = 1 \mid X = x)$. If n denotes the sample size and Y_i the response of the ith subject, the average treatment effect may be estimated by the *Horvitz-Thompson* *estimate (HT)*

$$n^{-1}\sum_{i=1}^{n}\left\{\frac{Z_iY_i}{\hat{\pi}(X_i)}-\frac{(1-Z_i)Y_i}{1-\hat{\pi}(X_i)}\right\}$$

or the *Hájek inverse probability estimate (IPW)*

$$\frac{\sum_i Z_iY_i/\hat{\pi}(X_i)}{\sum_i Z_i/\hat{\pi}(X_i)}-\frac{\sum_i(1-Z_i)Y_i/(1-\hat{\pi}(X_i))}{\sum_i(1-Z_i)/(1-\hat{\pi}(X_i))}$$

where $\hat{\pi}(x)$ is an estimate of $\pi(x)$. Clearly, $\hat{\pi}(x)$ cannot be 0 or 1.

The propensity scores are traditionally estimated by logistic regression, but this approach encounters difficulties if there are missing values in the covariates or if the number of covariates is large. Recently, random forest has been used, but it too has difficulties with missing values. Even when there are no missing values, the propensity score estimates from logistic regression and random forest are not easy to interpret.

A classification tree for predicting Z is much more interpretable than a forest, but one or more terminal nodes may be pure (i.e., all $Z_i = 0$ or all $Z_i = 1$), causing $\hat{\pi}(x_i)$, being the proportion of $Z = 1$ in the nodes, to be 0 or 1 there. To rectify this, GUIDE has a "propensity score" option that disallows such splits. Specifically, it only allows splits that yield in each subnode at least m observations each of $Z = 0$ and $Z = 1$. The value of m is a positive integer that may be specified by the user. If a GUIDE piecewise-constant model is used to estimate the propensity scores, the HT and IPW estimates are identical and reduce to the *node sample size weighted estimate*

$$n^{-1}\sum_t n_t\hat{\beta}_t$$

where the sum is over the terminal nodes and n_t and $\hat{\beta}_t$ are the node sample size and estimated treatment effect in node t.

We demonstrate the propensity score feature with the RHC data. Doctors believe that direct measurement of cardiac function by right heart catheterization for some critically ill patients yields better outcomes. The benefit of RHC has not been demonstrated in a randomized clinical trial due to ethical concerns. In observational studies, the relative risk of death was found to be higher in the elderly and in patients with acute myocardial infarction who received RHC. In such studies, the decision to use RHC is at the discretion of the physician. Therefore treatment assignment is confounded with patient factors that are also related to outcomes, e.g., patients with low blood pressure are more likely to get RHC, and such patients are also more likely to die. The data consist of observations on more than 60 variables for 5735 patients from 5 medical centers over 5 years (Connors et al., 1996). The treatment variable is swang1 (RHC or NoRHC), and the response variables are dth30 (1=death within 30 days, 0=survived more than 30 days) and death (1=eventual death, 0=censored). The data and description files are rhcdata.txt and rhcdsc4.txt. In the latter, the variable swang1 is designated as r, dth30 as d, and death as x.

17.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: prop30.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: prop30.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
Input your choice ([1:3], <cr>=1): 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: rhcdsc4.txt
Reading data description file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
32 N variables changed to S
Warning: model changed to linear in treatment
D variable is dth30
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 2
Assigning integer codes to values of 18 categorical variables
Finished assigning codes to 10 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
```

```
Data checks complete
Creating missing value indicators ...
Rereading data ...
Treatment      #Cases    Proportion
NoRHC            3551    0.61918047
RHC              2184    0.38081953
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        9        0        0       35
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       18        0
Number of cases used for training: 5735
Number of split variables: 53
Number of cases excluded due to 0 weight or missing D: 0
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): prop30.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: prop30.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: prop30.r
Input rank of top variable to split root node ([1:50], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < prop30.in
```

17.2 Contents of prop30.out

```
Propensity score grouping and estimation of causal effects
Pruning by cross-validation
Data description file: rhcdsc4.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
35 N variables changed to S
Warning: model changed to linear in treatment
D variable is dth30
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 2
Treatment      #Cases    Proportion
NoRHC            3551    0.61918047
```

```
RHC             2184    0.38081953

Summary information for training sample of size 5735
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
      2  cat1       c                                 9
      3  cat2       c                                 6     4535
      4  ca         c                                 3
      :
     28  dth30      d    0.000        1.000
     29  aps1       s    3.000        147.0
      :
     44  ph1        s    6.579        7.770
     45  swang1     r                                 2
     46  wtkilo1    s    19.50        244.0                   515
      :
     61  race       c                                 3
     62  income     c                                 4

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        9        0        0       35
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       18        0
Number of cases used for training: 5735
Number of split variables: 53
Number of cases excluded due to 0 weight or missing D: 0

Missing values imputed with node means for fitting regression models in nodes
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Simple node models
Equal priors
Unit misclassification costs
Univariate split highest priority
Interaction splits 2nd priority; no linear splits
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 15
Minimum node sample size: 6
```

```
Top-ranked variables and chi-squared values at root node
     1  0.3346E+03   cat1
     2  0.2728E+03   aps1
     3  0.2430E+03   crea1
     :
    52  0.1052E+01   meta
    53  0.6357E+00   race

Size and CV mean cost and SE of subtrees:
 Tree    #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   1       354   3.581E-01   6.817E-03   6.202E-03   3.609E-01   5.354E-03
   2       353   3.581E-01   6.817E-03   6.202E-03   3.609E-01   5.354E-03
 208        18   3.278E-01   6.421E-03   4.257E-03   3.277E-01   6.447E-03
 209**      16   3.255E-01   6.349E-03   5.516E-03   3.205E-01   9.186E-03
 210        14   3.287E-01   6.301E-03   5.926E-03   3.290E-01   9.957E-03
 211        12   3.285E-01   6.339E-03   5.849E-03   3.268E-01   8.241E-03
 212         8   3.330E-01   6.355E-03   7.153E-03   3.315E-01   8.781E-03
 213         6   3.360E-01   6.287E-03   6.883E-03   3.325E-01   9.229E-03
 214         5   3.527E-01   6.506E-03   7.212E-03   3.511E-01   5.489E-03
 215         4   3.690E-01   6.337E-03   7.280E-03   3.705E-01   9.859E-03
 216         2   4.131E-01   5.710E-03   3.745E-03   4.112E-01   3.751E-03
 217         1   5.000E-01   8.419E-03   2.585E-16   5.000E-01   2.764E-16

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 16 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     Total     Train     Predicted        Node    Split          Interacting
      label     cases     cases     class            cost    variables      variable
          1      5735      5735     NoRHC          5.000E-01 cat1
          2      4572      4572     RHC            4.469E-01 pafi1
          4      2218      2218     RHC            3.640E-01 crea1
          8       823       823     RHC            4.738E-01 pafi1
         16T      370       370     RHC            3.757E-01 resp
         17       453       453     NoRHC          4.385E-01 trauma
         34T       14        14     RHC            9.298E-02 -
         35       439       439     NoRHC          4.193E-01 card
         70T      107       107     RHC            4.213E-01 crea1
         71T     332      332    NoRHC          3.624E-01 bili1 :aps1
          9     1395     1395    RHC            3.044E-01 adld3p
         18T    1144     1144    RHC            2.608E-01 wtkilo1
         19      251      251    NoRHC          4.675E-01 resp1
         38T     114      114    RHC            3.483E-01 resp1
         39T     137      137    NoRHC          2.852E-01 gastr
          5     2354     2354    NoRHC          4.682E-01 cat1
         10     1076     1076    RHC            4.030E-01 meanbp1
         20T     798      798    RHC            3.358E-01 bili1
         21T     278      278    NoRHC          3.753E-01 cat1 :age
         11     1278     1278    NoRHC          3.462E-01 cat2
         22      291      291    RHC            4.813E-01 wtkilo1
         44T     108      108    NoRHC          3.287E-01 pafi1
         45T     183      183    RHC            3.834E-01 resp
         23T     987      987    NoRHC          2.898E-01 wtkilo1
          3     1163     1163    NoRHC          2.615E-01 aps1
          6T     895      895    NoRHC          1.666E-01 card
          7      268      268    RHC            4.691E-01 cat2
         14T      72       72    RHC            3.052E-01 meanbp1
         15      196      196    NoRHC          4.635E-01 income
         30T      25       25    RHC            2.570E-01 wblc1
         31T     171      171    NoRHC          4.154E-01 card

Number of terminal nodes of final tree: 16
Total number of nodes of final tree: 31
Second best split variable (based on curvature test) at root node is aps1

Classification tree:
For categorical variable splits, values not in training data go to the right

Node 1: cat1 = "ARF", "CHF", "MOSF w/Malignancy", "MOSF w/Sepsis"
  Node 2: pafi1 <= 188.43750
    Node 4: crea1 <= 1.2498779
      Node 8: pafi1 <= 116.48438
        Node 16: RHC
      Node 8: pafi1 > 116.48438 or NA
        Node 17: trauma = "Yes"
          Node 34: RHC
        Node 17: trauma /= "Yes"
          Node 35: card = "Yes"
            Node 70: RHC
          Node 35: card /= "Yes"
            Node 71: NoRHC
    Node 4: crea1 > 1.2498779 or NA
      Node 9: adld3p = NA
        Node 18: RHC
```

```
       Node 9: adld3p /= NA
         Node 19: resp1 <= 29.500000 or NA
           Node 38: RHC
         Node 19: resp1 > 29.500000
           Node 39: NoRHC
   Node 2: pafi1 > 188.43750 or NA
     Node 5: cat1 = "CHF", "MOSF w/Sepsis"
       Node 10: meanbp1 <= 98.500000 or NA
         Node 20: RHC
       Node 10: meanbp1 > 98.500000
         Node 21: NoRHC
     Node 5: cat1 /= "CHF", "MOSF w/Sepsis"
       Node 11: cat2 = "MOSF w/Sepsis"
         Node 22: wtkilo1 <= 66.449950
           Node 44: NoRHC
         Node 22: wtkilo1 > 66.449950 or NA
           Node 45: RHC
       Node 11: cat2 /= "MOSF w/Sepsis"
         Node 23: NoRHC
 Node 1: cat1 /= "ARF", "CHF", "MOSF w/Malignancy", "MOSF w/Sepsis"
   Node 3: aps1 <= 61.500000
     Node 6: NoRHC
   Node 3: aps1 > 61.500000 or NA
     Node 7: cat2 = "MOSF w/Sepsis"
       Node 14: RHC
     Node 7: cat2 /= "MOSF w/Sepsis"
       Node 15: income = "$25-$50k", "> $50k"
         Node 30: RHC
       Node 15: income /= "$25-$50k", "> $50k"
         Node 31: NoRHC

 ***************************************************************

 Predictor means below are means of cases with no missing values.
 Regression coefficients are computed from the complete cases.

Node 1: Intermediate node
 A case goes into Node 2 if cat1 = "ARF", "CHF", "MOSF w/Malignancy", "MOSF w/Sepsis"
 cat1 mode = "ARF"
 Number of observations in node = 5735
 Regressor     Coefficient  t-stat       p-value
 Constant      0.3064        38.80        0.000
 swang1.RHC    0.7364E-01    5.756       0.9026E-08
 Number of observations in node = 5735
 ----------------------------
Node 2: Intermediate node
```

```
 A case goes into Node 4 if pafi1 <= 188.43750
 pafi1 mean = 215.63083
 Number of observations in node = 4572
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if crea1 <= 1.2498779
 crea1 mean = 2.1359302
 Number of observations in node = 2218
 ----------------------------
Node 8: Intermediate node
 A case goes into Node 16 if pafi1 <= 116.48438
 pafi1 mean = 120.46293
 Number of observations in node = 823
 ----------------------------
Node 16: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.3115        8.801       0.7772E-15
 swang1.RHC    0.9494E-01    1.907       0.5729E-01
 Number of observations in node = 370
 ----------------------------
Node 17: Intermediate node
 A case goes into Node 34 if trauma = "Yes"
 trauma mode = "No"
 Number of observations in node = 453
 ----------------------------
Node 34: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.1388E-16   0.7101E-16    1.000
 swang1.RHC    0.8333E-01   0.3948       0.6999
 Number of observations in node = 14
 ----------------------------
Node 35: Intermediate node
 A case goes into Node 70 if card = "Yes"
 card mode = "No"
 Number of observations in node = 439
 ----------------------------
Node 70: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.2759        5.134       0.1314E-05
 swang1.RHC   -0.1330       -1.675       0.9692E-01
 Number of observations in node = 107
 ----------------------------
Node 71: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.3049        10.31        0.000
 swang1.RHC    0.2070E-01   0.3563       0.7219
 Number of observations in node = 332
 ----------------------------
Node 9: Intermediate node
 A case goes into Node 18 if adld3p = NA
 adld3p mean = 0.95617530
 Number of observations in node = 1395
 ----------------------------
Node 18: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.4460        18.28       0.1665E-14
 swang1.RHC    0.1338E-01   0.4371       0.6622
 Number of observations in node = 1144
 ----------------------------
Node 19: Intermediate node
 A case goes into Node 38 if resp1 <= 29.500000 or NA
 resp1 mean = 29.781377
 Number of observations in node = 251
 ----------------------------
Node 38: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.1132        3.781       0.2521E-03
 swang1.RHC   -0.1132       -2.766       0.6640E-02
 Number of observations in node = 114
 ----------------------------
Node 39: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.7273E-01    2.911       0.4218E-02
 swang1.RHC    0.1347E-02   0.2393E-01   0.9809
 Number of observations in node = 137
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if cat1 = "CHF", "MOSF w/Sepsis"
 cat1 mode = "ARF"
 Number of observations in node = 2354
 ----------------------------
Node 10: Intermediate node
 A case goes into Node 20 if meanbp1 <= 98.500000 or NA
 meanbp1 mean = 74.108451
 Number of observations in node = 1076
 ----------------------------
Node 20: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.2111        9.138        0.000
 swang1.RHC    0.9482E-01    3.041       0.2437E-02
 Number of observations in node = 798
 ----------------------------
```

```
Node 21: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.1576        5.723       0.2719E-07
 swang1.RHC    0.1357        2.559       0.1103E-01
 Number of observations in node = 278
 ----------------------------
Node 11: Intermediate node
 A case goes into Node 22 if cat2 = "MOSF w/Sepsis"
 cat2 mode = "NA"
 Number of observations in node = 1278
 ----------------------------
Node 22: Intermediate node
 A case goes into Node 44 if wtkilo1 <= 66.449950
 wtkilo1 mean = 72.582100
 Number of observations in node = 291
 ----------------------------
Node 44: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.3133        6.046       0.2252E-07
 swang1.RHC    0.4675E-01   0.4341       0.6651
 Number of observations in node = 108
 ----------------------------
Node 45: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.3261        6.466       0.8565E-09
 swang1.RHC    0.9150E-01    1.279       0.2024
 Number of observations in node = 183
 ----------------------------
Node 23: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.2725        17.00        0.000
 swang1.RHC    0.5074E-01    1.418       0.1567
 Number of observations in node = 987
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if aps1 <= 61.500000
 aps1 mean = 47.874463
 Number of observations in node = 1163
 ----------------------------
Node 6: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.3425        20.29        0.000
 swang1.RHC    0.4522E-01   0.8866       0.3756
 Number of observations in node = 895
 ----------------------------
Node 7: Intermediate node
```

```
 A case goes into Node 14 if cat2 = "MOSF w/Sepsis"
 cat2 mode = "NA"
 Number of observations in node = 268
 ----------------------------
Node 14: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.7000        8.598       0.1447E-11
 swang1.RHC    0.6190E-01   0.5807       0.5633
 Number of observations in node = 72
 ----------------------------
Node 15: Intermediate node
 A case goes into Node 30 if income = "$25-$50k", "> $50k"
 income mode = "Under $11k"
 Number of observations in node = 196
 ----------------------------
Node 30: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.4444        2.617       0.1543E-01
 swang1.RHC   -0.6944E-01  -0.3271       0.7466
 Number of observations in node = 25
 ----------------------------
Node 31: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.5294        11.96        0.000
 swang1.RHC    0.2206        2.748       0.6641E-02
 Number of observations in node = 171
 ----------------------------
 Regression estimates are weighted means over terminal nodes
 Regressor     Coefficient  z-stat       p-value
 Constant      0.3160        38.52        0.000
 swang1.RHC    0.5191E-01    3.597       0.3222E-03

 Average treatment effect of swang1 level "RHC" vs level "NoRHC" =  5.1909E-02

 Observed and fitted values are stored in prop30.fit
 LaTeX code for tree is in prop30.tex
 R code is stored in prop30.r
```

The results at the end of `prop30.out` show that the average treatment effect is 0.061634. The LaTeX tree is shown in Figure 40. The number beside each terminal node is the proportion of observations with `swang1 = RHC` ($Z = 1$). The pair below each node are the sample means of Y corresponding to $Z = 0$ and 1. GUIDE treats "`NoRHC`" as $Z = 0$ because it precedes "`RHC`" in alphabetical order.

The file `prop30.fit` gives the proportions of `swang1` in the rightmost two columns.

Here are the top 5 rows of the file:

```
train   node  observed  predicted  "P(NoRHC)"     "P(RHC)"
    y      6  "NoRHC"    "NoRHC"   0.89050E+00  0.10950E+00
    y     20  "RHC"      "RHC"     0.45113E+00  0.54887E+00
    y     45  "RHC"      "RHC"     0.50273E+00  0.49727E+00
    y     18  "NoRHC"    "RHC"     0.36451E+00  0.63549E+00
    y     20  "RHC"      "RHC"     0.45113E+00  0.54887E+00
```

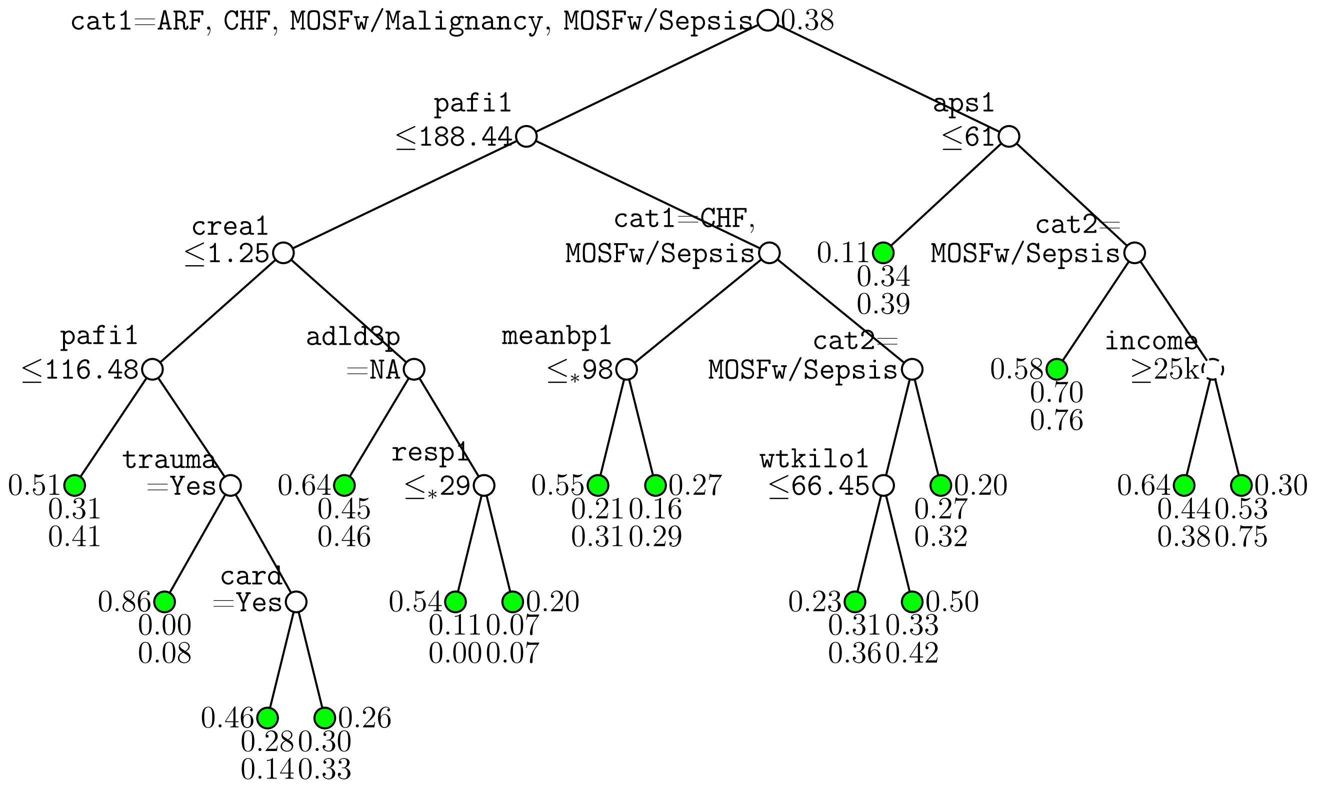

Figure 40: GUIDE v.38.0 0.25-SE tree for propensity score grouping and estimation of effects of `swang1` on `dth30`. Tree constructed with 5735 observations. Maximum number of split levels is 15 and minimum node sample size is 6. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. Circles with dashed lines are nodes with no significant split variables. Sample means of `dth30` for `swang1` levels `NoRHC` and `RHC`, respectively, printed below nodes. Sample proportion of `swang1` = `RHC` printed beside nodes. Second best split variable at root node is `aps1`.

18 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 dxcurren 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
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 5
Option 5 is for differential item functioning.
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
2 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;
choose univariate otherwise or if item response
Input 1 for multivariate, 2 for univariate ([1:2], <cr>=1): 2
D variables can be normalized to have unit variance,
e.g., if they have different scales or units
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 ...
Finished checking
```

```
Missing values found in D variables
Assigning integer codes to values of 1 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Some D variables have missing values
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
      1978          0          0        4        0        0        2
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        1        0
Number of cases used for training: 1977
Number of split variables: 3
Number of cases excluded due to 0 weight or missing D: 1
Finished reading data file
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 will contain info for DIF items.
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): dif.tex
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 rank of top variable to split root node ([1:3], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < dif.in
```

The importance scores are in the file `dif.scr`. They show that `age` is most important, followed by `gender` and `education`.

```
Rank        Score    Variable
1.00  8.94327E+00  age
2.00  5.06849E+00  gender
3.00  3.38749E+00  education
```

The word '`yes`' in the last column of `dif.pv` below shows which item has DIF. In this example, only item #10 (`memory`) has DIF.

```
Item      Itemname    education          age       gender   DIF
   1         satis    0.492E-01    0.399E-01    0.101E+00   no
   2          drop    0.146E-01    0.228E+00    0.923E+00   no
   3         empty    0.207E-02    0.141E+00    0.185E+00   no
   4         bored    0.312E-05    0.212E+00    0.299E+00   no
   5        spirit    0.960E+00    0.737E+00    0.388E-01   no
   6        afraid    0.318E-01    0.472E-03    0.273E-02   no
   7         happy    0.763E+00    0.345E+00    0.251E-01   no
   8          help    0.463E-01    0.611E+00    0.443E-02   no
   9          home    0.371E+00    0.120E+00    0.814E-03   no
  10        memory    0.373E+00    0.000E+00    0.206E-01   yes
  11         alive    0.169E+00    0.155E+00    0.438E+00   no
  12         worth    0.332E+00    0.726E+00    0.696E+00   no
  13        energy    0.660E+00    0.652E+00    0.126E-03   no
  14          hope    0.638E+00    0.392E+00    0.213E+00   no
  15        better    0.517E+00    0.621E+00    0.447E+00   no
```

Figure 41 shows the tree.

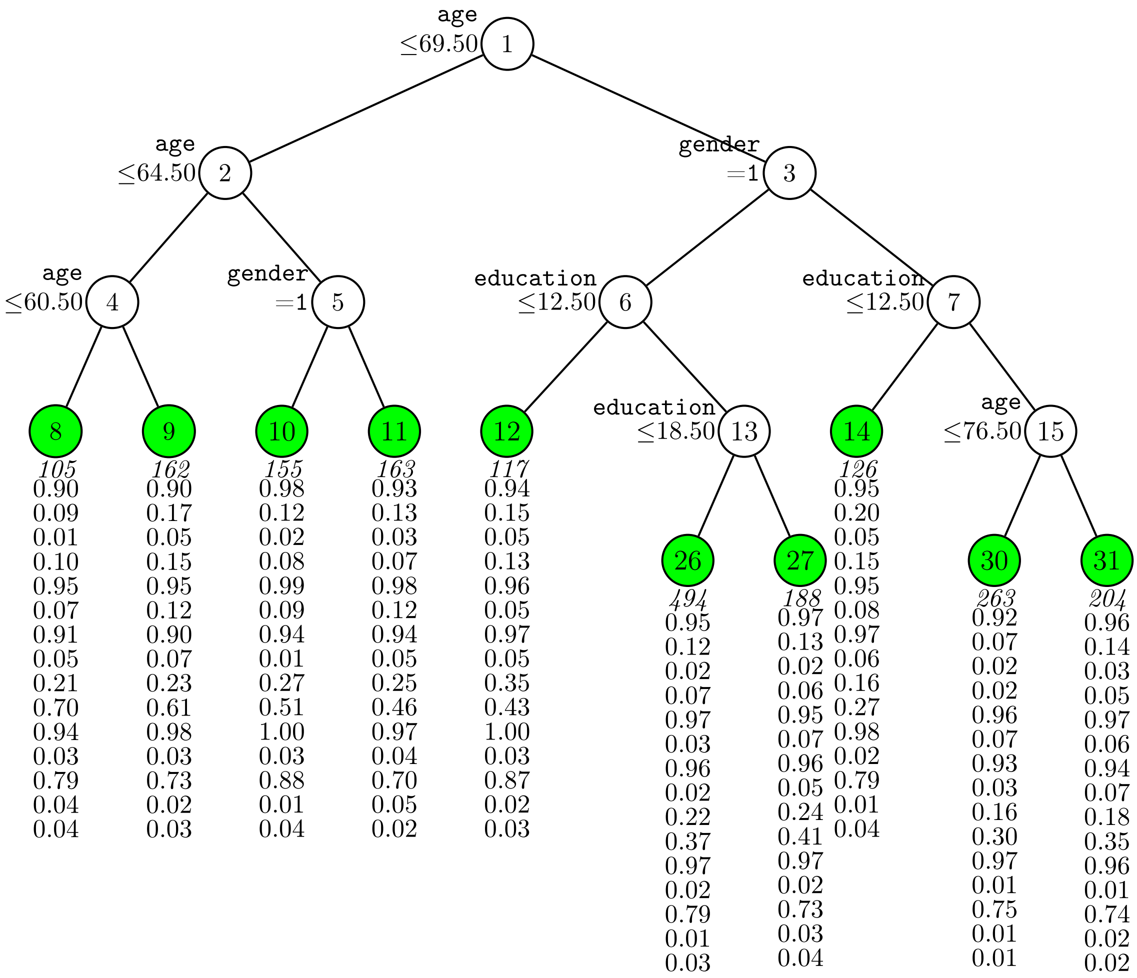

Figure 41: GUIDE v.38.0 importance scoring or DIF regression tree for predicting response variables `satis`, `drop`, `empty`, `bored`, `spirit`, `afraid`, `happy`, `help`, `home`, `memory`, `alive`, `worth`, `energy`, `hope`, and `better`, without using PCA at each node. Tree constructed with 1977 observations (excluding observations with non-positive weight or missing values in `D`, `T`, `R` or `Z` variables). Maximum number of split levels is 4 and minimum node sample size is 98. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (*in italics*) and predicted values of `satis`, `drop`, `empty`, `bored`, `spirit`, `afraid`, `happy`, `help`, `home`, `memory`, `alive`, `worth`, `energy`, `hope`, and `better` printed below nodes. Second best split variable at root node is `gender`.

19 Bootstrap confidence intervals

Owing to the numerous procedures that are performed during tree construction (such as selection of the variable and the split set to partition each intermediate node), proper statistical inference must account for the multiple testing and estimation issues. Otherwise, the error variance will be underestimated. Suppose, for example, we wish to obtain confidence intervals for the proportion of "`RHC`" in each terminal node of the tree in Figure 1. Let n denote the sample size in a node and $\hat{p}$ the proportion of observations in it with the response value `RHC`. The usual $(1-\alpha)$ binomial interval is then $\hat{p} \pm z_{1-\alpha/2}\sqrt{\hat{p}(1-\hat{p})/n}$, where z_α is the α -quantile of the standard normal distribution. This formula yields intervals that are too short because it does not account for the extra variance due to model construction. Bonferroni corrections, which are traditionally used for multiple testing, are inapplicable here because the number of tests are not specified in advance. For example, the number of chi-squared tests at each node depends on the number of variables eligible to split the node and the number of levels of splits depends on the total sample size, extent of pruning, and other parameters such as the minimum sample size in each node.

As with the Bonferroni correction, a natural solution is to change the multiplier $z_{1-\alpha/2}$ to a larger value. The bootstrap method provides one simple solution. Called "bootstrap calibration", the procedure is described and analyzed in Loh (1987, 1991) in the context of estimating a nonparametric mean; it is extended to subgroup analysis from regression tree models in Loh et al. (2016, 2019c) and Loh and Zhou (2020). The R code below implements the procedure. It can be used by following these steps:

1. Change the name of the data file (`rhcdata.txt` here) to `realdata.txt`.

2. Change the name of the description file (`rhcdsc1.txt` here) to `real.dsc`.

3. Change the name of the GUIDE input file (`classin.txt` here) to `real.in`.

4. Change the word "`RHC`" in line 1 of the R code to the name of the desired class in the data file.

5. In Windows, change the word "`system`" in lines 32, 32, 74 and 75 to "`shell`" if necessary.

6. Source the program in R.

```
class.name <- "RHC"  ## name of desired class in realdata.txt
nboot <- 1000
probs <- c(0.80,0.90,0.95,0.98)
zstat <- rep(0,nboot)
### write bootstrap description file boot.dsc
file <- readLines("real.dsc")  ## read real description file
write("bootdata.txt",file="boot.dsc")
len <- length(file)
write(file[2:length(file)],"boot.dsc",append=TRUE)
write(paste(len-2,"w␣w"),"boot.dsc",append=TRUE)
### write bootstrap input file boot.in
file <- readLines("real.in")  ## read real input file
file2 <- gsub("real.","boot.",file) ## replace "real." with "boot."
write(file2,"boot.in")
### read real data
z0 <- read.table("realdata.txt",header=TRUE)
nobs <- nrow(z0)
zt <- cbind(z0,rep(0,nobs)) ### add column of weight 0
write("Bootstrap␣simultaneous␣intervals␣by␣linear␣interpolation␣of␣z",
      "results.txt")
write("trials␣␣z80␣␣␣␣z90␣␣␣␣z95␣␣␣␣␣z98␣␣␣␣␣␣bias.err␣␣␣␣␣sd.err",
      "results.txt", append=TRUE)
err.test <- rep(0,nboot) ## misclassification rates
bias <- 0
for(i in 1:nboot){
    zb <- z0[sample(nobs,nobs,replace=TRUE),]
    zb <- cbind(zb,rep(1,nobs)) ### add column of weight 1
    write.table(zb,"bootdata.txt",col.names=TRUE,row.names=FALSE)
    write.table(zt,"bootdata.txt",col.names=FALSE,row.names=FALSE,
                append=TRUE)
    system("rm␣-f␣log.txt␣boot.out␣boot.fit")
    system("guide␣<␣boot.in␣>␣log.txt")
    bfit <- read.table("boot.fit",header=TRUE)  ## read boot results
    test <- bfit$train == "n"
    err.test[i] <- sum(bfit$observed[test] != bfit$predicted[test])/nobs
    err.resub <- sum(bfit$observed[!test] != bfit$predicted[!test])/nobs
    bias <- bias+(err.resub-err.test[i])
    unodes <- unique(sort(bfit$node))
    for(j in 1:length(unodes)){
        gp <- bfit$node == unodes[j] & bfit$train == "y" ## training data
        n0 <- sum(bfit$observed[gp] != class.name)
        n1 <- sum(bfit$observed[gp] == class.name)
        ntot <- n0+n1
        estp <- n1/ntot
        if(n1 == 0 | n0 == 0){
            p <- (n1+0.5)/(ntot+1)
            sd <- sqrt(p*(1-p)/(ntot+1))
        } else {
            sd <- sqrt(estp*(1-estp)/ntot)
        }
        gp <- bfit$node == unodes[j] & bfit$train == "n"  ## real data
        n0 <- sum(bfit$observed[gp] != class.name)
        n1 <- sum(bfit$observed[gp] == class.name)
        realp <- n1/(n0+n1)
        zstat[i] <- max(zstat[i],abs(realp-estp)/sd)
    }
    if(i %% 100 == 0){
        sd.err <- sqrt(var(err.test[1:i])) ## linear interpolation
        q <- quantile(zstat[1:i],probs=probs,type=4)
        write(c(i,q,bias/i,sd.err),"results.txt",append=TRUE,ncol=7)
    }
}
### find calibrated z.alpha
write(paste("No.␣bootstraps␣=␣",nboot),"results.txt",append=TRUE)
write(c("Calibrated␣z␣at␣levels",probs),file="results.txt",ncol=5,
      append=TRUE)
q <- quantile(zstat,probs=probs,type=4) ## linear interpolation
write(q,"results.txt",append=TRUE,ncol=4)
write(paste("Bootstrap␣estimate␣of␣bias␣of␣error␣rate␣=␣",bias/nboot),
      "results.txt",append=TRUE)
write(paste("Bootstrap␣estimate␣of␣SD␣of␣error␣rate␣=␣",
      sqrt(var(err.test))),"results.txt",append=TRUE)
### fit real data
system("rm␣-f␣log.txt␣real.out␣real.fit")
system("guide␣<␣real.in␣>␣log.txt")
realfit <- read.table("real.fit",header=TRUE)
train <- realfit$train == "y"
err.obs <- sum(realfit$observed[train] != realfit$predicted[train])/nobs
write(paste("Real␣data␣observed␣error␣rate␣=␣",err.obs),"results.txt",
      append=TRUE)
k <- 3     ## 95% level
z0 <- q[k] ## 95% z value
write(c("Simultaneous␣intervals␣at␣level",probs[k]),
      file="results.txt",ncol=2,append=TRUE)
write(paste0("Node␣␣N␣␣␣P(",class.name,")␣halfwid␣␣␣left␣␣␣right"),
      "results.txt", append=TRUE)
unodes <- unique(sort(realfit$node))
for(j in 1:length(unodes)){
    gp <- realfit$node == unodes[j] & realfit$train == "y"
    n0 <- sum(realfit$observed[gp] != class.name)
    n1 <- sum(realfit$observed[gp] == class.name)
    ntot <- n0+n1
    if(n1 == 0 | n0 == 0){
        p <- (n1+0.5)/(ntot+1)
        sd <- sqrt(p*(1-p)/(ntot+1))
    } else {
        p <- n1/ntot
        sd <- sqrt(p*(1-p)/(ntot))
    }
    p <- n1/ntot
    halfwid <- z0*sd
    left <- p-halfwid
    rght <- p+halfwid
    write(c(unodes[j],ntot,p,halfwid,left,rght),"results.txt",
          append=TRUE,ncol=6)
}
## write(sort(zstat),"zstat.txt",ncol=1) ## output sorted zstat values
```

Figure 42 gives the contents of the file `results.txt`. It shows that the calibrated z-multiplier is 3.961722, 4.325215, 4.690964, or 5.337637 for 80%, 90%, 95%, or 98% simultaneous confidence intervals. For 95% intervals, the left and right end points of the intervals in each terminal node are given in the bottom half of the file. These intervals are printed below the terminal nodes in Figure 43.

20 Tree ensembles

A tree ensemble is a collection of trees. GUIDE has two methods of constructing an ensemble.

GUIDE forest. This the preferred method. 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; Random Forest uses the biased CART method. One consequence is that GUIDE forest can be very much faster than Random Forest if the dependent variable is a class variable having more than two distinct values and some categorical predictor variables have many categories.
2. GUIDE forest is applicable to data with missing values. The R implementation of Random Forest (Liaw and Wiener, 2002) requires apriori imputation of missing values in the predictor variables.

```
Bootstrap simultaneous intervals by linear interpolation of z
trials  z80    z90    z95     z98      bias.err     sd.err
100 4.036962 4.458809 4.545827 4.922293 -0.03357803 0.005906056
200 4.123996 4.508203 4.777955 5.035208 -0.03335222 0.005670584
300 4.093978 4.513735 4.918732 5.117146 -0.0335048 0.00598086
400 4.108083 4.519645 4.835633 5.28808 -0.03360811 0.005930667
500 4.108083 4.508203 4.826329 5.117146 -0.03377507 0.005887693
600 4.144132 4.548011 4.895352 5.408027 -0.03397879 0.005812075
700 4.123996 4.529434 4.889087 5.408027 -0.03377357 0.005839512
800 4.117319 4.51814 4.845685 5.365021 -0.03369159 0.00588305
900 4.108552 4.50332 4.835633 5.408027 -0.03358888 0.005924705
1000 4.108083 4.495735 4.845685 5.397256 -0.03353304 0.005951228
No. bootstraps =  1000
Calibrated z at levels 0.8 0.9 0.95 0.98
4.108083 4.495735 4.845685 5.397256
Bootstrap estimate of bias of error rate =  -0.0335330427201395
Bootstrap estimate of SD of error rate =  0.00595122775778847
Real data observed error rate =  0.296251089799477
Simultaneous intervals at level 0.95
Node  N   P(RHC) halfwid   left   right
5 566 0.3816254 0.09894446 0.282681 0.4805699
7 2760 0.2355072 0.03913718 0.1963701 0.2746444
8 655 0.6961832 0.08707675 0.6091065 0.78326
18 244 0.6270492 0.1500158 0.4770334 0.7770649
19 218 0.3853211 0.1597212 0.2255999 0.5450423
25 66 0.3484848 0.2842088 0.06427609 0.6326936
26 110 0.6363636 0.2222518 0.4141119 0.8586154
27 601 0.3627288 0.09503228 0.2676965 0.4577611
48 438 0.6552511 0.1100458 0.5452053 0.7652969
49 77 0.3506494 0.2635033 0.08714608 0.6141526
```

Figure 42: Contents of `results.txt`

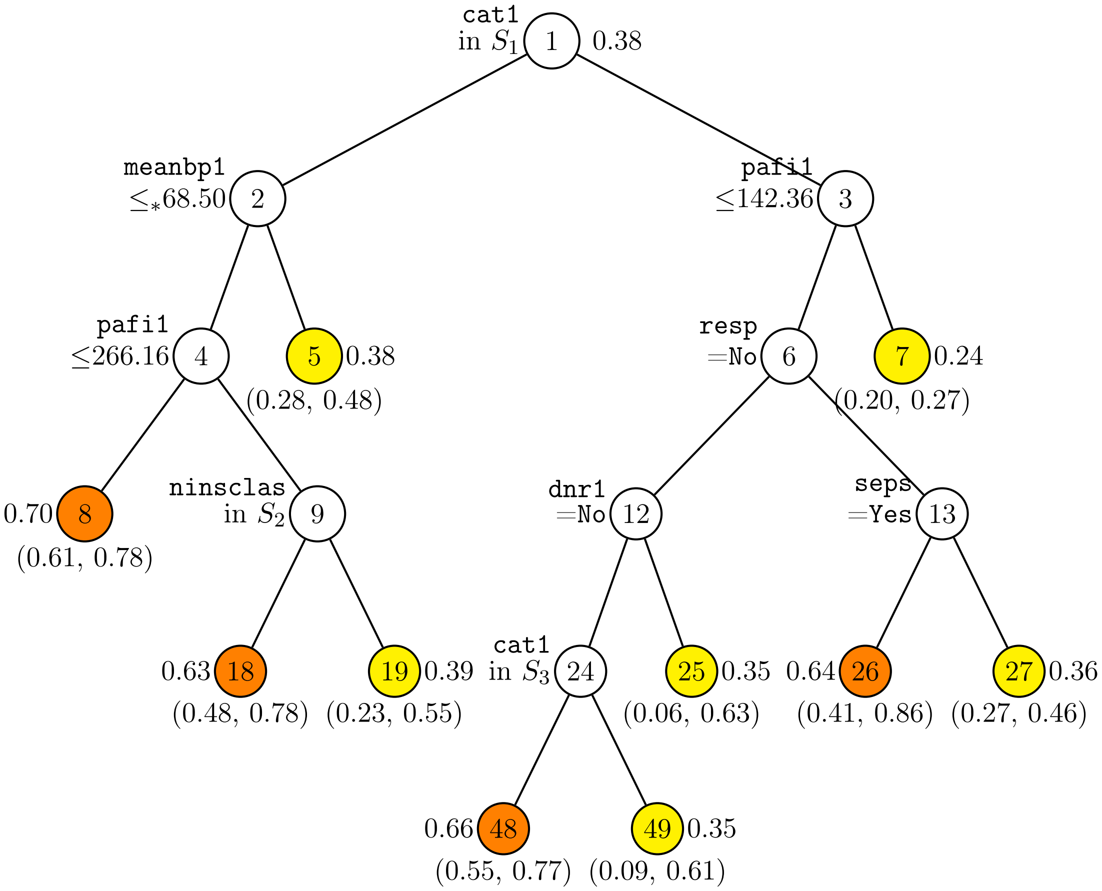

Figure 43: GUIDE v.38.0 0.25-SE classification tree for predicting `swang1` using estimated priors and unit misclassification costs. Tree constructed with 5735 observations. Maximum number of split levels is 15 and minimum node sample size is 57. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '≤ or missing'. Set S_1 = {`CHF`, `MOSF w/Sepsis`}. Set S_2 = {`No insurance`, `Private`, `Private & Medicare`}. Set S_3 = {`ARF`, `Lung Cancer`, `MOSF w/Malignancy`}. Predicted classes and sample sizes printed below terminal nodes; class sample proportion for `swang1` = `RHC` beside nodes. Bootstrap calibrated 95% simultaneous intervals for proportion of `RHC` below nodes.

The default number of trees for GUIDE forest is 1000 if there are fewer than 500 training samples and 100 predictor variables; otherwise, the default is 500.

Bagged GUIDE. This fits *pruned* GUIDE trees to bootstrap samples of the training data (Breiman, 1996). Each tree is pruned by 5-fold cross-validation. The default number of trees is 200 if there are fewer than 500 training samples and 100 predictor variables; otherwise, the default is 100.

With the default settings, GUIDE forest is typically much faster than bagged GUIDE.

20.1 GUIDE forest: CE data

20.1.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: gf.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: gf.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):
Input 1 for classification, 2 for least-squares 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: ceclass.dsc
Reading data description file ...
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
422 N variables changed to S
D variable is INTRDVX_
Reading data file ...
Number of records in data file: 4693
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
```

```
Finished recoding
Number of classes: 3
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 42 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: S variable MISC2PQ is constant
Warning: S variable MISC2CQ is constant
Warning: S variable TCARTRKP is constant
Warning: S variable TCARTRKC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable VMISCHEP is constant
Warning: S variable VMISCHEC is constant
Warning: S variable ROTHRFLP is constant
Warning: S variable ROTHRFLC is constant
Smallest positive weight:  1.3507E+03
Largest positive weight:   7.0269E+04
Class  #Cases     Proportion
C        1771     0.37737055
D        2838     0.60473045
T          84     0.01789900
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693          0       4693       16        0        0      422
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      171        0       42        0
Number of cases used for training: 4693
Number of split variables: 464
Number of cases excluded due to 0 weight or missing D: 0
Finished reading data file
Warning: No linear splits; number of S variables must be < 225
Choose 1 for estimated priors, 2 for equal priors, 3 to input 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):
Warning: All positive weights treated as 1
```

```
Input name of file to store predicted class and probability: gf.pro
Input rank of top variable to split root node ([1:464], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < gf.in
```

20.1.2 Contents of gf.out

Note: Owing to the intrinsic randomness in forests, your results may differ from those shown below. "OOB" stands for "out-of-bag".

```
Random forest of classification trees
No pruning
Data description file: ceclass.dsc
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
422 N variables changed to S
D variable is INTRDVX_
Number of records in data file: 4693
Length of longest entry in data file: 11
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 3
Warning: S variable MISC2PQ is constant
Warning: S variable MISC2CQ is constant
Warning: S variable TCARTRKP is constant
Warning: S variable TCARTRKC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable VMISCHEP is constant
Warning: S variable VMISCHEC is constant
Warning: S variable ROTHRFLP is constant
Warning: S variable ROTHRFLC is constant
Smallest and largest positive weights are  1.3507E+03 and 7.0269E+04
Training sample class proportions of D variable INTRDVX_:
Class  #Cases    Proportion
C        1771    0.37737055
D        2838    0.60473045
T          84    0.01789900

Summary information for training sample of size 4693
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
Levels of M variables are for missing values in associated variables
                                                     #Codes/
                                                     Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  DIRACC     c                                  2      155
      2  DIRACC_    m                                  1
      3  AGE_REF    s    18.00        87.00
      4  AGE_REF_   m                                  0
      :
     50  FINLWT21   w    1351.       0.7027E+05
      :
    514  INTRDVX_   d                                  3
      :
    651  FSTAXOWE   s   -2505.       0.5991E+05
    652  FSTA_OWE   m                                  0
    653  ETOTA      s    1199.       0.2782E+06

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693          0       4693       16        0        0      422
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      171        0       42        0
Number of cases used for training: 4693
Number of split variables: 464
Number of cases excluded due to 0 weight or missing D: 0

Number of trees in ensemble: 500
Number of variables used for splitting: 155
Warning: No linear splits; number of S variables must be < 225
Simple node models
Estimated priors
Unit misclassification costs
Warning: All positive weights treated as 1
Univariate split highest priority
No interaction splits
No linear splits
Fraction of cases used for splitting each node: .0213
Maximum number of split levels: 19
Minimum node sample size: 23
Mean number of terminal nodes:    140.0

Classification matrix for training sample:
Predicted       True class
class             C         D         T
```

```
 C               1295       68         7
 D                476     2770        77
 T                  0        0         0
 Total           1771     2838        84

 Number of cases used for tree construction: 4693
 Number misclassified: 628
 Resubstitution estimate of mean misclassification cost: .1338

 Number of OOB cases: 4693
 Number OOB misclassified: 1044
 OOB estimate of mean misclassification cost: .2225
 Mean number of trees per OOB observation: 184.09

 Predicted class probabilities are stored in gf.pro
```

Following are the top few rows of the file gf.pro, which give the estimated class posterior probabilities and the predicted and observed values of each case in the data.

```
train   "P(C)"      "P(D)"      "P(T)"   predicted observed
y   0.24084E+00  0.73894E+00  0.20225E-01    "D"    "D"
y   0.27788E+00  0.71166E+00  0.10466E-01    "D"    "D"
y   0.14421E+00  0.85277E+00  0.30243E-02    "D"    "D"
y   0.19806E+00  0.79337E+00  0.85726E-02    "D"    "D"
y   0.13849E+00  0.84833E+00  0.13183E-01    "D"    "D"
y   0.18939E+00  0.72913E+00  0.81487E-01    "D"    "D"
y   0.56962E+00  0.41926E+00  0.11125E-01    "C"    "C"
y   0.43331E+00  0.54046E+00  0.26231E-01    "D"    "D"
y   0.22160E+00  0.76909E+00  0.93104E-02    "D"    "D"
y   0.15031E+00  0.84448E+00  0.52123E-02    "D"    "D"
y   0.52962E+00  0.45551E+00  0.14861E-01    "C"    "C"
y   0.28137E+00  0.69455E+00  0.24074E-01    "D"    "D"
y   0.43100E+00  0.56295E+00  0.60551E-02    "D"    "D"
y   0.32716E+00  0.64616E+00  0.26681E-01    "D"    "D"
y   0.26778E+00  0.70553E+00  0.26693E-01    "D"    "D"
y   0.22588E+00  0.59997E+00  0.17415E+00    "D"    "T"
```

20.2 Bagged GUIDE

20.2.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
```

```
Name of batch input file: bg.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: bg.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 least-squares 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: ceclass.dsc
Reading data description file ...
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
422 N variables changed to S
D variable is INTRDVX_
Reading data file ...
Number of records in data file: 4693
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 3
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 42 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: S variable MISC2PQ is constant
Warning: S variable MISC2CQ is constant
Warning: S variable TCARTRKP is constant
Warning: S variable TCARTRKC is constant
```

```
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable VMISCHEP is constant
Warning: S variable VMISCHEC is constant
Warning: S variable ROTHRFLP is constant
Warning: S variable ROTHRFLC is constant
Smallest positive weight:  1.3507E+03
Largest positive weight:   7.0269E+04
Class  #Cases     Proportion
C        1771     0.37737055
D        2838     0.60473045
T          84     0.01789900
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693          0       4693       16        0        0      422
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      171        0       42        0
Number of cases used for training: 4693
Number of split variables: 464
Number of 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 to input 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):
Warning: All positive weights treated as 1
Input name of file to store predicted class and probability: bg.pro
Input rank of top variable to split root node ([1:464], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < bg.in
```

Results

```
Ensemble of bagged classification trees
Pruning by cross-validation
Data description file: ceclass.dsc
Training sample file: cedata.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 33
422 N variables changed to S
D variable is INTRDVX_
Number of records in data file: 4693
Length of longest entry in data file: 11
```

```
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 3
Warning: S variable MISC2PQ is constant
Warning: S variable MISC2CQ is constant
Warning: S variable TCARTRKP is constant
Warning: S variable TCARTRKC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable VMISCHEP is constant
Warning: S variable VMISCHEC is constant
Warning: S variable ROTHRFLP is constant
Warning: S variable ROTHRFLC is constant
Smallest and largest positive weights are  1.3507E+03 and 7.0269E+04
Training sample class proportions of D variable INTRDVX_:
Class  #Cases     Proportion
C        1771     0.37737055
D        2838     0.60473045
T          84     0.01789900

Summary information for training sample of size 4693
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
Levels of M variables are for missing values in associated variables
                                                   #Codes/
                                                   Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  DIRACC     c                                  2      155
      2  DIRACC_    m                                  1
      3  AGE_REF    s    18.00        87.00
      4  AGE_REF_   m                                  0
      :
     50  FINLWT21   w    1351.       0.7027E+05
      :
    514  INTRDVX_   d                                  3
      :
    651  FSTAXOWE   s   -2505.       0.5991E+05
    652  FSTA_OWE   m                                  0
    653  ETOTA      s    1199.       0.2782E+06

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4693          0       4693       16        0        0      422
     #P-var   #M-var   #B-var   #C-var   #I-var
          0      171        0       42        0
 Number of cases used for training: 4693
 Number of split variables: 464
 Number of cases excluded due to 0 weight or missing D: 0

 Number of trees in ensemble: 100
 Pruning by v-fold cross-validation, with v = 5
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: 0.2500

 Warning: No interaction tests; too many predictor variables
 Simple node models
 Estimated priors
 Unit misclassification costs
 Warning: All positive weights treated as 1
 Univariate split highest priority
 No interaction splits
 No linear splits
 Fraction of cases used for splitting each node: .0213
 Maximum number of split levels: 19
 Minimum node sample size: 23
 Mean number of terminal nodes:    42.71

 Classification matrix for training sample:
 Predicted       True class
 class              C         D         T
 C                933       104         4
 D                838      2734        80
 T                  0         0         0
 Total           1771      2838        84

 Number of cases used for tree construction: 4693
 Number misclassified: 1026
 Resubstitution estimate of mean misclassification cost: .2186

 Number of OOB cases: 4693
 Number OOB misclassified: 1211
 OOB estimate of mean misclassification cost: .2580
 Mean number of trees per OOB observation: 36.70

 Predicted class probabilities are stored in bg.pro
```

The top few lines of bg.pro follow.

```
train   "P(C)"      "P(D)"      "P(T)"    predicted observed
```

```
y  0.23952E+00  0.74062E+00  0.19855E-01   "D"   "D"
y  0.26571E+00  0.71911E+00  0.15187E-01   "D"   "D"
y  0.18557E+00  0.80576E+00  0.86795E-02   "D"   "D"
y  0.21886E+00  0.77037E+00  0.10773E-01   "D"   "D"
y  0.18362E+00  0.79827E+00  0.18107E-01   "D"   "D"
y  0.19654E+00  0.74734E+00  0.56120E-01   "D"   "D"
y  0.49944E+00  0.48560E+00  0.14956E-01   "C"   "C"
y  0.50615E+00  0.47238E+00  0.21473E-01   "C"   "D"
y  0.23564E+00  0.75206E+00  0.12297E-01   "D"   "D"
y  0.17405E+00  0.81524E+00  0.10715E-01   "D"   "D"
y  0.49660E+00  0.48759E+00  0.15814E-01   "C"   "C"
y  0.31610E+00  0.66736E+00  0.16547E-01   "D"   "D"
y  0.38996E+00  0.60141E+00  0.86378E-02   "D"   "D"
y  0.34223E+00  0.62059E+00  0.37171E-01   "D"   "D"
y  0.26170E+00  0.71616E+00  0.22138E-01   "D"   "D"
y  0.23953E+00  0.65024E+00  0.11022E+00   "D"   "T"
```

21 Other features

21.1 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.

21.2 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 or 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.

21.3 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`.

21.4 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 $\texttt{wtgain}^2$, 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 $\texttt{wtgain}^2$ for fitting terminal node models only. Note: The line defines $\texttt{wtgain}^2$ as $\texttt{wtgain}^2 \times \texttt{wtgain}^0$. Since we can equivalently define the variable by $\texttt{wtgain}^2 = \texttt{wtgain}^1 \times \texttt{wtgain}^1$, 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 `wtgain`^{2} in each terminal node of the tree.

21.5 Data formatting functions

GUIDE has a utility function for reformatting data files into forms required by some old 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 NHTSA comma-separated data are reformatted to tab-delimited for R or Splus.

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: format.in
 Input 1 for model fitting, 2 for importance or DIF scoring,
       3 for data conversion ([1:3], <cr>=1): 3
 Name of batch output file: format.out
 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: nhtsaclass.dsc
nhtsaclass.dsc
 Reading data description file ...
 Training sample file: nhtsadata.csv
 Missing value code: NA
 Records in data file start on line 2
 Warning: 48 N variables changed to S
 Dependent variable is HIC2
 Reading data file ...
 Number of records in data file: 3310
 Length of longest entry in data file: 19
 Checking for missing values ...
 Total number of cases: 3310
 Number of classes: 2

 Warning: "x" variables will be excluded
 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: newdata.txt
 Input file is created!
 Run GUIDE with the command: guide < format.in
```

References

Andersen, P. K., Hansen, M. G., and Klein, J. P. (2004). Regression analysis of restricted mean survival time based on pseudo-observations. *Lifetime Data Analysis*, 10:335–350.

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 geriatic depression scale in an Asian population. *Journal of Affective Disorders*, 108:285–290.

Cameron, A. A. and Trivedi, P. K. (1998). *Regression Analysis of Count Data*. Cambridge University Press.

Chambers, J. M. and Hastie, T. J. (1992). An appetizer. In Chambers, J. M. and Hastie, T. J., editors, *Statistical Models in S*, pages 1–12. Wadsworth & Brooks/Cole, Pacific Grove.

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.

Chen, P. Y. and Tsiatis, A. A. (2001). Causal inference on the difference of the restricted mean lifetime between two groups. *Biometrics*, 57:1030–1038.

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.

Connors, Jr., A. F., , Speroff, T., Dawson, N. V., et al. (1996). The effectiveness of right heart catheterization in the initial care of critically ill patients. *JAMA*, 276(11):889–897.

Deb, P. and Trivedi, P. K. (1997). Demand for medical care by the elderly: a finite mixture approach. *Journal of Applied Econometrics*, 12:313–336.

Hothorn, T. (2017). *TH.data: TH's Data Archive*. R package version 1.0-8.

Hothorn, T., Hornik, K., and Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15:651–674.

Hothorn, T. and Zeileis, A. (2015). partykit: A modular toolkit for recursive partytioning in r. *Journal of Machine Learning Research*, 16:3905–3909.

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.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

Loh, W.-Y. (1987). Calibrating confidence coefficients. *Journal of the American Statistical Association*, 82:155–162.

Loh, W.-Y. (1991). Bootstrap calibration for confidence interval construction and selection. *Statistica Sinica*, 1:477–491.

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. (2021). Logistic regression tree analysis. In Pham, H., editor, *Handbook of Engineering Statistics*. Springer, 2nd edition. To appear.
http://www.stat.wisc.edu/~loh/treeprogs/guide/logistic2.pdf.

Loh, W.-Y., Cao, L., and Zhou, P. (2019a). Subgroup identification for precision medicine: a comparative review of thirteen methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(5):e1326.
http://www.stat.wisc.edu/~loh/treeprogs/guide/wires19.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. (2019b). Classification and regression trees and forests for incomplete data from sample surveys. *Statistica Sinica*, 29:431–453.
http://www.stat.wisc.edu/~loh/treeprogs/guide/LECL19.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. (2019c). Subgroups from regression trees with adjustment for prognostic effects and post-selection inference. *Statistics in Medicine*, 38:545–557. http://www.stat.wisc.edu/~loh/treeprogs/guide/sm19.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., Zhang, Q., Zhang, W., and Zhou, P. (2020). Missing data, imputation and regression trees. *Statistica Sinica*. In press. http://www.stat.wisc.edu/~loh/treeprogs/guide/LZZZ20.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/AOAS596.pdf.

Loh, W.-Y. and Zhou, P. (2020). The GUIDE approach to subgroup identification. In Ting, N., Cappelleri, J. C., Ho, S., and Chen, D.-G., editors, *Design and analysis of Subgroups with Biopharmaceutical Applications*, pages 147–165. Springer. http://www.stat.wisc.edu/~loh/treeprogs/guide/LZ20.pdf.

Loh, W.-Y. and Zhou, P. (2021). Variable importance scores. arXiv 2102.07765v1. http://www.stat.wisc.edu/~loh/treeprogs/guide/LZ21.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.

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.

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.

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.

Tian, L., Zhao, L., and Wei, L. J. (2014). Predicting the restricted mean event time with the subject's baseline covariates in survival analysis. *Biostatistics*, 15:222–233.

Witten, I. and Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations*. Morgan Kaufmann, San Fransico, CA. http://www.cs.waikato.ac.nz/ml/weka.

Zeileis, A. (2006). Object-oriented computation of sandwich estimators. *Journal of Statistical Software*, 16(9):1–16.