User Manual for GUIDE ver. 36.2*

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

January 10, 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.
5. Missing-value flag variables.
6. Periodic or cyclic variables, such as angular direction, hour of day, day of week, month of year, and seasons.
7. Subgroup identification for differential treatment effects.
8. Linear splits and kernel and nearest-neighbor node models for classification trees.
9. Weighted least squares, least median of squares, logistic, quantile, Poisson, and relative risk (proportional hazards) regression models.
10. Univariate, multivariate, censored, and longitudinal response variables.
11. Pairwise interaction detection at each node.
12. Categorical variables for splitting only, fitting only (via 0-1 dummy variables), or both in regression tree models.
13. 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), and Loh et al. (2007, 2019b, 2016, 2015, 2019c). A list of third-party applications of GUIDE, CRUISE, QUEST, and LOTUS is maintained in `http://www.stat.wisc.edu/~loh/apps.html`. This manual illustrates the use of the GUIDE software and the interpretation of the output.

[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
Case weights	Yes	Yes	No	Yes[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] non-negative integer weights
[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** and **gfortran**. Both versions are compiled in Ubuntu 20.0. If necessary, make the gunzipped file executable by issuing the command "`chmod a+x guide`" in a `Terminal` window.

macOS 11.1 (Big Sur): Double-click the file `guide.gz` to gunzip it and then make it executable by typing the command "`chmod a+x guide`" in a `Terminal` application in the folder where the file is located. This version requires **Xcode 12.3** and **gfortran 10.2** to be installed. 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 /`

macOS 10.15 (Catalina): Double-click the file `guide.gz` to gunzip it and then make it executable by typing the command "`chmod a+x guide`" in a `Terminal` application in the folder where the file is located. This version is compiled with NAG Fortran 6.2 and requires no additional software besides the file `guide.gz`.

macOS 10.14 (Mojave): Make the unzipped file executable by typing the command "`chmod a+x guide`" in a `Terminal` application in the folder where the file is located. This version requires **Xcode** and **gfortran 8.2** or later to be installed. 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`.

macOS 10.13.6 (High Sierra): Double-click the file `guide.gz` to gunzip it and then make it executable by typing the command "`chmod a+x guide`" in a `Terminal` application in the folder where the file is located. This version requires **Xcode** and **gfortran 5.1** or later to be installed. 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 (where it can be invoked by typing `C:\guide` in a terminal window—see Section 3.1),
2. a folder that contains your data files, or
3. a folder on your search path.

2.2 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

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

After LaTeX is installed, PostScript and pdf versions of a LaTeX file produced by GUIDE can be obtained by typing the following three commands in a **Terminal** (Linux or Mac) or **Command Prompt** (Win) window. (We assume below that the LaTeX file is called `diagram.tex`.)

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`.

Don't use the menu commands of the LaTeX app to process the LaTeX files; type the lines in a terminal window.

In Mac OSX, the file `diagram.ps` may be opened with the *Preview* app, which can convert it to jpg, png, pdf, and other formats. In Windows, the same can be done with the free *ImageMagick* app (https://www.imagemagick.org/). To include the tree diagrams 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. Windows users who do not know how to do this may read http://www.digitalcitizen.life/command-prompt-how-use-basic-commands. Mac OSX users see https://wiredpen.com/resources/basic-unix-commands-for-osx/.

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 x
26 das2d3pc x
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 x
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 **b**oth 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 **C**ategorical variable used for splitting only.

d **D**ependent variable or **d**eath 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 (**d**eath) 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 **f**itting 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 **i**ndicator variables for fitting regression models within nodes.

m **M**issing value flag variable. Each such variable should follow immediately after an **n**, **p** or **s** variable in the description file. Otherwise, the variable is automatically converted to **c**. See Sec. 5 for an example.

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

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

Table 3: Predictor variable role descriptors

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

r Categorical treatment (**R**x) variable used only for fitting the linear models in the nodes of the tree. It is not used for splitting the nodes.

s Numerical-valued variable only used for **s**plitting 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 **T**ime variable, either time to event for proportional hazards models or observation time for longitudinal models.

w **W**eight 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 E**x**cluded variable. Models may be fitted to different subsets of variables by indicating excluded variables in the description file without editing the data file.

z **O**ffset 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 36.2 (Build date: January 8, 2021)
Compiled with GFortran 10.2.0 on macOS Big Sur 11.1
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.
```

Table 4: Demographic and disease category variables

Name	Description and values in parentheses
age	Age (18.04–101.80)
sex	Sex
race	Race
edu	Years of education (0–30)
income	Income
ninsclass	Type of medical insurance (Medicaid, Medicare, Medicare & Medicaid, No insurance, Private, Private & Medicare)
cat1	Primary disease category (ARF, COPD, CHF, cirrhosis, coma, colon cancer, lung cancer, MOSF w/malignancy, MOSF w/sepsis)
cat2	Secondary disease category (same as cat1, 4535 values missing)
ca	Cancer (none, localized, metastatic)

```
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. Variables `dth30` and `death` are 0-1 indicator variables for death within 30 days and 6 months, respectively. Other variables are given in Tables 4–7; six of them, `cat2`, `meanbp1`, `hrt1`, `resp1`, `wtkilo1`, and `urin1`, have 4335, 80, 159, 136, 515, and 3028, respectively, missing values.

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.

Table 5: Admission diagnosis variables

Name	Description
card	Cardiovascular (binary)
gastr	Gastrointestinal (binary)
hema	Hematologic (binary)
meta	Metabolic (binary)
neuro	Neurological (binary)
ortho	Orthopedic (binary)
renal	Renal (binary)
resp	Respiratory (binary)
seps	Sepsis (binary)
trauma	Trauma (binary)
adld3p	Activities of daily living scale Day 3
das2d3pc	Day 3 Duke Activity Status Index

Table 6: Comorbidity illness indicator and outcome variables

Name	Description
amihx	Definite myocardial infarction
cardiohx	Acute MI, vascular disease, severe cardiovascular symptoms
chfhx	Congestive heart failure
chrpulhx	Pulmonary disease
dementhx	Dementia, stroke, Parkinson's
gibledhx	Upper GI bleeding
immunhx	Immunosuppression, organ transplant, HIV, diabetes, connective tissue disease
liverhx	Cirrhosis, hepatic failure
malighx	Solid tumor, metastatic disease, leukemia, myeloma, lymphoma
psychhx	Psychiatric history, psychosis, severe depression
renalhx	Renal disease
death	death within 180 days (censoring indicator)
survtime	Survival time in days

Table 7: Day 1 variables

Name	Description
alb1	Albumin Day 1
aps1	Acute physiology component of APACHE III Day 1 (3–147)
bili1	Bilirubin Day 1 (0.1–58.2)
crea1	Creatinine Day 1 (0.1–25.1)
dnr1	Do-not-resuscitate status Day 1 (binary)
hema1	Hematocrit Day 1 (2–66.19)
hrt1	Heart rate Day 1 (159 missing)
meanbp1	Mean blood pressure Day 1 (80 missing)
paco21	PaCo2 Day 1
pafi1	PaO2/(0.01*FIO2) Day 1 (11.6–937.5)
ph1	PH Day 1
pot1	Potassium Day 1
resp1	Respiratory rate Day 1 (136 missing)
scoma1	Glasgow Coma Score Day 1
sod1	Sodium Day 1
temp1	Temperature Day 1
urin1	urine output Day 1 (3028 missing)
wblc1	White blood cell count Day 1
wtkilo1	Weight Day 1 (515 missing)

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
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 categorical variable values ...
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       3443       13        0        0       20
    #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: 50
 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)
  36.2      (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)
    0.500   (SE number for pruning)
 1          (1=estimated priors, 2=equal priors, 3=other priors)
 1          (1=unit misclassification costs, 2=other)
 2          (1=split point from quantiles, 2=use exhaustive search)
 1          (1=default max. number of split levels, 2=specify no. in next line)
 1          (1=default min. node size, 2=specify min. value in next line)
```

```
 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
20 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
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
     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       3443       13        0        0       20
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       30        0
The above table shows that there are 5735 patient records, of which
3443 contain one or more missing values among the ordinal variables.
No record has missing values in the D variable. (Ordinal variables
are N, F, S, and P, of which there are 0, 0, 20, and 0, respectively.)
In addition, there are 30 C variables. 13 other variables are excluded.

Number of cases used for training: 5735
Number of split variables: 50
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: 0.5000

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
     6  0.1482E+03   neuro
     7  0.1247E+03   alb1
     8  0.1178E+03   card
     9  0.1077E+03   hema1
    10  0.9651E+02   wtkilo1
    11  0.9475E+02   resp
```

```
    12  0.7634E+02   seps
    13  0.7589E+02   cat2
    14  0.6675E+02   bili1
    15  0.6475E+02   dnr1
    16  0.5661E+02   paco21
    17  0.4780E+02   chrpulhx
    18  0.4191E+02   hrt1
    19  0.4063E+02   transhx
    20  0.3675E+02   ninsclas
    21  0.3393E+02   dementhx
    22  0.3110E+02   ph1
    23  0.2956E+02   resp1
    24  0.2602E+02   psychhx
    25  0.2088E+02   income
    26  0.2022E+02   gastr
    27  0.1927E+02   renal
    28  0.1845E+02   cardiohx
    29  0.1630E+02   urin1
    30  0.1563E+02   sod1
    31  0.1469E+02   age
    32  0.1366E+02   malighx
    33  0.1240E+02   wblc1
    34  0.1206E+02   edu
    35  0.1200E+02   ca
    36  0.1168E+02   sex
    37  0.8807E+01   immunhx
    38  0.7795E+01   amihx
    39  0.6616E+01   chfhx
    40  0.6411E+01   gibledhx
    41  0.5011E+01   hema
    42  0.4201E+01   scoma1
    43  0.3175E+01   liverhx
    44  0.3055E+01   pot1
    45  0.1861E+01   temp1
    46  0.1376E+01   renalhx
    47  0.1052E+01   meta
    48  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       72   3.259E-01   6.189E-03   5.481E-03   3.235E-01   5.839E-03
    2       71   3.259E-01   6.189E-03   5.481E-03   3.235E-01   5.839E-03
    :
   38       19   3.179E-01   6.149E-03   5.334E-03   3.191E-01   5.314E-03
   39+      17   3.172E-01   6.145E-03   3.358E-03   3.139E-01   3.477E-03
   40++     12   3.167E-01   6.143E-03   2.739E-03   3.147E-01   2.121E-03
```

```
  41**     10   3.175E-01   6.147E-03   2.273E-03   3.188E-01   3.560E-03
  42        8   3.205E-01   6.162E-03   3.577E-03   3.217E-01   6.541E-03
  43        6   3.229E-01   6.175E-03   3.773E-03   3.249E-01   7.965E-03
  44        5   3.228E-01   6.174E-03   3.471E-03   3.249E-01   5.539E-03
  45        3   3.325E-01   6.221E-03   3.956E-03   3.365E-01   6.220E-03
  46        2   3.751E-01   6.393E-03   4.248E-03   3.801E-01   3.186E-03
  47        1   3.808E-01   6.412E-03   2.782E-04   3.805E-01   4.832E-04
Above shows that the largest tree has 72 terminal nodes.

0-SE tree based on mean is marked with * and has 12 terminal nodes
0-SE tree based on median is marked with + and has 17 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
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 aps1
         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 9: Intermediate node
 A case goes into Node 18 if ninsclas = "No insurance", "Private",
 "Private & Medicare"
 ninsclas mode = "Private"
 Class      Number   Posterior
 NoRHC         225  0.4870E+00
 RHC           237  0.5130E+00
 Number of training cases misclassified = 225
 Predicted class is RHC
 ----------------------------
Node 18: Terminal node
 Class      Number   Posterior
 NoRHC          91  0.3730E+00
 RHC           153  0.6270E+00
 Number of training cases misclassified = 91
 Predicted class is RHC
```

```
 ----------------------------
Node 19: Terminal node
 Class      Number   Posterior
 NoRHC         134  0.6147E+00
 RHC            84  0.3853E+00
 Number of training cases misclassified = 84
 Predicted class is NoRHC
 ----------------------------
Node 5: Terminal node
 Class      Number   Posterior
 NoRHC         350  0.6184E+00
 RHC           216  0.3816E+00
 Number of training cases misclassified = 216
 Predicted class is NoRHC
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if pafi1 <= 142.35938
 pafi1 mean = 211.08630
 Class      Number   Posterior
 NoRHC        2777  0.6853E+00
 RHC          1275  0.3147E+00
 Number of training cases misclassified = 1275
 Predicted class is NoRHC
 ----------------------------
Node 6: Intermediate node
 A case goes into Node 12 if resp = "No"
 resp mode = "Yes"
 Class      Number   Posterior
 NoRHC         667  0.5163E+00
 RHC           625  0.4837E+00
 Number of training cases misclassified = 625
 Predicted class is NoRHC
 ----------------------------
Node 12: Intermediate node
 A case goes into Node 24 if dnr1 = "No"
 dnr1 mode = "No"
 Class      Number   Posterior
 NoRHC         244  0.4200E+00
 RHC           337  0.5800E+00
 Number of training cases misclassified = 244
 Predicted class is RHC
 ----------------------------
Node 24: Intermediate node
 A case goes into Node 48 if cat1 = "ARF", "Lung Cancer", "MOSF w/Malignancy"
 cat1 mode = "ARF"
 Class      Number   Posterior
```

```
 NoRHC         201  0.3903E+00
 RHC           314  0.6097E+00
 Number of training cases misclassified = 201
 Predicted class is RHC
 ----------------------------
Node 48: Terminal node
 Class      Number   Posterior
 NoRHC         151  0.3447E+00
 RHC           287  0.6553E+00
 Number of training cases misclassified = 151
 Predicted class is RHC
 ----------------------------
Node 49: Terminal node
 Class      Number   Posterior
 NoRHC          50  0.6494E+00
 RHC            27  0.3506E+00
 Number of training cases misclassified = 27
 Predicted class is NoRHC
 ----------------------------
Node 25: Terminal node
 Class      Number   Posterior
 NoRHC          43  0.6515E+00
 RHC            23  0.3485E+00
 Number of training cases misclassified = 23
 Predicted class is NoRHC
 ----------------------------
Node 13: Intermediate node
 A case goes into Node 26 if seps = "Yes"
 seps mode = "No"
 Class      Number   Posterior
 NoRHC         423  0.5949E+00
 RHC           288  0.4051E+00
 Number of training cases misclassified = 288
 Predicted class is NoRHC
 ----------------------------
Node 26: Terminal node
 Class      Number   Posterior
 NoRHC          40  0.3636E+00
 RHC            70  0.6364E+00
 Number of training cases misclassified = 40
 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 is obtained by **typing these 3 three lines in the terminal**:

```
> latex file
> dvips file
> ps2pdf file.ps
```

This produces a file named `file.pdf` that can be opened by a pdf program. Note: `file.tex` is the name of the LaTeX file produced by GUIDE. In this example, because the name of the file is `class.tex`, we would type this:

```
> latex class
> dvips class
> ps2pdf class.ps
```

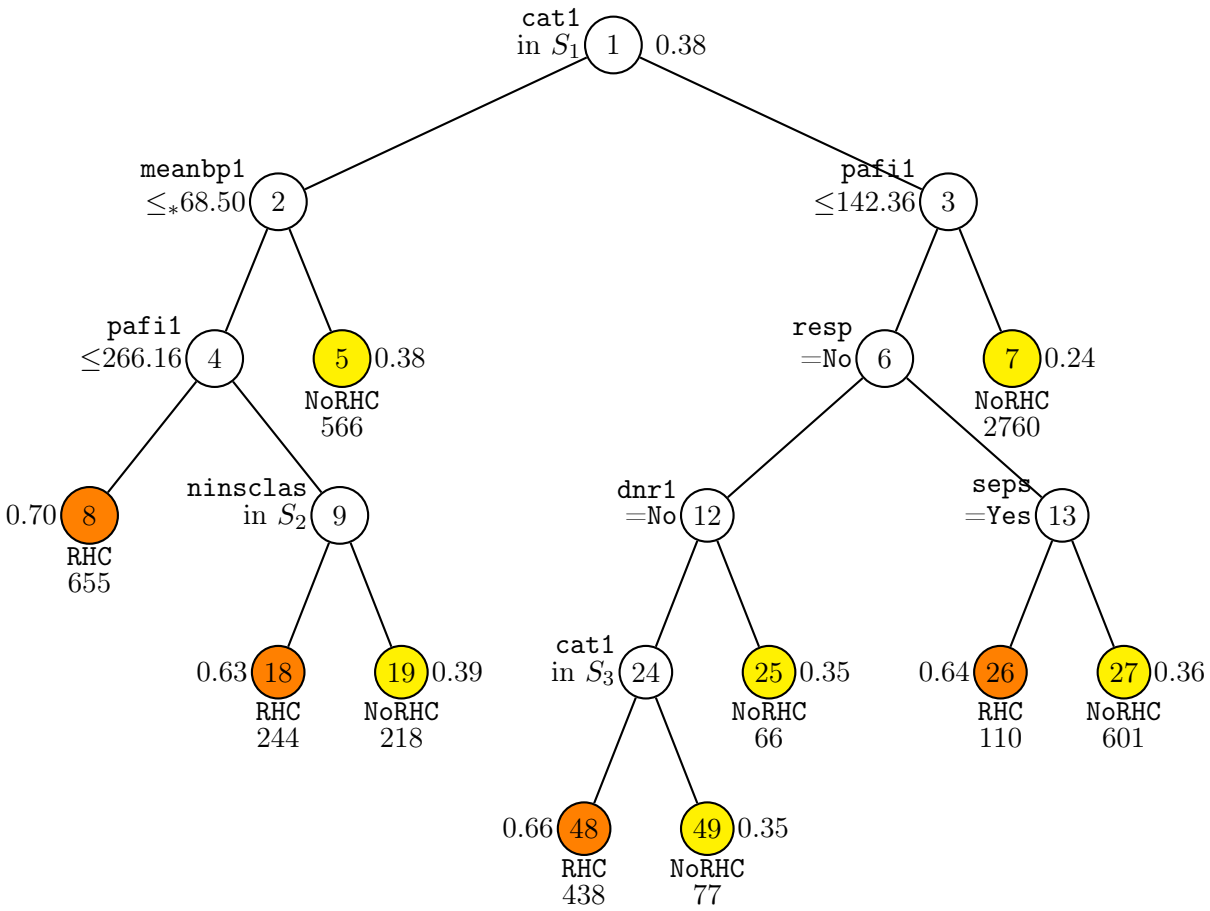

Figure 1: GUIDE v.36.2 0.50-SE classification tree for predicting `swang1` using estimated priors and unit misclassification costs. Tree constructed with 5735 observations. Maximum number of split levels is 15 and minimum node sample size is 57. At each split, an observation goes to the left branch if 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`.

Do not use the menu of LaTeX program to obtain the pdf file!

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)"
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 $(N p_j(t) + \pi_j)/(N+1)$; 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 categorical variable values ...
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       3443       13        0        0       20
    #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: 50
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>=0.50):
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)
  36.2      (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)
     0.500  (SE number for pruning)
 1          (1=estimated priors, 2=equal priors, 3=other priors)
 1          (1=unit misclassification costs, 2=other)
 2          (1=split point from quantiles, 2=use exhaustive search)
 1          (1=default max. number of split levels, 2=specify no. in next line)
 1          (1=default min. node size, 2=specify min. value in next line)
 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
20 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
      :
     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
      :
     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       3443       13        0        0       20
```

```
    #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: 50
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: 0.5000

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
     :
    47  0.1052E+01   meta
    48  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       69   3.128E-01   6.122E-03   7.471E-03   3.043E-01   1.181E-02
   2       68   3.128E-01   6.122E-03   7.471E-03   3.043E-01   1.181E-02
   :
  35       11   3.085E-01   6.099E-03   5.294E-03   3.054E-01   7.208E-03
  36+       8   3.065E-01   6.088E-03   5.477E-03   3.023E-01   8.055E-03
  37**      7   3.051E-01   6.080E-03   4.316E-03   3.043E-01   4.803E-03
  38        5   3.175E-01   6.147E-03   4.664E-03   3.188E-01   7.916E-03
  39        3   3.261E-01   6.190E-03   4.740E-03   3.240E-01   8.099E-03
  40        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 7 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.

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
          4T    1174     1174    RHC            3.705E-01 resp1 +urin1
          5T     509      509    NoRHC          3.340E-01 resp1 +aps1
          3     4052     4052    NoRHC          3.147E-01 pafi1 +crea1
          6     1992     1992    NoRHC          4.538E-01 meanbp1 +paco21
         12     1220     1220    RHC            4.549E-01 resp
         24T     642      642    RHC            3.894E-01 dnr1
         25      578      578    NoRHC          4.723E-01 pafi1 +scoma1
         50T      77       77    RHC            2.208E-01 -
         51T     501      501    NoRHC          4.251E-01 malighx
         13T     772      772    NoRHC          3.096E-01 resp
          7T    2060     2060    NoRHC          1.801E-01 cat2

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 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: RHC
  Node 2: 0.24316737 * pafi1 + meanbp1 > 153.28329
    Node 5: NoRHC
Node 1: cat1 /= "CHF", "MOSF w/Sepsis"
  Node 3: -63.408853 * crea1 + pafi1 <= 88.542610
    Node 6: 2.9786426 * paco21 + meanbp1 <= 201.01756 or NA
      Node 12: resp = "No"
        Node 24: RHC
      Node 12: resp /= "No"
        Node 25: -0.18672165 * scoma1 + pafi1 <= 61.304990
          Node 50: RHC
        Node 25: -0.18672165 * scoma1 + pafi1 > 61.304990 or NA
          Node 51: NoRHC
```

```
     Node 6: 2.9786426 * paco21 + meanbp1 > 201.01756
       Node 13: NoRHC
   Node 3: -63.408853 * crea1 + pafi1 > 88.542610 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 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: Terminal node
 Class      Number   Posterior
 NoRHC         435  0.3705E+00
 RHC           739  0.6295E+00
 Number of training cases misclassified = 435
 Predicted class is RHC
 ----------------------------
Node 5: Terminal node
 Class      Number   Posterior
 NoRHC         339  0.6660E+00
 RHC           170  0.3340E+00
 Number of training cases misclassified = 170
 Predicted class is NoRHC
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if -63.408853 * crea1 + pafi1 <= 88.542610
 Linear combination mean = 90.778616
 Class      Number   Posterior
 NoRHC        2777  0.6853E+00
```

```
 RHC          1275  0.3147E+00
 Number of training cases misclassified = 1275
 Predicted class is NoRHC
 ----------------------------
Node 6: Intermediate node
 A case goes into Node 12 if 2.9786426 * paco21 + meanbp1 <= 201.01756
 Linear combination mean = 195.51588
 Class      Number   Posterior
 NoRHC        1088  0.5462E+00
 RHC           904  0.4538E+00
 Number of training cases misclassified = 904
 Predicted class is NoRHC
 ----------------------------
Node 12: Intermediate node
 A case goes into Node 24 if resp = "No"
 resp mode = "No"
 Class      Number   Posterior
 NoRHC         555  0.4549E+00
 RHC           665  0.5451E+00
 Number of training cases misclassified = 555
 Predicted class is RHC
 ----------------------------
Node 24: Terminal node
 Class      Number   Posterior
 NoRHC         250  0.3894E+00
 RHC           392  0.6106E+00
 Number of training cases misclassified = 250
 Predicted class is RHC
 ----------------------------
Node 25: Intermediate node
 A case goes into Node 50 if -0.18672165 * scoma1 + pafi1 <= 61.304990
 Linear combination mean = 122.16910
 Class      Number   Posterior
 NoRHC         305  0.5277E+00
 RHC           273  0.4723E+00
 Number of training cases misclassified = 273
 Predicted class is NoRHC
 ----------------------------
Node 50: Terminal node
 Class      Number   Posterior
 NoRHC          17  0.2208E+00
 RHC            60  0.7792E+00
 Number of training cases misclassified = 17
 Predicted class is RHC
 ----------------------------
Node 51: Terminal node
```

```
 Class      Number   Posterior
 NoRHC         288  0.5749E+00
 RHC           213  0.4251E+00
 Number of training cases misclassified = 213
 Predicted class is NoRHC
 ----------------------------
Node 13: Terminal node
 Class      Number   Posterior
 NoRHC         533  0.6904E+00
 RHC           239  0.3096E+00
 Number of training cases misclassified = 239
 Predicted class is NoRHC
 ----------------------------
Node 7: Terminal node
 Class      Number   Posterior
 NoRHC        1689  0.8199E+00
 RHC           371  0.1801E+00
 Number of training cases misclassified = 371
 Predicted class is NoRHC
 ----------------------------

 Classification matrix for training sample:
 Predicted      True class
 class          NoRHC       RHC
 NoRHC           2849       993
 RHC              702      1191
 Total           3551      2184

 Number of cases used for tree construction: 5735
 Number misclassified: 1695
 Resubstitution estimate of mean misclassification cost: 0.29555362

 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 `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.

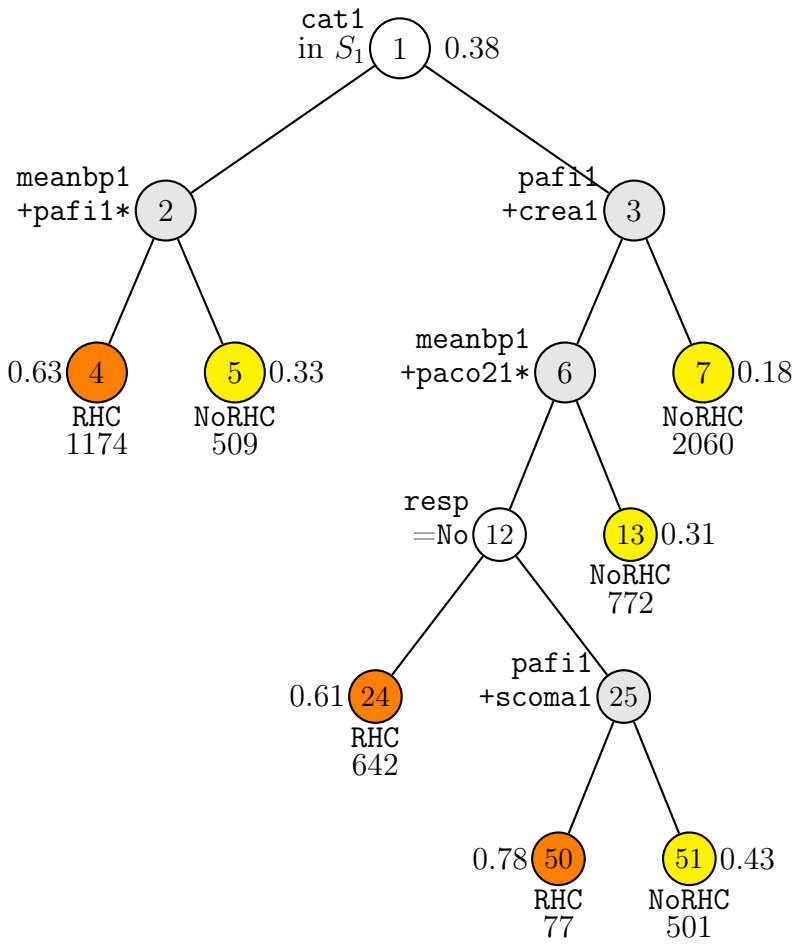

Figure 2: GUIDE v.36.2 0.50-SE classification tree for predicting `swang1` using linear split priority, estimated priors and unit misclassification costs. Tree constructed with 5735 observations. Maximum number of split levels is 15 and minimum node sample size is 57. At each split, an observation goes to the left branch if and only if the condition is satisfied. 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`}. Intermediate nodes in lightgray indicate linear splits. 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`.

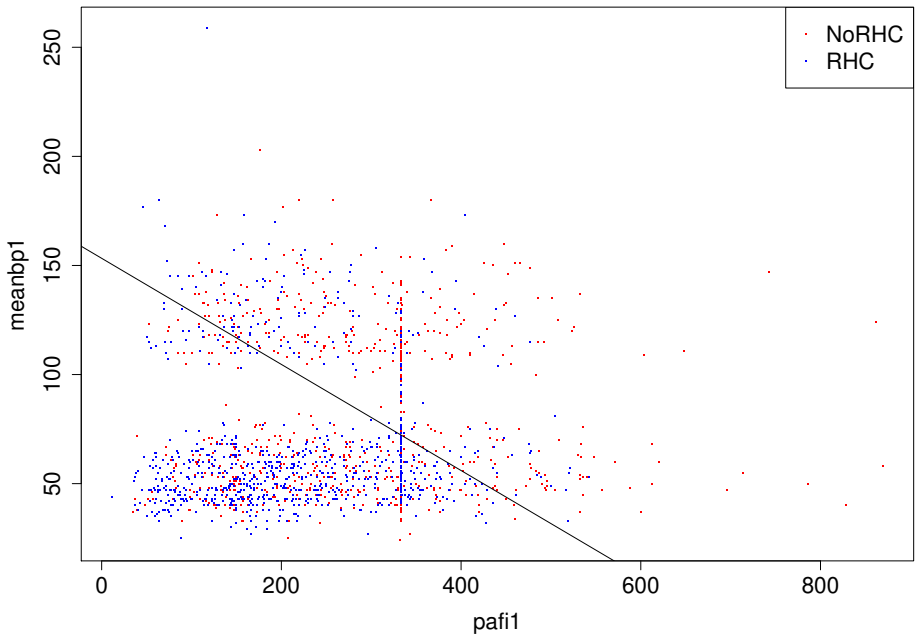

Figure 3: Plot of `meanbp1` vs `pafi1` for data and split in node 2 of tree in Figure 2

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$node == 4 | z1$node == 5
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 C variable values ...
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       3443       13        0        0       20
    #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: 50
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>=0.50):
Choose 1 for estimated priors, 2 for equal priors, 3 for priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
Choose a split point selection method for numerical variables:
Choose 1 to use faster method based on sample quantiles
Choose 2 to use exhaustive search
Input 1 or 2 ([1:2], <cr>=2):
Default max. number of split levels: 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
20 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
     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
     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       3443       13        0        0       20
    #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: 50
 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: 0.5000

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       75   3.308E-01   6.213E-03   5.577E-03   3.290E-01   4.869E-03
   2       72   3.308E-01   6.213E-03   5.577E-03   3.290E-01   4.869E-03
   :
  43       11   3.198E-01   6.159E-03   5.199E-03   3.185E-01   6.850E-03
  44+       9   3.187E-01   6.153E-03   5.082E-03   3.168E-01   6.371E-03
  45*       6   3.173E-01   6.146E-03   5.205E-03   3.173E-01   7.361E-03
  46**      5   3.187E-01   6.153E-03   5.474E-03   3.182E-01   6.676E-03
  47++      3   3.212E-01   6.166E-03   4.637E-03   3.191E-01   6.892E-03
  48        2   3.224E-01   6.172E-03   4.503E-03   3.211E-01   5.039E-03
  49        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 6 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

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            3.910E-01 resp1 +resp1 +pafi1
          4T     426      426    RHC            2.653E-01 wtkilo1 +wtkilo1 +pafi1
          5     1257     1257    NoRHC          4.145E-01 meanbp1 +meanbp1 +alb1
         10T     924      924    RHC            3.939E-01 ninsclas +ninsclas +malighx
         11T     333      333    NoRHC          3.093E-01 cardiohx +cardiohx +pafi1
          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: 5
 Total number of nodes of final tree: 9
 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: resp1 <= 17.000000 or NA
     Node 4: Mean cost = 0.26525822
   Node 2: resp1 > 17.000000
     Node 5: meanbp1 <= 98.500000 or NA
       Node 10: Mean cost = 0.39393939
     Node 5: meanbp1 > 98.500000
       Node 11: Mean cost = 0.30930931
 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 resp1 <= 17.000000 or NA
 resp1 mean = 27.996990
 pafi1 mean = 249.20858
                                    Bandwidth
 Class      Number   Posterior   resp1   pafi1   Correlation
 NoRHC         774  0.4599E+00   6.8704E+00   7.6307E+01    -0.1967
```

```
 RHC           909  0.5401E+00   8.3817E+00   6.8628E+01    -0.0456
 Number of training cases misclassified = 658
 If node model is inapplicable due to missing values, predicted class is "RHC"
 ----------------------------
Node 4: Terminal node
 wtkilo1 mean = 78.773147
 pafi1 mean = 239.16476
                                    Bandwidth
 Class      Number   Posterior   wtkilo1   pafi1   Correlation
 NoRHC         130  0.3052E+00   1.8063E+01   8.6994E+01    -0.3144
 RHC           296  0.6948E+00   1.5322E+01   8.1718E+01    -0.1371
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if meanbp1 <= 98.500000 or NA
 meanbp1 mean = 73.641200
 alb1 mean = 3.1195305
                                    Bandwidth
 Class      Number   Posterior   meanbp1   alb1   Correlation
 NoRHC         644  0.5123E+00   2.4785E+01   4.0669E-01     0.1433
 RHC           613  0.4877E+00   1.1827E+01   5.1338E-01     0.0126
 Number of training cases misclassified = 521
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------
Node 10: Terminal node
 ninsclas mode = Private
 malighx mode = 0
 Class      Number   Posterior
 NoRHC         414  0.4481E+00
 RHC           510  0.5519E+00
 ----------------------------
Node 11: Terminal node
 cardiohx mode = 0
 pafi1 mean = 257.20148
                                  Bandwidth
 Class      Number   Posterior   pafi1
 NoRHC         230  0.6907E+00   1.0393E+02
 RHC           103  0.3093E+00   1.2153E+02
 ----------------------------
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           2942      1086
 RHC              609      1098
 Total           3551      2184

 Number of cases used for tree construction: 5735
 Number misclassified: 1695
 Resubstitution estimate of mean misclassification cost: 0.29555362

 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 5 terminal nodes and a cross-validation estimate of misclassification cost of 0.3187. 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"
```

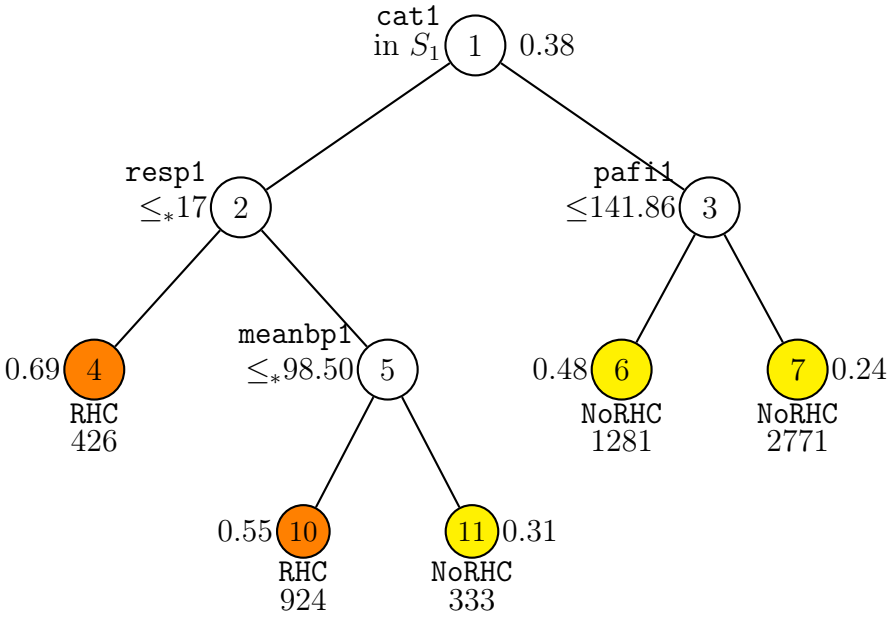

Figure 4: GUIDE v.36.2 0.50-SE classification tree for predicting `swang1` using bivariate kernel discriminant node models, estimated priors and unit misclassification costs. Tree constructed with 5735 observations. Maximum number of split levels is 15 and minimum node sample size is 57. At each split, an observation goes to the left branch if and only if the condition is satisfied. 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`.

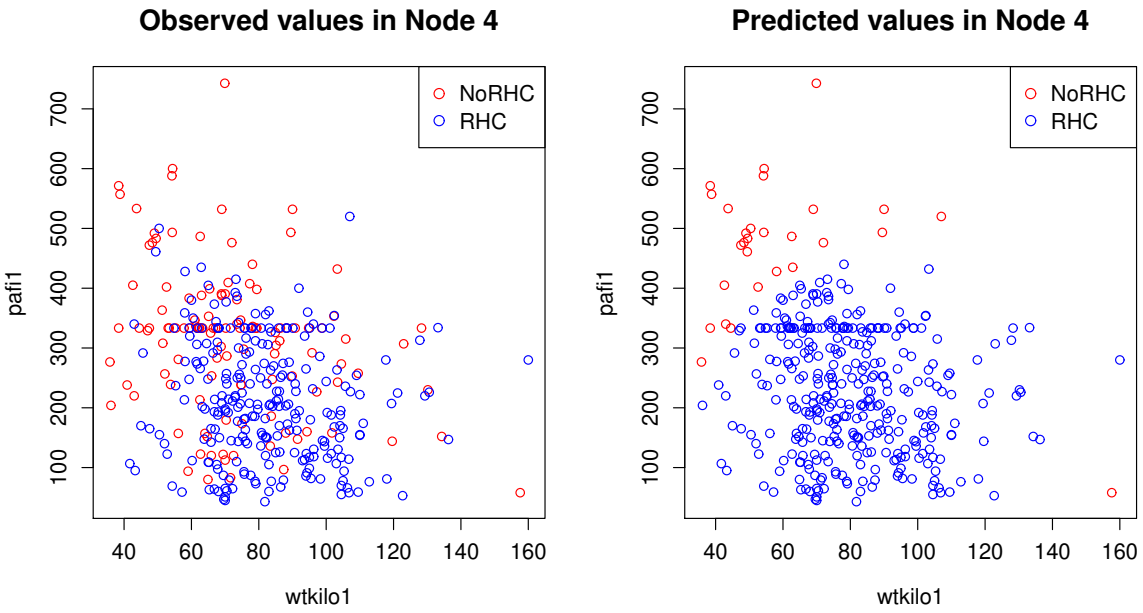

Figure 5: Plots of observed and predicted values for data in node 4 of tree in Figure 4

```
y    10     0.37705      0.62295    "RHC"     "RHC"
y     7     0.60626      0.39374    "RHC"     "NoRHC"
y     7     0.77436      0.22564    "NoRHC"   "NoRHC"
y    10     0.39935      0.60065    "RHC"     "RHC"
```

Figure 5 shows plots of the data and the predicted values in terminal node 4 of the tree in the space of variables `wtkilo1` and `pafi1` selected by GUIDE (see the information for these terminal nodes in `ker2.out`). The R code for making the plot is below.

```
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$node == 4
x <- z0$wtkilo1[gp]
y <- z0$pafi1[gp]
classv <- z0$swang1[gp]
plot(x,y,xlab="wtkilo1",ylab="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 4")
plot(x,y,xlab="wtkilo1",ylab="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 4")
```

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 C variable values ...
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       3443       13        0        0       20
    #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: 50
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>=0.50):
Choose 1 for estimated priors, 2 for equal priors, 3 for priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
Choose a split point selection method for numerical variables:
Choose 1 to use faster method based on sample quantiles
Choose 2 to use exhaustive search
Input 1 or 2 ([1:2], <cr>=2):
Default max. number of split levels: 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
20 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		
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	
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       3443       13        0        0       20
    #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: 50
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: 0.5000

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       75   3.304E-01   6.211E-03   5.654E-03   3.290E-01   5.183E-03
   2       72   3.304E-01   6.211E-03   5.654E-03   3.290E-01   5.183E-03
   :
  46       13   3.275E-01   6.197E-03   5.237E-03   3.226E-01   5.366E-03
  47++     10   3.231E-01   6.175E-03   4.384E-03   3.223E-01   3.883E-03
  48        7   3.329E-01   6.223E-03   5.216E-03   3.310E-01   6.184E-03
  49        6   3.318E-01   6.218E-03   5.029E-03   3.293E-01   5.110E-03
  50**      5   3.247E-01   6.183E-03   3.937E-03   3.278E-01   5.475E-03
  51        4   3.283E-01   6.201E-03   4.812E-03   3.296E-01   6.122E-03
  52        3   3.287E-01   6.203E-03   6.373E-03   3.304E-01   7.470E-03
  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 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

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
          2T    1683     1683    RHC            3.179E-01 resp1 +resp1 +pafi1
          3     4052     4052    NoRHC          2.848E-01 pafi1 +pafi1 +crea1
          6T    1281     1281    NoRHC          3.052E-01 aps1 +aps1 +resp1
          7     2771     2771    NoRHC          2.317E-01 meanbp1 +meanbp1 +crea1
         14     1456     1456    NoRHC          2.740E-01 cat2 +cat2 +crea1
         28T     386      386    NoRHC          2.798E-01 wtkilo1 +wtkilo1 +ph1
         29T    1070     1070    NoRHC          2.467E-01 crea1 +crea1 +wtkilo1
         15T    1315     1315    NoRHC          1.612E-01 hema1 +hema1 +card

Number of terminal nodes of final tree: 5
Total number of nodes of final tree: 9
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: Mean cost = 0.31788473
Node 1: cat1 /= "CHF", "MOSF w/Sepsis"
  Node 3: pafi1 <= 141.85938
    Node 6: Mean cost = 0.30523029
  Node 3: pafi1 > 141.85938 or NA
    Node 7: meanbp1 <= 69.500000 or NA
      Node 14: cat2 = "Colon Cancer", "MOSF w/Sepsis"
        Node 28: Mean cost = 0.27979275
      Node 14: cat2 /= "Colon Cancer", "MOSF w/Sepsis"
        Node 29: Mean cost = 0.24672897
    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: Terminal node
 Number of nearest neighbors = 8
 resp1 mean = 27.996990 SD = 12.437032
 pafi1 mean = 249.20858 SD = 115.80759
               correlation = -0.11546553
 Class      Number   Posterior
 NoRHC         774  0.4599E+00
 RHC           909  0.5401E+00
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if pafi1 <= 141.85938
 Number of nearest neighbors = 9
 pafi1 mean = 211.08630 SD = 107.82115
 crea1 mean = 1.8973326 SD = 1.8532752
               correlation = 0.56202754E-001
 Class      Number   Posterior
 NoRHC        2777  0.6853E+00
 RHC          1275  0.3147E+00
 Number of training cases misclassified = 1154
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------
Node 6: Terminal node
 Number of nearest neighbors = 8
 aps1 mean = 60.373927 SD = 17.920707
 resp1 mean = 30.854487 SD = 13.984748
               correlation = 0.37400675
 Class      Number   Posterior
 NoRHC         661  0.5160E+00
 RHC           620  0.4840E+00
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if meanbp1 <= 69.500000 or NA
 Number of nearest neighbors = 8
 meanbp1 mean = 85.416758 SD = 37.728426
 crea1 mean = 1.8756021 SD = 1.8750908
```

```
               correlation = -0.69337494E-001
 Class      Number   Posterior
 NoRHC        2116  0.7636E+00
 RHC           655  0.2364E+00
 Number of training cases misclassified = 642
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------
Node 14: Intermediate node
 A case goes into Node 28 if cat2 = "Colon Cancer", "MOSF w/Sepsis"
 Number of nearest neighbors = 8
 cat2 mode = NA
 crea1 mean = 2.0729392
 Class      Number   Posterior
 NoRHC        1013  0.6957E+00
 RHC           443  0.3043E+00
 Number of training cases misclassified = 399
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------
Node 28: Terminal node
 Number of nearest neighbors = 6
 wtkilo1 mean = 71.459000 SD = 18.499387
 ph1 mean = 7.3607852 SD = 0.12537065
               correlation = 0.18360358E-001
 Class      Number   Posterior
 NoRHC         211  0.5466E+00
 RHC           175  0.4534E+00
 ----------------------------
Node 29: Terminal node
 Number of nearest neighbors = 7
 crea1 mean = 1.8170665 SD = 1.6590547
 wtkilo1 mean = 72.277453 SD = 19.080371
               correlation = 0.11163209
 Class      Number   Posterior
 NoRHC         802  0.7495E+00
 RHC           268  0.2505E+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           3043      1008
RHC              508      1176
Total           3551      2184

Number of cases used for tree construction: 5735
Number misclassified: 1516
Resubstitution estimate of mean misclassification cost: 0.26434176

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. The row with two asterisks (**) in the output file `nn2.out` shows that the tree has 5 terminal nodes and a cross-validation estimate of misclassification cost of 0.3248. 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 21 of the tree in the space of variables `resp1` and `pafi1` selected by GUIDE (see the information for these terminal nodes in `nn2.out`).

File `nn2.fit` gives the terminal node number and observed and predicted classes of each observation in the data file. Below are the few 10 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       6    "NoRHC"  "RHC"
    y       2    "RHC"    "NoRHC"
    y      28    "RHC"    "NoRHC"
    y      29    "NoRHC"  "NoRHC"
    y       2    "RHC"    "RHC"
    y      15    "NoRHC"  "NoRHC"
    y      29    "NoRHC"  "NoRHC"
    y      15    "NoRHC"  "NoRHC"
    y      29    "NoRHC"  "NoRHC"
```

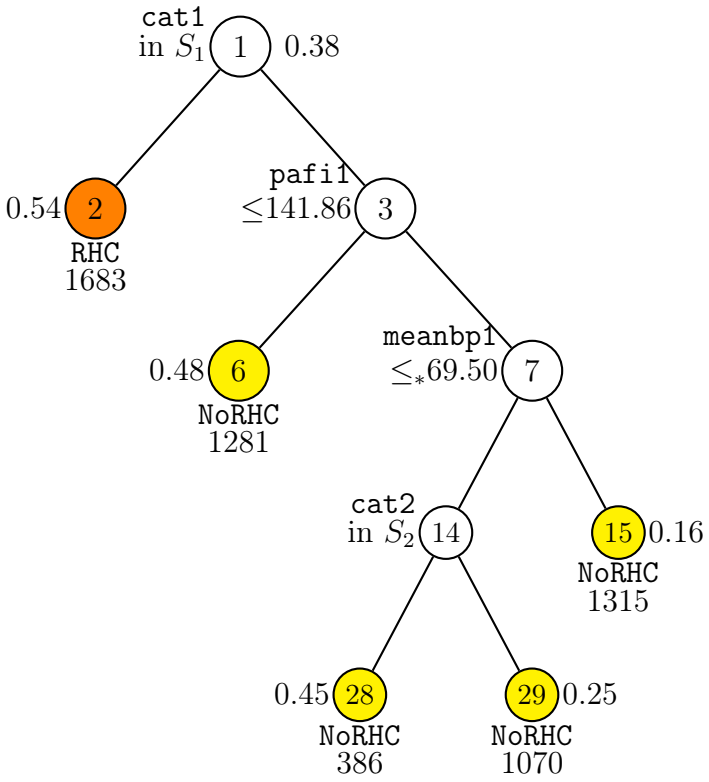

Figure 6: GUIDE v.36.2 0.50-SE classification tree for predicting `swang1` using bivariate nearest-neighbor node models, estimated priors and unit misclassification costs. Tree constructed with 5735 observations. Maximum number of split levels is 15 and minimum node sample size is 57. At each split, 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`}. Set S_2 = {`Colon Cancer`, `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`.

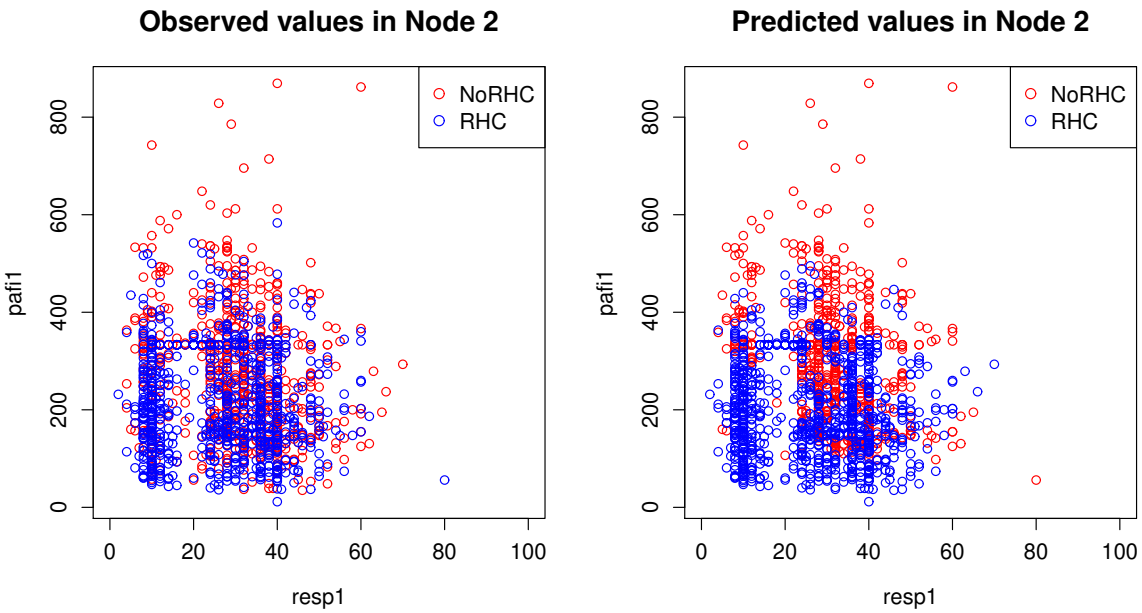

Figure 7: Plots of observed and predicted values for data in node 2 of tree in Figure 6

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_ ≠ 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
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
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
TEXTILPQ	Household textiles last quarter
TOBACCPQ	Tobacco and smoking supplies 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, ..., c)
3. New CHILDAGE $\leq_* c$ (equivalent to original CHILDAGE = 0, 1, ..., 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. Following are the results.

```
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
423 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
      5  AGE2       s    22.00        87.00                 1903
      6  AGE2_      m                                 1
      :
     50  FINLWT21   w    1351.       0.7027E+05
     51  FJSSDEDX   s    0.000       0.3042E+05
     52  FJSS_EDX   m                                 0
      :
    514  INTRDVX_   d                                 3
    522  IRAB       s    1.000        6.000                 4514
    523  IRAB_      m                                 2
      :
    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: 0.5000

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
     4  0.2981E+03   ERANKH
     5  0.2881E+03   POV_CY
     6  0.2881E+03   POV_PY
     7  0.2796E+03   RETSURVX
     8  0.2607E+03   FEDRFNDX
     9  0.2460E+03   RETSURVB
    10  0.2460E+03   RETSRVBX
   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
```

3	73	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
4	72	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
5	71	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
6	70	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
7	69	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
8	68	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
9	67	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
10	65	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
11	64	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
12	63	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
13	62	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
14	61	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
15	60	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
16	58	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
17	57	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
18	56	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
19	54	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
20	53	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
21	52	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
22	51	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
23	49	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
24	47	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
25	46	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
26	45	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
27	43	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
28	42	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
29	41	3.060E-01	6.727E-03	6.920E-03	3.067E-01	7.374E-03
30	40	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 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
R code is stored in ceclass.r
```

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 ≤ 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 ≤ 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
```

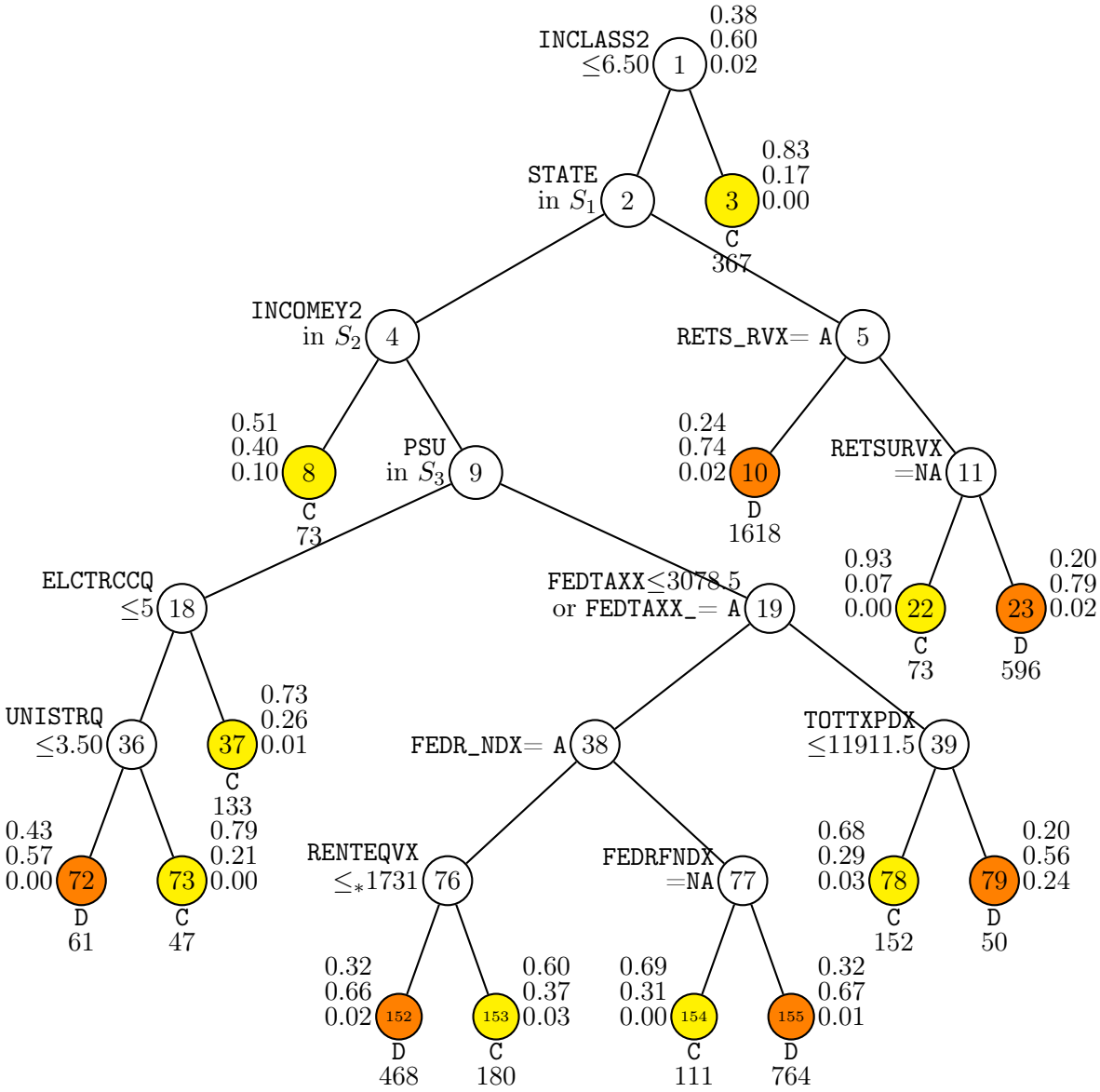

Figure 8: GUIDE v.36.0 0.50-SE classification tree for predicting `INTRDVX_` using estimated priors and unit misclassification costs. Tree constructed with 4693 observations. Maximum number of split levels is 14 and minimum node sample size is 46. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ 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`.

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 used by designating them as `P` in the description file. There can be multiple `P` variables in the same data set. Unlike the other types of variables, each line in the description file containing a `P` variable must have the value of its period (e.g., 360 for angular measurements, 24 for hour of day, 7 for day of week, and 12 for month of year) immediately after `P` on the same line.

We demonstrate this with the files `nhtsadata.csv` and `nhtsaclass.dsc`, which are obtained from vehicle crash test results from the National Highway Transportation Safety Administration (NHTSA) (www-nrd.nhtsa.dot.gov/database/veh/). The variable `HIC` is the head injury criterion, which measures the severity of head injury. For this illustration, we construct a classification tree with equal priors to predict the dichotomized variable `HIC2`, which equals 1 if `HIC` > 999, and equals 0 otherwise. Many experts believe that HIC > 999 is absolutely life threatening.

The contents of `nhtsaclass.dsc` are partially reproduced here:

```
nhtsadata.csv
NA
2
1 TSTNO x
2 BARRIG c
3 BARSHP c
4 BARANG p 360
:
:
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
```

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.

Because the data are from a designed experiment, the class proportions of `HIC2` = `0` and `1` are not necessarily representative of those in real accidents. If we knew

Table 13: Some variable definitions for NHTSA data

Variable	Meaning
BARSHP	barrier shape
BX8	distance from rear surface of vehicle to upper trailing edge of right door
BX12	distance from rear surface of vehicle to bottom of a post of right side
COLMEC	steering column collapse mechanism
HH	distance from head to windshield header
HR	distance from head to header to side of occupant
IMPANG	impact angle
MODELD	vehicle model
OCCAGE	dummy occupant age
OCCTYP	dummy occupant type
PDOF	principal direction of force
YEAR	vehicle model year

the class prior probabilities in real accidents, we can use them to build a model for predicting HIC2. Since we do not know the true class priors, we instead use *equal priors* to build a model to estimate the class probabilities for the experimental data. The result is **not** a prediction model, but a model for estimating $P(\texttt{HIC2} = 1)$, as in logistic regression. Following are the steps to construct an input file for equal priors.

6.1 Input file creation

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

```
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: 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
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 ...
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
Assigning integer codes to C variable values ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Class  #Cases     Proportion
0        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
Default number of cross-validations:           10
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Best tree may be chosen based on mean or median CV estimate
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50):
Choose 1 for estimated priors, 2 for equal priors, 3 for priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1): 2
this is where equal priors are chosen
```

```
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: 13
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 32
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): equalp.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):
Input 0 for #errors, 1 for class proportions, 2 for nothing ([0:2], <cr>=1):
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save 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
      :
    144  RSTTOR     c                                2
    145  RSTUNK     c                                3
    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: 0.5000

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
     4  0.2918E+03   OCCAGE
     5  0.2738E+03   HS
     6  0.2193E+03   RSTDPL
     7  0.1758E+03   CTRL2
     8  0.1644E+03   HR
     9  0.1504E+03   OCCHT
    10  0.1490E+03   OCCWT
     :
    85  0.4655E+00   BARANG
    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

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 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
```

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
```

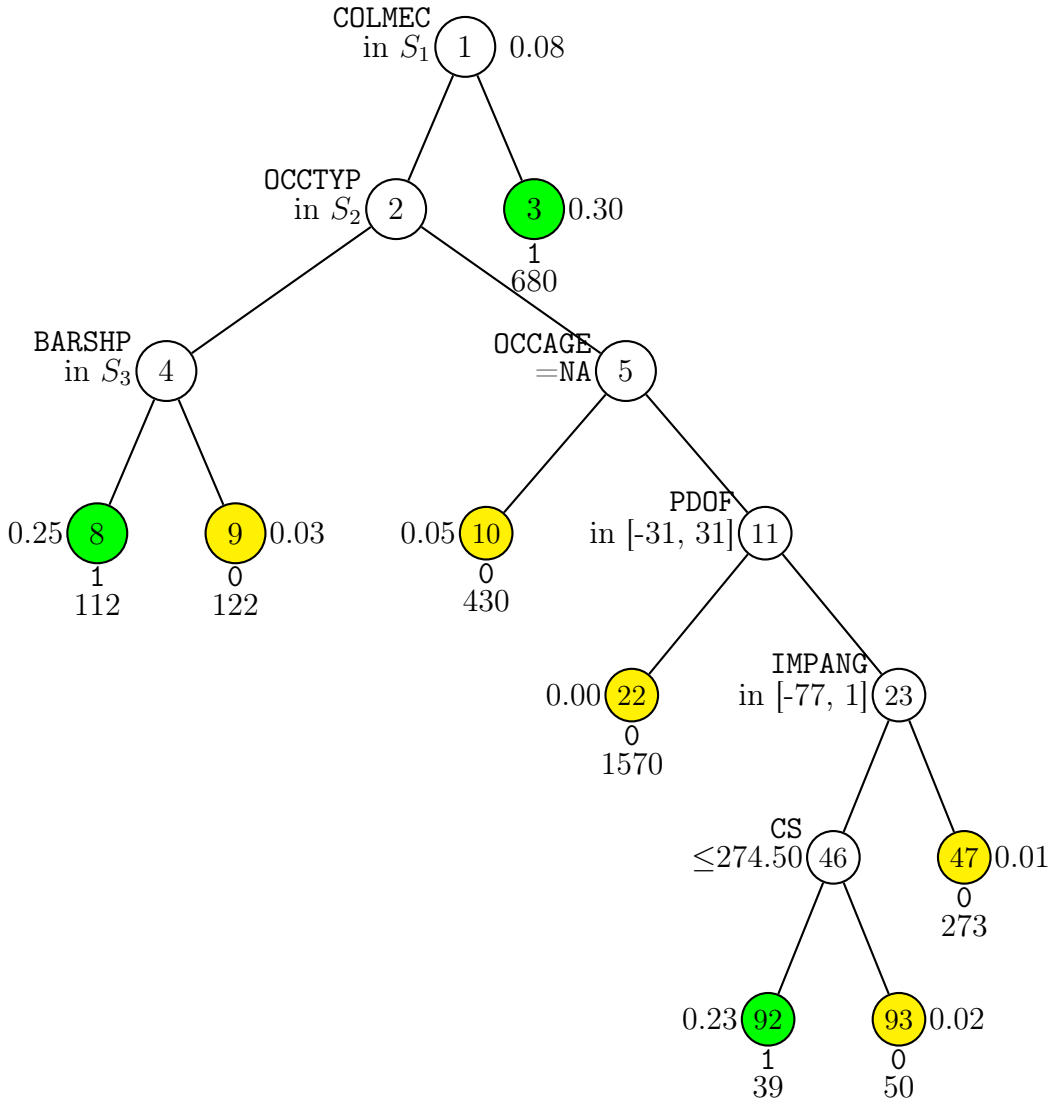

Figure 9: GUIDE v.36.2 0.50-SE classification tree for predicting HIC2 using equal priors and unit misclassification costs. Tree constructed with 3276 observations (excluding observations with non-positive weight or missing values in D, T, R or Z variables). Maximum number of split levels is 13 and minimum 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.

```
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 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>=0): 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
423 N variables changed to S
Dependent 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 ...
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 C variable values ...
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: variable MISC2PQ is constant
Warning: variable MISC2CQ is constant
Warning: variable TCARTRKP is constant
Warning: variable TCARTRKC is constant
Warning: variable TOTHVHRP is constant
```

```
Warning: variable TOTHVHRC is constant
Warning: variable VMISCHEP is constant
Warning: variable VMISCHEC is constant
Warning: variable ROTHRFLP is constant
Warning: variable ROTHRFLC is constant
Warning: variable WELFREBX has all values missing
Warning: variable LUMPSUMB has all values missing
Warning: variable LMPSUMBX has all values missing
Warning: variable OTHRINCB is constant
Warning: variable OTRINCBX has all values missing
Warning: variable NETRENTB is constant
Warning: variable NETRNTBX is constant
Warning: variable OTHLONBX is constant
Warning: 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       15        0        0      423
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      172        0       41        0
Weight variable FINLWT21 in column: 50
Number of cases used for training: 2922
Number of split variables: 464
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):
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 ([1:464], <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
      5  AGE2       s    22.00        87.00                 1225
      6  AGE2_      m                                 1
      :
     50  FINLWT21   w    1351.       0.7027E+05
      :
    507  FSMPFRMX   s  -0.4000E+06   0.1090E+07
    508  FSMP_RMX   m                                 0
    513  INTRDVX    d    1.000       0.9834E+05
    522  IRAB       s    1.000        6.000                2826
    523  IRAB_      m                                 2
      :
    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: 0.5000

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
     5  0.1033E+03   RENTEQVX
     6  0.7999E+02   INCOMEY1
     7  0.7380E+02   INCNONW1
     8  0.7014E+02   EARNCOMP
     9  0.6979E+02   INC_HRS1
    10  0.6821E+02   AGE2
     :
   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

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
         76T    283      283     1  6.941E+03  4.538E+12   FFTAXOWE
         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: 10
Total number of nodes of final tree: 19
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: INTRDVX-mean = 6940.8765
           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 9: Intermediate node
 A case goes into Node 18 if RENTEQVX <= 1261.5000 or NA
 RENTEQVX mean = 1352.5146
 ----------------------------
Node 18: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      3046.        7.164       0.1704E-11
 INTRDVX mean = 3046.33
 ----------------------------
Node 19: Intermediate node
 A case goes into Node 38 if EMRTPNOP <= 3.1665000
 EMRTPNOP mean = 596.07431
 ----------------------------
Node 38: Intermediate node
 A case goes into Node 76 if FFTAXOWE <= 10182.500
 FFTAXOWE mean = 11903.193
 ----------------------------
Node 76: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      6941.        6.638       0.1626E-09
 INTRDVX mean = 6940.88
 ----------------------------
Node 77: Intermediate node
 A case goes into Node 154 if FJSSDEDX <= 3557.5000
```

```
 FJSSDEDX mean = 7258.1343
 ----------------------------
Node 154: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.3739E+05    5.446       0.2058E-05
 INTRDVX mean = 37391.5
 ----------------------------
Node 155: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      9204.        3.832       0.2336E-03
 INTRDVX mean = 9204.11
 ----------------------------
Node 39: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      4371.        5.330       0.1825E-06
 INTRDVX mean = 4371.06
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if STATE = "1", "12", "15", "2", "31", "48", "49", "51", "53"
 STATE mode = "6"
 ----------------------------
Node 10: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      2796.        2.451       0.2031E-01
 INTRDVX mean = 2796.30
 ----------------------------
Node 11: Intermediate node
 A case goes into Node 22 if STATE = "17", "24", "25", "36", "6"
 STATE mode = "6"
 ----------------------------
Node 22: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      5393.        3.090       0.2762E-02
 INTRDVX mean = 5393.32
 ----------------------------
Node 23: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.6450E+05    7.033       0.9796E-07
 INTRDVX mean = 64504.4
 ----------------------------
```

```
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: 0.2848

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

In the above results, the pruned tree is marked with two asterisks (tree #64). It has 9 terminal nodes and a cross-validation estimate of prediction mean squared error of 4.297E+12. Figure 10 shows the tree. The first split is on amount of stocks, with `STOCKX` $\leq$ \$1,048,495 or missing going to node 2 (in the tree diagram, the symbol "$\leq_*$" stands for "$\leq$ or missing"). Node 3 consists of 14 observations with a mean `INTRDVX` of \$85,803. 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 \$2,895.36.

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
```

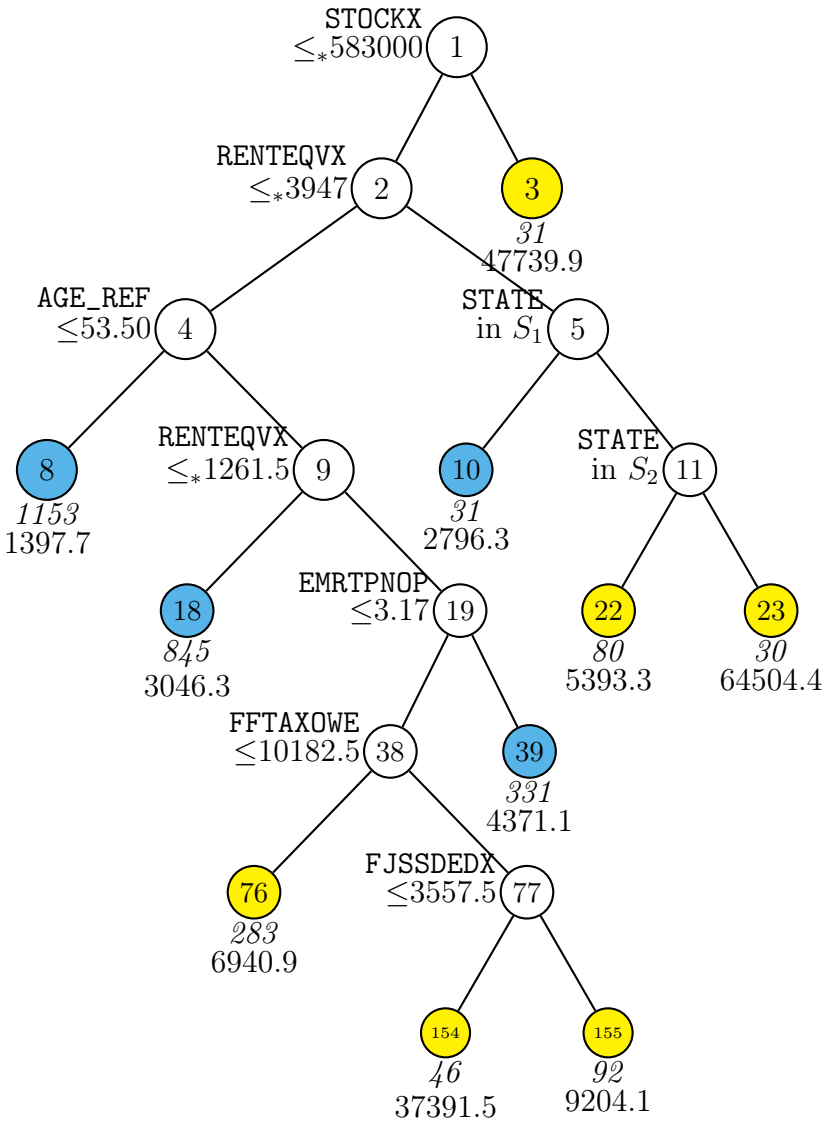

Figure 10: GUIDE v.36.2 0.50-SE piecewise constant weighted least-squares regression tree for predicting `INTRDVX`. Tree constructed with 2922 observations (excluding observations with non-positive weight or missing values in `D`, `T`, `R` or `Z` variables). Maximum number of split levels is 12 and minimum node sample size is 29. 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 $S_1 = \{$`1`, `12`, `15`, `2`, `31`, `48`, `49`, `51`, `53`$\}$. Set $S_2 = \{$`17`, `24`, `25`, `36`, `6`$\}$. 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`.

```
1. Create a GUIDE input file
Input your choice: 1
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=proportional hazards,
 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>=0): 2
Input 1 for default options, 2 otherwise ([1:2], <cr>=1): 2
option 2 allows more choices, including storage of selected variables
and their regression coefficients below
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
Dependent 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 ...
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 C variable values ...
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: variable OTHRINCB is constant
Warning: variable NETRENTB is constant
Warning: variable NETRNTBX is constant
Warning: variable OTHLONBX is constant
Warning: 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       15      423        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      172        0       41        0
Weight variable FINLWT21 in column: 50
Number of cases used for training: 2922
Number of split variables: 464
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>=0.50):
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
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 29
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 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: 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 ([1:464], <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
```

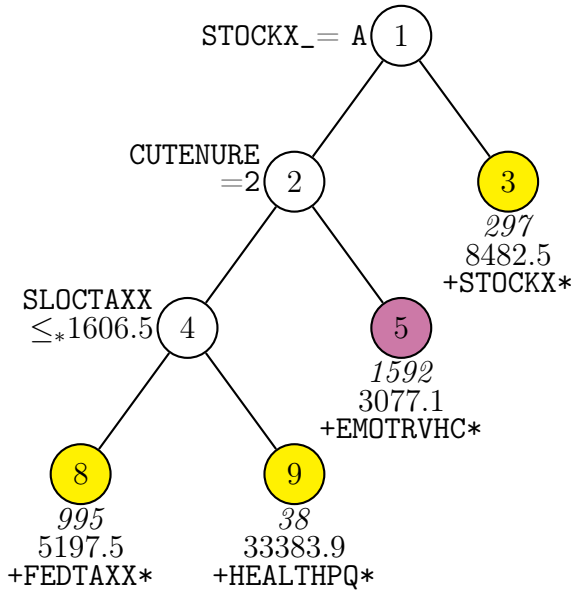

Figure 11: GUIDE v.36.2 0.50-SE piecewise simple linear weighted least-squares regression tree for predicting INTRDVX. Tree constructed with 2922 observations (excluding observations with non-positive weight or missing values in D, T, R or Z variables). Maximum number of split levels is 12 and minimum node sample size is 29. 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'. Sample size (*in italics*), mean of INTRDVX, and sign and name of regressor variable printed below nodes, with colors distinguishing between positive and negative signs. Terminal nodes with means above and below value of 4696.6 at root node are colored yellow and purple, respectively. An asterisk at end of name of regressor indicates its slope is significant at the 0.05 level (unadjusted for post selection). 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

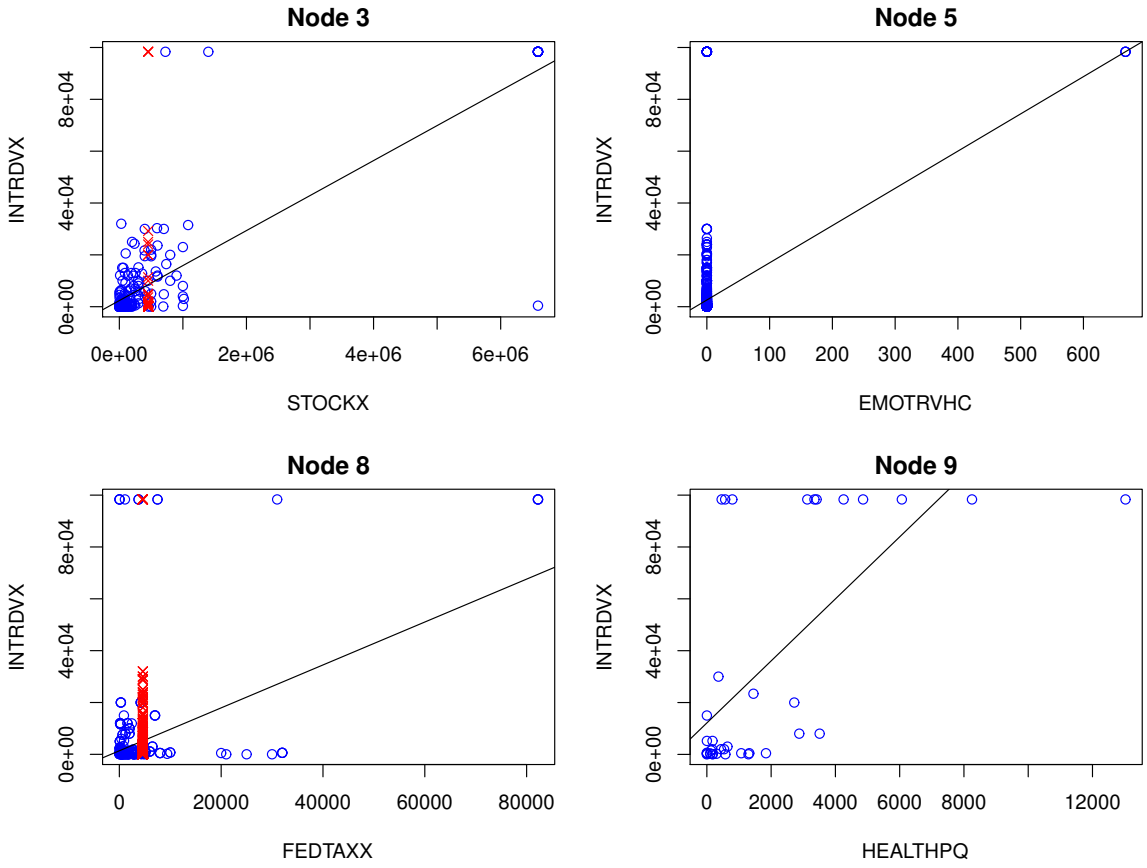

Figure 12: Plots of data and regression lines in terminal nodes of tree in Figure 11. Red colored points are imputed with node means.

```
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=proportional hazards,
 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
Dependent 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 ...
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 C variable values ...
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: variable OTHRINCB is constant
Warning: variable NETRENTB is constant
Warning: variable NETRNTBX is constant
Warning: variable OTHLONBX is constant
Warning: 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       15      423        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      172        0       41        0
Weight variable FINLWT21 in column: 50
Number of cases used for training: 2922
Number of split variables: 464
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):
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 ([1:464], <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: variable WELFREBX has all values missing
Warning: variable LUMPSUMB has all values missing
Warning: variable LMPSUMBX has all values missing
Warning: variable OTHRINCB is constant
Warning: variable NETRENTB is constant
Warning: variable NETRNTBX is constant
Warning: variable OTHLONBX is constant
Warning: 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
      :
    507  FSMPFRMX   n  -0.4000E+06   0.1090E+07
```

```
    508  FSMP_RMX   m                                    0
    513  INTRDVX    d    1.000       0.9834E+05
    522  IRAB       n   1.0000E+00    6.000                    2826
    523  IRAB_      m                                    2
      :
    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       27      412        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      171        0       41        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: .5000

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
     4  0.8419E+02   NETRENTX
     5  0.7993E+02   FRRETIRX
     6  0.7857E+02   INCNONW1
     :
   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
```

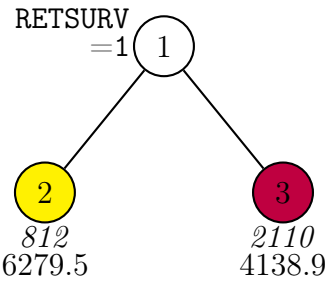

Figure 14: GUIDE v.36.2 0.50-SE piecewise linear weighted least-squares regression tree with stepwise variable selection for predicting INTRDVX. Tree constructed with 2922 observations (excluding observations with non-positive weight or missing values in D, T, R or Z variables). Maximum number of split levels is 12 and minimum node sample size is 416. 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.

```
INTRDVX mean = 4138.86
Predicted values truncated at 1.00000 & 98338.0
----------------------------
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 0.90-quantile regression model.

8.1 Piecewise constant: one quantile

8.1.1 Input file creation

```
Choose type of regression model:
1=linear, 2=quantile, 3=Poisson, 4=proportional hazards,
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):
Input quantile probability ([0.00:1.00], <cr>=0.50): 0.90
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
423 N variables changed to S
Dependent 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 ...
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 C variable values ...
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: variable OTHRINCB is constant
Warning: variable NETRENTB is constant
Warning: variable NETRNTBX is constant
Warning: variable OTHLONBX is constant
Warning: 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       15        0        0      423
```

```
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      172        0       41        0
Number of cases used for training: 2922
Number of split variables: 464
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: All positive weights treated as one
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): 2
Input file name: quantcon.r
Input rank of top variable to split root node ([1:464], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < quantcon.in
```

Contents of quantcon.out

```
Quantile regression tree with quantile probability 0.9000
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
412 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: variable OTHRINCB is constant
Warning: variable NETRENTB is constant
Warning: variable NETRNTBX is constant
Warning: variable OTHLONBX is constant
Warning: 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
      :
    507  FSMPFRMX   s  -0.4000E+06   0.1090E+07
    508  FSMP_RMX   m                                 0
    513  INTRDVX    d    1.000       0.9834E+05
    522  IRAB       s    1.000        6.000                  2826
    523  IRAB_      m                                 2
      :
    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: 0.5000

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  0.1348E+03   STOCKX
     2  0.1241E+03   STOCKYRX
     :
   410  0.4948E-03   SMLAPPPQ
```

```
   411  0.1448E-03   TOTHENTP

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       74   4.299E+07   3.267E+06   3.522E+06   4.467E+07   4.946E+06
   2       72   4.299E+07   3.267E+06   3.522E+06   4.467E+07   4.947E+06
   :
  21*      47   4.277E+07   3.267E+06   3.471E+06   4.386E+07   4.605E+06
   :
  34       23   4.365E+07   3.284E+06   3.370E+06   4.476E+07   4.942E+06
  35**     22   4.379E+07   3.290E+06   3.356E+06   4.410E+07   4.755E+06
  36       21   4.479E+07   3.330E+06   3.667E+06   4.416E+07   4.442E+06
  37       20   4.497E+07   3.333E+06   3.717E+06   4.416E+07   4.698E+06
  38       18   4.504E+07   3.333E+06   3.693E+06   4.417E+07   4.617E+06
  39       17   4.504E+07   3.333E+06   3.693E+06   4.417E+07   4.617E+06
  40       16   4.512E+07   3.336E+06   3.718E+06   4.417E+07   4.679E+06
  41       13   4.538E+07   3.395E+06   3.948E+06   4.530E+07   5.729E+06
  42++     12   4.532E+07   3.462E+06   4.129E+06   4.124E+07   6.796E+06
  43        8   5.197E+07   4.372E+06   4.284E+06   5.679E+07   7.524E+06
  44        7   5.532E+07   4.684E+06   3.471E+06   5.880E+07   4.430E+06
  45        3   5.821E+07   4.797E+06   4.532E+06   5.880E+07   4.494E+06
  46        2   6.407E+07   5.118E+06   3.455E+06   6.663E+07   4.771E+06
  47        1   7.071E+07   5.380E+06   2.648E+06   7.004E+07   1.779E+06

0-SE tree based on mean is marked with * and has 47 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

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  9.500E+03  STOCKX
         2      2891     2891     1  8.000E+03  AGE_REF
         4       990      990     1  1.000E+03  STATE
         8       347      347     1  3.400E+03  INCLASS2
        16T      174      174     1  1.500E+03  STATE
        17       173      173     1  8.000E+03  SLOCTAXX
```

```
 34T     141      141     1  5.000E+03  FEDTAXX
 35T      32       32     1  9.834E+04  -
  9T     643      643     1  5.000E+02  PROPTXPQ
  5     1901     1901     1  1.300E+04  STATE
 10       85       85     1  9.834E+04  FINCATAX
 20T      56       56     1  2.100E+04  -
 21T      29       29     1  9.834E+04  -
 11     1816     1816     1  1.000E+04  FINCATAX
 22     1585     1585     1  8.000E+03  INC_HRS1
 44      852      852     1  1.200E+04  FFTAXOWE
 88      399      399     1  4.000E+03  PSU
176T      30       30     1  2.000E+04  -
177T     369      369     1  3.156E+03  INCLASS
 89      453      453     1  2.300E+04  INCNONW2
178      305      305     1  1.800E+04  PSU
356T      39       39     1  3.200E+04  -
357      266      266     1  1.200E+04  STATE
714T      51       51     1  2.500E+04  -
715T     215      215     1  9.912E+03  EARNCOMP
179      148      148     1  3.000E+04  FINCBTAX
358T     113      113     1  2.200E+04  BUILDING
359T      35       35     1  9.834E+04  -
 45      733      733     1  2.700E+03  STOCKYRX
 90      685      685     1  2.200E+03  STATE
180      126      126     1  1.300E+04  PROPTXCQ
360T      95       95     1  6.800E+03  INC_HRS2
361T      31       31     1  9.834E+04  -
181T     559      559     1  1.400E+03  INCWEEK1
 91T      48       48     1  1.200E+04  -
 23      231      231     1  9.834E+04  INCNONW1
 46      186      186     1  2.370E+04  STATE
 92T      29       29     1  9.834E+04  -
 93      157      157     1  1.300E+04  HEALTHCQ
186T     126      126     1  5.000E+03  RETPENPQ
187T      31       31     1  1.700E+04  -
 47T      45       45     1  9.834E+04  -
  3T      31       31     1  9.834E+04  -
Warning: tree very large, omitting node numbers in LaTeX file

Number of terminal nodes of final tree: 22
Total number of nodes of final tree: 43
Second best split variable (based on curvature test) at root node is 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: AGE_REF <= 49.500000
    Node 4: STATE = "10", "11", "15", "18", "2", "22", "23", "25", "26", "4",
             "41", "48", "53", "6", "8"
      Node 8: INCLASS2 <= 5.5000000
        Node 16: INTRDVX sample quantile = 1500.0000
      Node 8: INCLASS2 > 5.5000000 or NA
        Node 17: SLOCTAXX = NA & SLOC_AXX = "A"
          Node 34: INTRDVX sample quantile = 5000.0000
        Node 17: not (SLOCTAXX = NA & SLOC_AXX = "A")
          Node 35: INTRDVX sample quantile = 98338.000
    Node 4: STATE /= "10", "11", "15", "18", "2", "22", "23", "25", "26", "4",
             "41", "48", "53", "6", "8"
      Node 9: INTRDVX sample quantile = 500.00000
  Node 2: AGE_REF > 49.500000 or NA
    Node 5: STATE = "32", "45", "53", "54", "8"
      Node 10: FINCATAX <= 102940.50
        Node 20: INTRDVX sample quantile = 21000.000
      Node 10: FINCATAX > 102940.50 or NA
        Node 21: INTRDVX sample quantile = 98338.000
    Node 5: STATE /= "32", "45", "53", "54", "8"
      Node 11: FINCATAX <= 162276.00
        Node 22: INC_HRS1 <= 4.5000000 or NA
          Node 44: FFTAXOWE <= 18.000000
            Node 88: PSU = "1211", "1319", "1320", "1422", "1424"
              Node 176: INTRDVX sample quantile = 20000.000
            Node 88: PSU /= "1211", "1319", "1320", "1422", "1424"
              Node 177: INTRDVX sample quantile = 3156.0000
          Node 44: FFTAXOWE > 18.000000 or NA
            Node 89: INCNONW2 = "4"
                    or (INCNONW2 = NA & INCN_NW2 = "A")
              Node 178: PSU = "1109", "1110", "1207", "1318", "1422", "1424", "1429"
                Node 356: INTRDVX sample quantile = 32000.000
              Node 178: PSU /= "1109", "1110", "1207", "1318", "1422", "1424", "1429"
                Node 357: STATE = "12", "15", "39", "4", "41", "51"
                  Node 714: INTRDVX sample quantile = 25000.000
                Node 357: STATE /= "12", "15", "39", "4", "41", "51"
                  Node 715: INTRDVX sample quantile = 9912.0000
            Node 89: INCNONW2 /= "4"
                    & not (INCNONW2 = NA & INCN_NW2 = "A")
              Node 179: FINCBTAX <= 114502.00
                Node 358: INTRDVX sample quantile = 22000.000
              Node 179: FINCBTAX > 114502.00 or NA
                Node 359: INTRDVX sample quantile = 98338.000
        Node 22: INC_HRS1 > 4.5000000
          Node 45: STOCKYRX <= 22500.000  or STOCKYRX = NA & STOC_YRX = "A"
```

```
            Node 90: STATE = "22", "23", "26", "34", "41", "48", "55"
              Node 180: PROPTXCQ <= 370.83335
                Node 360: INTRDVX sample quantile = 6800.0000
              Node 180: PROPTXCQ > 370.83335 or NA
                Node 361: INTRDVX sample quantile = 98338.000
            Node 90: STATE /= "22", "23", "26", "34", "41", "48", "55"
              Node 181: INTRDVX sample quantile = 1400.0000
          Node 45: not (STOCKYRX <= 22500.000 or STOCKYRX = NA & STOC_YRX = "A")
            Node 91: INTRDVX sample quantile = 12000.000
      Node 11: FINCATAX > 162276.00 or NA
        Node 23: INCNONW1 = NA & INCN_NW1 = "A"
          Node 46: STATE = "1", "13", "26", "33", "34", "39", "47"
            Node 92: INTRDVX sample quantile = 98338.000
          Node 46: STATE /= "1", "13", "26", "33", "34", "39", "47"
            Node 93: HEALTHCQ <= 1127.6667
              Node 186: INTRDVX sample quantile = 5000.0000
            Node 93: HEALTHCQ > 1127.6667 or NA
              Node 187: INTRDVX sample quantile = 17000.000
        Node 23: not (INCNONW1 = NA & INCN_NW1 = "A")
          Node 47: INTRDVX sample quantile = 98338.000
 Node 1: STOCKX > 583000.00
   Node 3: INTRDVX sample quantile = 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 STOCKX <= 583000.00 or NA
 STOCKX mean = 453208.43
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if AGE_REF <= 49.500000
 AGE_REF mean = 55.254691
 ----------------------------
```

```
Node 4: Intermediate node
 A case goes into Node 8 if STATE = "10", "11", "15", "18", "2", "22", "23", "25", "26", "4",
 "41", "48", "53", "6", "8"
 STATE mode = "NA"
 ----------------------------
Node 8: Intermediate node
 A case goes into Node 16 if INCLASS2 <= 5.5000000
 INCLASS2 mean = 5.0028349
 ----------------------------
Node 16: Terminal node
 ----------------------------
Node 17: Intermediate node
 A case goes into Node 34 if SLOCTAXX = NA & SLOC_AXX = "A"
 SLOCTAXX mean = 5486.0864
 ----------------------------
Node 34: Terminal node
 ----------------------------
Node 35: Terminal node
 ----------------------------
Node 9: Terminal node
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if STATE = "32", "45", "53", "54", "8"
 STATE mode = "NA"
 ----------------------------
Node 10: Intermediate node
 A case goes into Node 20 if FINCATAX <= 102940.50
 FINCATAX mean = 108996.84
 ----------------------------
Node 20: Terminal node
 ----------------------------
Node 21: Terminal node
 ----------------------------
Node 11: Intermediate node
 A case goes into Node 22 if FINCATAX <= 162276.00
 FINCATAX mean = 86037.575
 ----------------------------
Node 22: Intermediate node
 A case goes into Node 44 if INC_HRS1 <= 4.5000000 or NA
 INC_HRS1 mean = 38.574691
 ----------------------------
Node 44: Intermediate node
 A case goes into Node 88 if FFTAXOWE <= 18.000000
 FFTAXOWE mean = 2731.9918
 ----------------------------
Node 88: Intermediate node
```

```
 A case goes into Node 176 if PSU = "1211", "1319", "1320", "1422", "1424"
 PSU mode = "NA"
 ----------------------------
Node 176: Terminal node
 ----------------------------
Node 177: Terminal node
 ----------------------------
Node 89: Intermediate node
 A case goes into Node 178 if INCNONW2 = "4"
    or INCNONW2 = NA & INCN_NW2 =  "A"
 INCN_NW2 mode = "A"
 ----------------------------
Node 178: Intermediate node
 A case goes into Node 356 if PSU = "1109", "1110", "1207", "1318", "1422", "1424", "1429"
 PSU mode = "NA"
 ----------------------------
Node 356: Terminal node
 ----------------------------
Node 357: Intermediate node
 A case goes into Node 714 if STATE = "12", "15", "39", "4", "41", "51"
 STATE mode = "NA"
 ----------------------------
Node 714: Terminal node
 ----------------------------
Node 715: Terminal node
 ----------------------------
Node 179: Intermediate node
 A case goes into Node 358 if FINCBTAX <= 114502.00
 FINCBTAX mean = 82411.017
 ----------------------------
Node 358: Terminal node
 ----------------------------
Node 359: Terminal node
 ----------------------------
Node 45: Intermediate node
 A case goes into Node 90 if STOCKYRX <= 22500.000 or STOC_YRX = "A"
 STOCKYRX mean = 75844.837
 ----------------------------
Node 90: Intermediate node
 A case goes into Node 180 if STATE = "22", "23", "26", "34", "41", "48", "55"
 STATE mode = "NA"
 ----------------------------
Node 180: Intermediate node
 A case goes into Node 360 if PROPTXCQ <= 370.83335
 PROPTXCQ mean = 271.03601
 ----------------------------
```

```
Node 360: Terminal node
 ----------------------------
Node 361: Terminal node
 ----------------------------
Node 181: Terminal node
 ----------------------------
Node 91: Terminal node
 ----------------------------
Node 23: Intermediate node
 A case goes into Node 46 if INCNONW1 = NA & INCN_NW1 = "A"
 INCN_NW1 mode = "A"
 ----------------------------
Node 46: Intermediate node
 A case goes into Node 92 if STATE = "1", "13", "26", "33", "34", "39", "47"
 STATE mode = "NA"
 ----------------------------
Node 92: Terminal node
 ----------------------------
Node 93: Intermediate node
 A case goes into Node 186 if HEALTHCQ <= 1127.6667
 HEALTHCQ mean = 652.57851
 ----------------------------
Node 186: Terminal node
 ----------------------------
Node 187: Terminal node
 ----------------------------
Node 47: 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 sample 0.90-quantile are given beneath each terminal node.

8.2 Simple linear

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
```

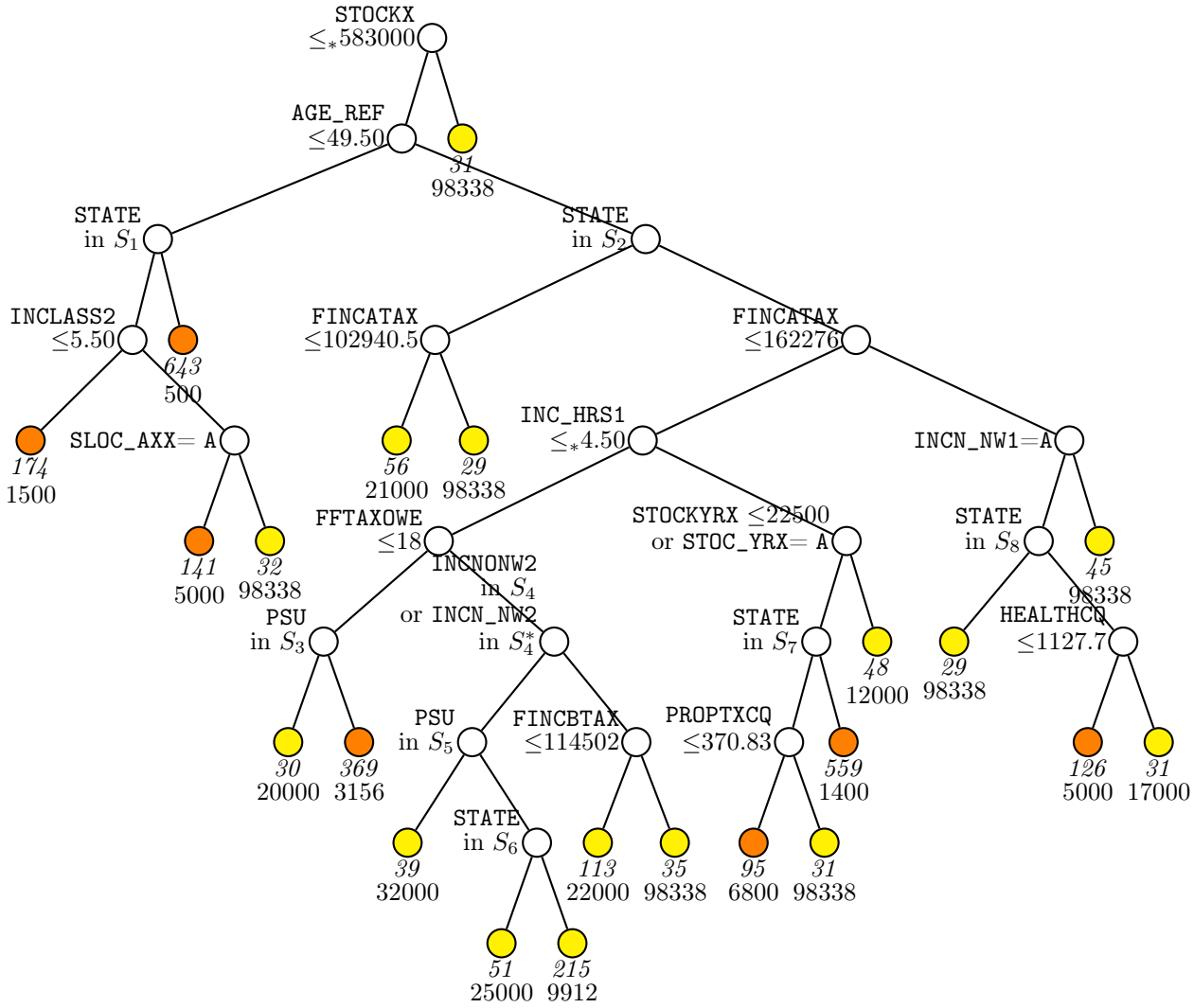

Figure 15: GUIDE v.36.2 0.50-SE piecewise constant 0.900-quantile regression tree for predicting `INTRDVX`. Tree constructed with 2922 observations (excluding observations with non-positive weight or missing values in `D`, `T`, `R` or `Z` variables). Maximum number of split levels is 12 and minimum node sample size is 29. 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 = {`10`, `11`, `15`, `18`, `2`, `22`, `23`, `25`, `26`, `4`, `41`, `48`, `53`, `6`, `8`}. Set S_2 = {`32`, `45`, `53`, `54`, `8`}. Set S_3 = {`1211`, `1319`, `1320`, `1422`, `1424`}. Set S_4 = {`4`}; S_4^* = {`A`}. Set S_5 = {`1109`, `1110`, `1207`, `1318`, `1422`, `1424`, `1429`}. Set S_6 = {`12`, `15`, `39`, `4`, `41`, `51`}. Set S_7 = {`22`, `23`, `26`, `34`, `41`, `48`, `55`}. Set S_8 = {`1`, `13`, `26`, `33`, `34`, `39`, `47`}. Sample size (*in italics*) and 0.900-quantile of `INTRDVX` printed below nodes. Terminal nodes with quantiles above and below value of 9500.0 at root node are colored yellow and orange, respectively. Second best split variable at root node is `STOCKYRX`.

```
File quantlin.in exists
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=proportional hazards,
 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
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
Dependent 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 ...
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 C variable values ...
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
:
:
```

```
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       15      423        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      172        0       41        0
Number of cases used for training: 2922
Number of split variables: 464
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: All positive weights treated as one
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): quantlin.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: quantlin.r
Input rank of top variable to split root node ([1:464], <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: variable OTHRINCB is constant
Warning: variable NETRENTB is constant
Warning: variable NETRNTBX is constant
```

```
Warning: variable OTHLONBX is constant
Warning: 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
      :
    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       27      412        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      171        0       41        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: .5000

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  0.1527E+03   STOCKX
```

```
     2  0.1405E+03   STOCKYRX
     :
   389  0.1395E-02   TOTHENTP
   390  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       78   3.909E+07   3.185E+06   3.312E+06   3.641E+07   3.818E+06
   2       77   3.909E+07   3.185E+06   3.312E+06   3.642E+07   3.818E+06
   :
  18       58   3.898E+07   3.185E+06   3.311E+06   3.639E+07   3.799E+06
  19*      56   3.896E+07   3.185E+06   3.306E+06   3.639E+07   3.771E+06
  20       55   3.900E+07   3.184E+06   3.306E+06   3.647E+07   3.772E+06
   :
  34       29   3.946E+07   3.205E+06   3.603E+06   3.653E+07   3.900E+06
  35+      23   3.947E+07   3.205E+06   3.611E+06   3.636E+07   3.839E+06
  36       22   3.958E+07   3.219E+06   3.672E+06   3.653E+07   3.812E+06
  37       20   3.958E+07   3.219E+06   3.672E+06   3.653E+07   3.812E+06
  38       19   3.956E+07   3.219E+06   3.674E+06   3.653E+07   3.822E+06
  39**     18   4.026E+07   3.228E+06   3.745E+06   3.714E+07   4.374E+06
  40       16   4.092E+07   3.236E+06   3.665E+06   3.735E+07   4.543E+06
  41       15   4.110E+07   3.238E+06   3.631E+06   3.735E+07   4.353E+06
  42++     14   4.150E+07   3.252E+06   4.038E+06   3.657E+07   4.755E+06
  43       13   4.199E+07   3.192E+06   3.276E+06   3.937E+07   4.874E+06
   :
  51        3   6.321E+07   5.016E+06   3.154E+06   6.488E+07   5.080E+06
  52        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 56 terminal nodes
0-SE tree based on median is marked with + and has 23 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as -- tree

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

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

D-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      29       29     2  3.000E+04  -
    143T     126      126     2  1.200E+04  MARITAL1
      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  OTHLODPQ
     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", "45", "48", "8"
              Node 142: INTRDVX sample quantile = 30000.000
            Node 71: STATE /= "13", "23", "32", "41", "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: OTHLODPQ <= 55.000000
      Node 12: INTRDVX sample quantile = 13000.000
    Node 6: OTHLODPQ > 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     3098.8
 STOCKX      0.14459E-01   25.000      0.45321E+06  0.65867E+07
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if FINCATAX <= 114116.50
 FINCATAX mean = 92406.011
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if INCNONW1 = "1", "5", "6"
 INCN_NW1 mode = "A"
 ----------------------------
Node 8: Intermediate node
 A case goes into Node 16 if FFTAXOWE <= 45.000000
 FFTAXOWE mean = 1991.5327
 ----------------------------
Node 16: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     2028.9
 ETOTALP     0.34622       730.23       7403.9      0.27817E+06
 ----------------------------
Node 17: Intermediate node
 A case goes into Node 34 if FRRETIRX <= 833.50000
 FRRETIRX mean = 17330.447
```

```
 ----------------------------
Node 34: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant    -5592.3
 FINCBTAX     1.0569       11000.       52294.      0.11710E+06
 ----------------------------
Node 35: Intermediate node
 A case goes into Node 70 if FRRETIRX <= 19265.000
 FRRETIRX mean = 20455.957
 ----------------------------
Node 70: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     5155.1
 FEDRFNDX     2.9806       17.000       2159.6       14279.
 ----------------------------
Node 71: Intermediate node
 A case goes into Node 142 if STATE = "13", "23", "32", "41", "45", "48", "8"
 STATE mode = "42"
 ----------------------------
Node 142: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     20000.
 TVRDIOCQ     67.568       0.0000       62.052       262.00
 ----------------------------
Node 143: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     593.48
 FEDTAXX      3.2582       80.000       3173.6       31396.
 ----------------------------
Node 9: Intermediate node
 A case goes into Node 18 if STATE = "11", "15", "2", "23", "25", "26", "41", "48", "53", "8"
 STATE mode = "NA"
 ----------------------------
Node 18: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     3950.0
 HLFBATHQ     10450.       0.0000      0.41022       2.0000
 ----------------------------
Node 19: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
```

```
 Constant     379.18
 PROPTXPQ     2.8997       0.0000       314.20       3870.0
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if STATE = "18", "22", "26", "32", "33", "34", "45", "54", "8"
 STATE mode = "NA"
 ----------------------------
Node 10: Intermediate node
 A case goes into Node 20 if HIGH_EDU <= 15.500000
 HIGH_EDU mean = 15.298749
 ----------------------------
Node 20: Intermediate node
 A case goes into Node 40 if OCCUCOD1 = "10", "2", "3", "4"
 OCCUCOD1 mode = "3"
 ----------------------------
Node 40: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     30000.
 TRANSCQ      42.446       0.0000       721.21       4500.0
 ----------------------------
Node 41: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     13277.
 MEDSUPCQ     106.33       0.0000       25.315       800.00
 ----------------------------
Node 21: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     98338.
 TEXTILPQ    -369.76       0.0000       16.363       260.00
 ----------------------------
Node 11: Intermediate node
 A case goes into Node 22 if AGE_REF <= 67.500000
 AGE_REF mean = 51.612125
 ----------------------------
Node 22: Intermediate node
 A case goes into Node 44 if CUTENURE = "2"
 CUTENURE mode = "1"
 ----------------------------
Node 44: Intermediate node
 A case goes into Node 88 if NO_EARNR <= 1.5000000
 NO_EARNR mean = 2.0063564
 ----------------------------
Node 88: Terminal node
```

```
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant    0.68170E+06
 HIGH_EDU    -41669.       14.000       15.480       16.000
 ----------------------------
Node 89: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     9000.0
 UTILRNTC     360.23       0.0000       2.9490       248.00
 ----------------------------
Node 45: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant    0.57253E+06
 INCLASS2    -94838.       4.0000       5.9945       6.0000
 ----------------------------
Node 23: Intermediate node
 A case goes into Node 46 if FEDRFNDX <= 92.500000 or FEDR_NDX = "A"
 FEDRFNDX mean = 4824.1804
 ----------------------------
Node 46: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     98338.
 FSALARYX   -0.33329       0.0000       71486.      0.26605E+06
 ----------------------------
Node 47: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     22000.
 TOBACCPQ     93.209       0.0000       65.273       871.00
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if STOCKX <= 583000.00 or STOCKX_ = "C"
 STOCKX mean = 1647662.0
 ----------------------------
Node 6: Intermediate node
 A case goes into Node 12 if OTHLODPQ <= 55.000000
 OTHLODPQ mean = 369.76259
 ----------------------------
Node 12: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant    -51345.
 UNISTRQ      21383.       3.0000       3.4863       10.000
```

```
 ----------------------------
Node 13: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     25000.
 SMLAPPCQ     2716.2       0.0000       3.2484       53.000
 ----------------------------
Node 7: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant    -69823.
 INCLASS      18684.       5.0000       8.4720       9.0000
 ----------------------------
 Observed and fitted values are stored in quantlin.fit
 LaTeX code for tree is in quantlin.tex
 R code is stored in quantlin.r
```

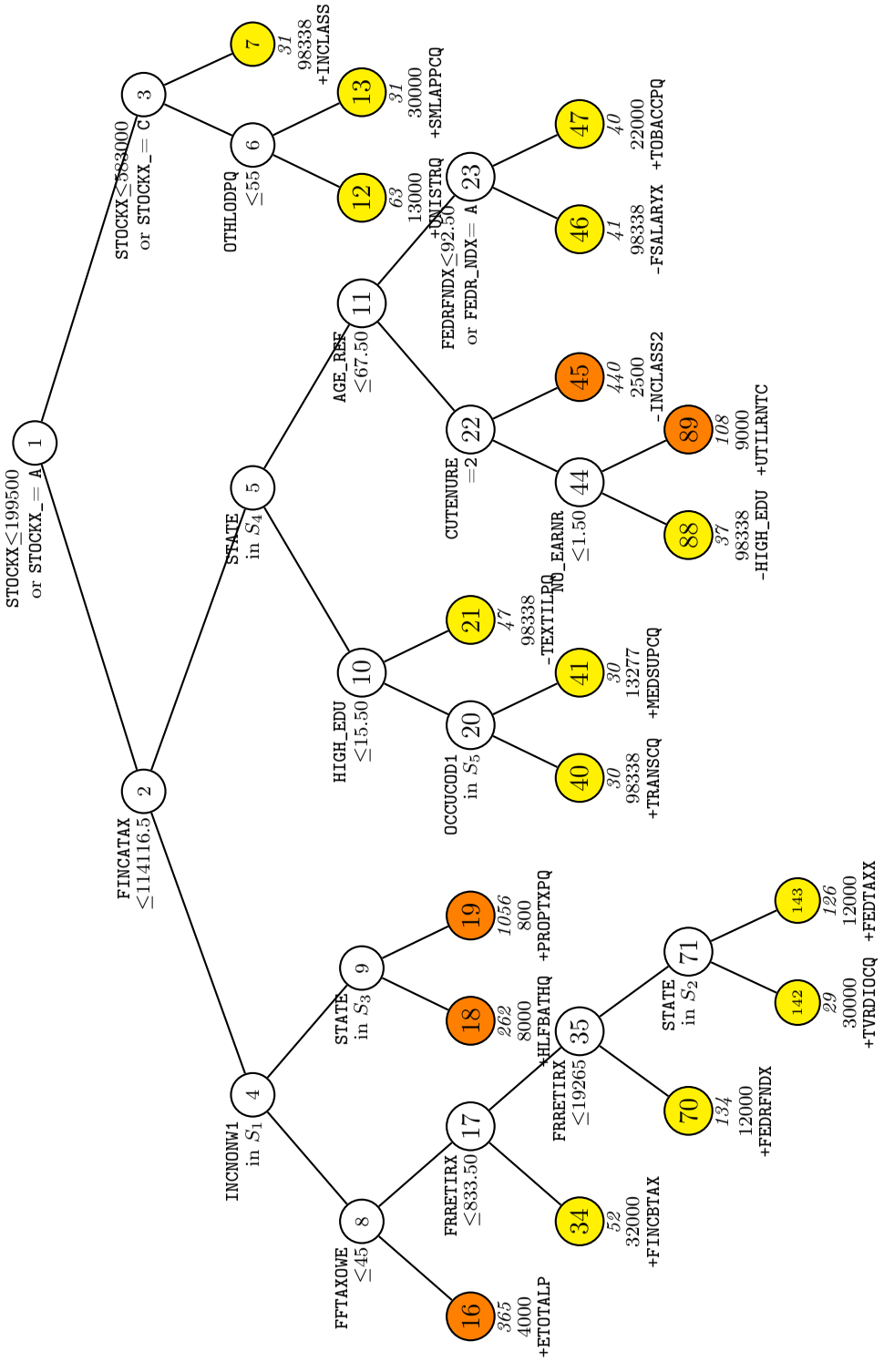

Figure 16: GUIDE v.36.0 0.50-SE piecewise simple linear 0.900-quantile regression tree for predicting INTRDVX. Tree constructed with 2922 observations (excluding observations with non-positive weight or missing values in D, T, R or Z variables). Maximum number of split levels is 12 and minimum node sample size is 29. 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 $S_2 = \{13, 23, 32, 41, 45, 48, 8\}$. 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 9500.0 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. To restrict the tree size, each node is required to have at least 50 observations.

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=proportional hazards,
 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): 2
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): 0.25
Input 2nd quantile probability ([0.00:1.00], <cr>=0.75): 0.75
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
423 N variables changed to S
Dependent 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 ...
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 C variable values ...
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: variable MISC2PQ is constant
Warning: variable MISC2CQ is constant
:
:
Warning: variable OTHLONBX is constant
Warning: 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       15        0        0      423
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      172        0       41        0
Number of cases used for training: 2922
Number of split variables: 464
Number of cases excluded due to 0 weight or missing D: 1771
Finished reading data file
Default number of cross-validations:           10
Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
Best tree may be chosen based on mean or median CV estimate
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.50):
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: All positive weights treated as one
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 14
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1): 2
Input minimum node size ([2:1461], <cr>=14): 50
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
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>=8):
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
       3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save regressor names in a file, 1 otherwise ([1:2], <cr>=1):
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): 2
Input file name: twoquant.r
Input 1 to overwrite it, 2 to choose another name ([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 < 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
     51  FJSSDEDX   s    0.000       0.3042E+05
     52  FJSS_EDX   m                                 0
      :
    513  INTRDVX    d    1.000       0.9834E+05
    522  IRAB       s    1.000        6.000                  2826
    523  IRAB_      m                                 2
      :
    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: 0.5000

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
     5  0.1390E+03   OCCUCOD1
     6  0.1278E+03   STATE
     7  0.1166E+03   AGE2
     8  0.1156E+03   INCOMEY1
     9  0.1154E+03   FJSSDEDX
    10  0.1142E+03   INC_HRS1
     :
   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 ++

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
          6T    1507     1507     1  3.600E+01  3.000E+03  STATE
          7T      30       30     1  1.160E+04  9.834E+04  -

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 CUTENURE

Regression tree:

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: INTRDVX sample quantiles = 36.000000, 3000.0000
  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 6: Terminal node
 Sample 0.250-quantile, 0.750-quantile, and median:
     3.6000E+01     3.0000E+03     3.0000E+02
 ----------------------------
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.

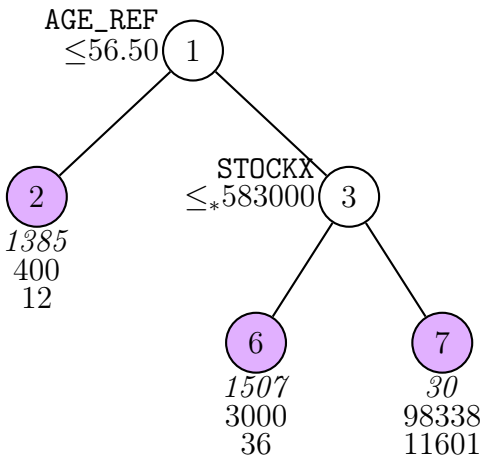

Figure 17: GUIDE v.36.2 0.50-SE piecewise constant 0.250 and 0.750-quantile regression tree for predicting `INTRDVX`. Tree constructed with 2922 observations (excluding observations with non-positive weight or missing values in `D`, `T`, `R` or `Z` variables). Maximum number of split levels is 12 and 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 '$\leq$ or missing'. Sample size (*in italics*) and sample 0.750 and 0.250-quantiles of `INTRDVX` printed below nodes. Second best split variable at root node is `CUTENURE`.

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=proportional hazards,
  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: ?
Warning: missing value code must be NA if R code is desired
Records in data file start on line 1
Warning: B variables changed to C
Dependent 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 ...
Assigning integer codes to C variable values ...
Re-checking data ...
Assigning codes to categorical and missing values ...
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.

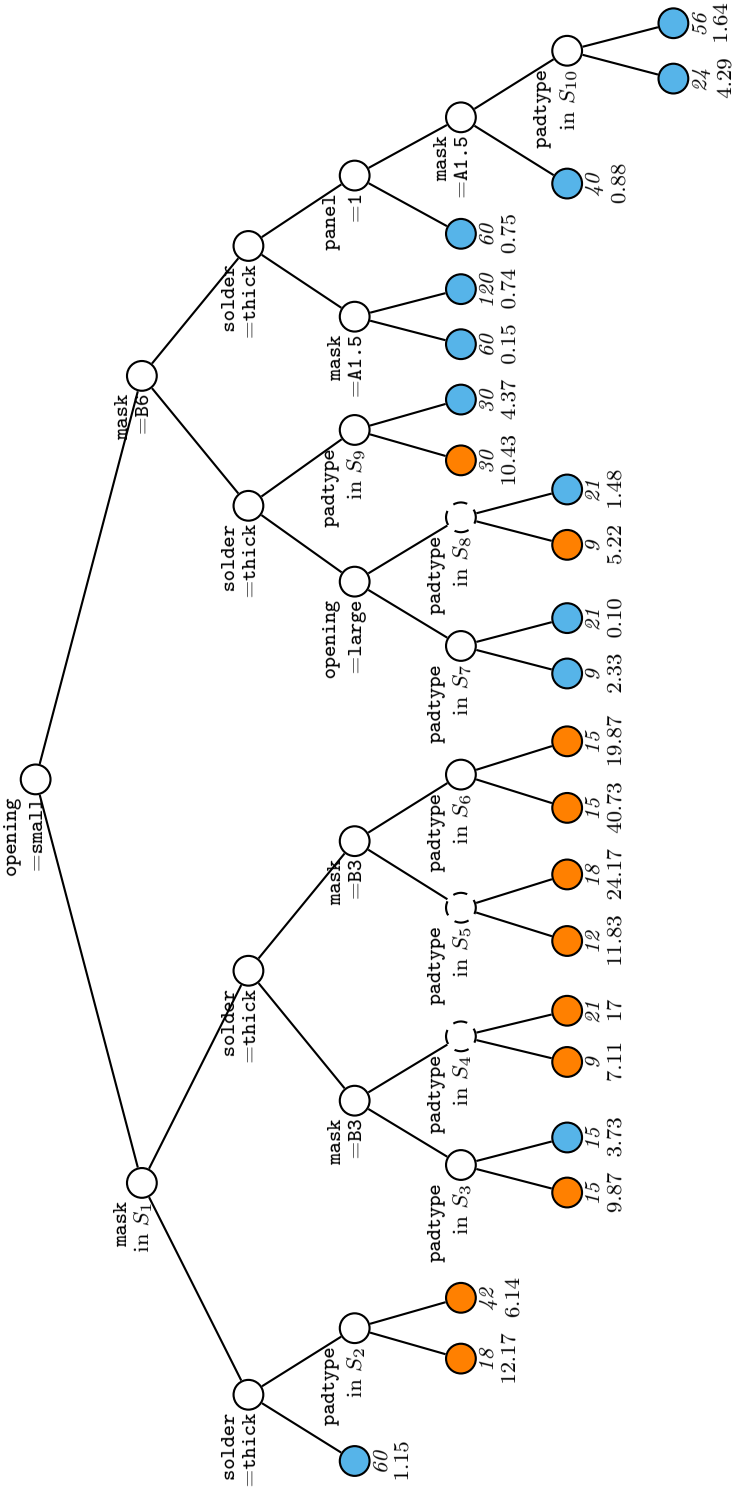

Figure 18: GUIDE v.36.2 0.50-SE piecewise constant Poisson regression tree for predicting `skips`. Tree constructed with 720 observations. Maximum number of split levels is 10 and minimum node sample size is 7. At each split, an observation goes to the left branch if and only if the condition is satisfied. 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 ...
Assigning integer codes to categorical variable values ...
Re-checking data ...
Assigning codes to categorical and missing values ...
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):
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: 0.5000

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
          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
```

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 Poisson regression with offset: lung cancer data

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

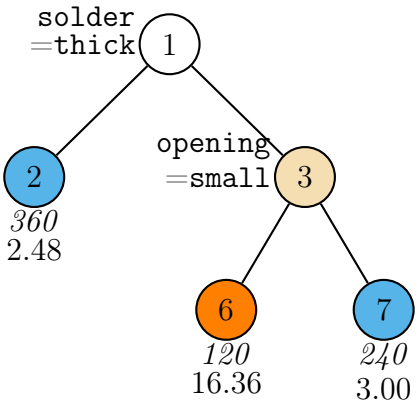

Figure 19: GUIDE v.36.2 0.50-SE multiple linear Poisson regression tree for predicting `skips`. Tree constructed with 720 observations. Maximum number of split levels is 10 and minimum node sample size is 21. At each split, an observation goes to the left branch if and only if the condition is satisfied. 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`.

```
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:

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

or, equivalently,

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

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 ...
Assigning integer codes to categorical variable values ...
Re-checking data ...
Assigning codes to categorical and missing values ...
Data checks complete
Number of cases with positive D values: 869
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 1
Creating dummy variables ...
Rereading data ...
     Total  #cases 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: 0.5000

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: 9
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       55   3.067E+00   3.630E-01   2.458E-01   3.007E+00   3.510E-01
   2       54   3.067E+00   3.630E-01   2.458E-01   3.007E+00   3.510E-01
   :
  36        4   2.243E+00   3.042E-01   2.489E-01   1.950E+00   3.281E-01
  37**      3   2.220E+00   3.271E-01   2.721E-01   1.910E+00   2.842E-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-002
 Node 1: agegp /= "45-54"
   Node 3: agegp = "55-64"
     Node 6: deaths sample rate = 0.13389777E-001
   Node 3: agegp /= "55-64"
     Node 7: deaths sample rate = 0.20932715E-001

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


 WARNING: p-values below 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.

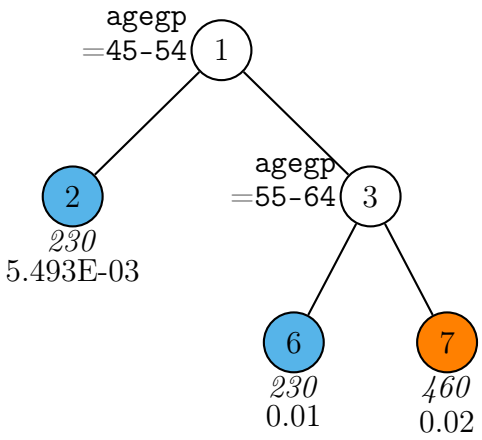

Figure 20: GUIDE v.36.2 0.50-SE multiple linear Poisson regression tree for predicting rate of `deaths`. Tree constructed with 920 observations. Maximum number of split levels is 10 and minimum node sample size is 9. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (*in italics*) and sample rate printed below nodes. Terminal nodes with means 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 x
26 das2d3pc x
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 x
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 ...
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 categorical variable values ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
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       3443       11        0        0       20
    #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: .649
Number of cases used for training: 5735
Number of split variables: 51
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
20 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
     :
     62  income     c                                4
     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       3443       11        0        0       20
   #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: 51
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: 0.5000

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: 5
Top-ranked variables and chi-squared values at root node
     1  0.2324E+03   cat1
```

```
     2  0.2274E+03   aps1
     :
    48  0.2328E-01   cardiohx
    49  0.1168E-01   chrpulhx

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       75   1.312E+00   1.808E-02   9.979E-03   1.309E+00   5.767E-03
   2       74   1.312E+00   1.808E-02   9.990E-03   1.309E+00   5.798E-03
   :
  37       15   1.290E+00   1.704E-02   1.012E-02   1.284E+00   7.943E-03
  38*      13   1.285E+00   1.675E-02   1.073E-02   1.277E+00   9.709E-03
  39       12   1.286E+00   1.672E-02   1.105E-02   1.279E+00   1.003E-02
  40       11   1.288E+00   1.669E-02   1.097E-02   1.279E+00   8.969E-03
  41--     10   1.287E+00   1.666E-02   1.029E-02   1.279E+00   8.659E-03
  42**      8   1.291E+00   1.662E-02   9.290E-03   1.282E+00   8.240E-03
  43++      6   1.296E+00   1.633E-02   1.003E-02   1.280E+00   1.342E-02
  44        5   1.310E+00   1.617E-02   1.020E-02   1.307E+00   7.596E-03
  45        4   1.337E+00   1.635E-02   1.025E-02   1.325E+00   1.179E-02
  46        3   1.356E+00   1.579E-02   8.738E-03   1.356E+00   9.784E-03
  47        2   1.400E+00   1.592E-02   1.309E-02   1.406E+00   1.843E-02
  48        1   1.435E+00   1.607E-02   6.168E-03   1.431E+00   9.875E-03

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

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

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

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

Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix    Node       Node    Split          Other
       label   cases      fit  rank   rel.risk   deviance  variable       variables
          1    5735     5735     1  1.000E+00  1.435E+00  cat1
          2     874      874     1  2.119E+00  1.466E+00  aps1
          4     583      583     1  1.844E+00  1.524E+00  dnr1
          8T    468      468     1  1.643E+00  1.453E+00  age
          9T    115      115     1  3.274E+00  1.532E+00  hema1
          5T    291      291     1  2.859E+00  1.247E+00  cat1
```

```
          3     4861     4861     1  8.831E-01  1.352E+00  aps1
          6     3400     3400     1  7.569E-01  1.225E+00  dnr1
         12     3086     3086     1  6.998E-01  1.156E+00  age
        24T     1085     1085     1  5.109E-01  1.102E+00  ca
        25T     2001     2001     1  7.984E-01  1.153E+00  wtkilo1
        13T      314      314     1  1.581E+00  1.517E+00  paco21
          7     1461     1461     1  1.279E+00  1.527E+00  dnr1
        14T     1290     1290     1  1.163E+00  1.467E+00  ca
        15T      171      171     1  2.916E+00  1.455E+00  swang1

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 aps1

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

Node 1: cat1 = "Coma", "Lung Cancer", "MOSF w/Malignancy"
  Node 2: aps1 <= 65.500000
    Node 4: dnr1 = "No"
      Node 8: Relative hazard (relative to root node) = 1.6428634
    Node 4: dnr1 /= "No"
      Node 9: Relative hazard (relative to root node) = 3.2736178
  Node 2: aps1 > 65.500000 or NA
    Node 5: Relative hazard (relative to root node) = 2.8586997
Node 1: cat1 /= "Coma", "Lung Cancer", "MOSF w/Malignancy"
  Node 3: aps1 <= 63.500000
    Node 6: dnr1 = "No"
      Node 12: age <= 56.303480
        Node 24: Relative hazard (relative to root node) = 0.51094711
      Node 12: age > 56.303480 or NA
        Node 25: Relative hazard (relative to root node) = 0.79841953
    Node 6: dnr1 /= "No"
      Node 13: Relative hazard (relative to root node) = 1.5812593
  Node 3: aps1 > 63.500000 or NA
    Node 7: dnr1 = "No"
      Node 14: Relative hazard (relative to root node) = 1.1630441
    Node 7: dnr1 /= "No"
      Node 15: Relative hazard (relative to root node) = 2.9157692

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

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 cat1 = "Coma", "Lung Cancer", "MOSF w/Malignancy"
 cat1 mode = "ARF"
 Coefficients of log-relative risk function:
 Regressor    Coefficient  t-stat      p-value
 Constant      0.000
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if aps1 <= 65.500000
 aps1 mean = 55.327231
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if dnr1 = "No"
 dnr1 mode = "No"
 ----------------------------
Node 8: Terminal node
 Coefficients of log-relative risk function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.4964
 ----------------------------
Node 9: Terminal node
 Coefficients of log-relative risk function:
 Regressor    Coefficient  t-stat      p-value
 Constant      1.186
 ----------------------------
Node 5: Terminal node
 Coefficients of log-relative risk function:
 Regressor    Coefficient  t-stat      p-value
 Constant      1.050
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if aps1 <= 63.500000
 aps1 mean = 54.549064
 ----------------------------
Node 6: Intermediate node
 A case goes into Node 12 if dnr1 = "No"
 dnr1 mode = "No"
```

```
 ----------------------------
Node 12: Intermediate node
 A case goes into Node 24 if age <= 56.303480
 age mean = 60.740517
 ----------------------------
Node 24: Terminal node
 Coefficients of log-relative risk function:
 Regressor    Coefficient  t-stat      p-value
 Constant    -0.6715
 ----------------------------
Node 25: Terminal node
 Coefficients of log-relative risk function:
 Regressor    Coefficient  t-stat      p-value
 Constant    -0.2251
 ----------------------------
Node 13: Terminal node
 Coefficients of log-relative risk function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.4582
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if dnr1 = "No"
 dnr1 mode = "No"
 ----------------------------
Node 14: Terminal node
 Coefficients of log-relative risk function:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.1510
 ----------------------------
Node 15: Terminal node
 Coefficients of log-relative risk function:
 Regressor    Coefficient  t-stat      p-value
 Constant      1.070
 ----------------------------
 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 survivaltime  logbasecumhaz  relativehaz   survivalprob  mediansurvtime
     y  25  2.400000E+02  -2.709325E-01  7.984177E-01  5.439260E-01    2.804946E+02
     y  25  4.500000E+01  -7.958768E-01  7.984177E-01  6.975068E-01    2.804946E+02
     y   5  3.170000E+02  -5.809518E-02  2.858721E+00  6.737782E-02    1.356823E+01
     y  25  3.700000E+01  -8.771591E-01  7.984177E-01  7.174012E-01    2.804946E+02
```

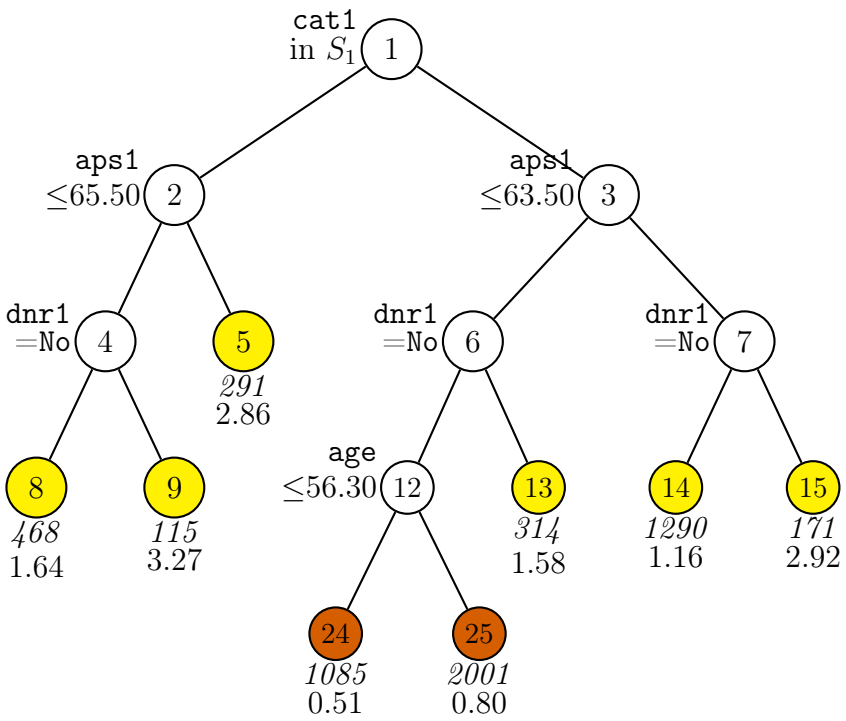

Figure 21: GUIDE v.36.2 0.50-SE piecewise constant proportional hazards regression tree for `survtime`. Tree constructed with 5735 observations. Maximum number of split levels is 15 and minimum node sample size is 57. At each split, an observation goes to the left branch if and only if the condition is satisfied. Set $S_1 = \{$`Coma`, `Lung Cancer`, `MOSF w/Malignancy`$\}$. Sample size (*in italics*) and relative hazards (relative to root node) printed below nodes. Terminal nodes with relative hazard above and below 1 are colored yellow and vermillion, respectively. Second best split variable at root node is `aps1`.

```
y  15  2.000000E+00  -3.962324E+00  2.915791E+00  9.460533E-01    1.317555E+01
```

The columns are:

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

`node`: terminal node label of observation.

`survivaltime`: observed survival time t.

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

`relativehaz`: hazard rate relative to that at the root node, i.e., $\exp(\beta)$. For example, the first subject, which is in node 25, has $\beta = -0.225123430735$ and so `relativehaz` $= \exp(-0.225123430735) = 0.7984177$. (The value of β may be obtained from `censored.out` or, for more accuracy, from `censored.r`).

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

$$\begin{aligned} \exp\{-\Lambda_0(t)\exp(\boldsymbol{\beta}'\mathbf{x})\} &= \exp\{-\exp(\texttt{logbasecumhaz}) \times \texttt{relativerisk}\} \\ &= \exp(-\exp(-0.2709325) \times 0.7984177) \\ &= 0.5439339. \end{aligned}$$

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

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$node))
leg.txt <- paste("Node",nodenum)
leg.col <- c("green","magenta","black","brown","orange","blue","cyan","red")
leg.lty <- 1:length(nodenum)
fit <- survfit(Surv(z0$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)
```

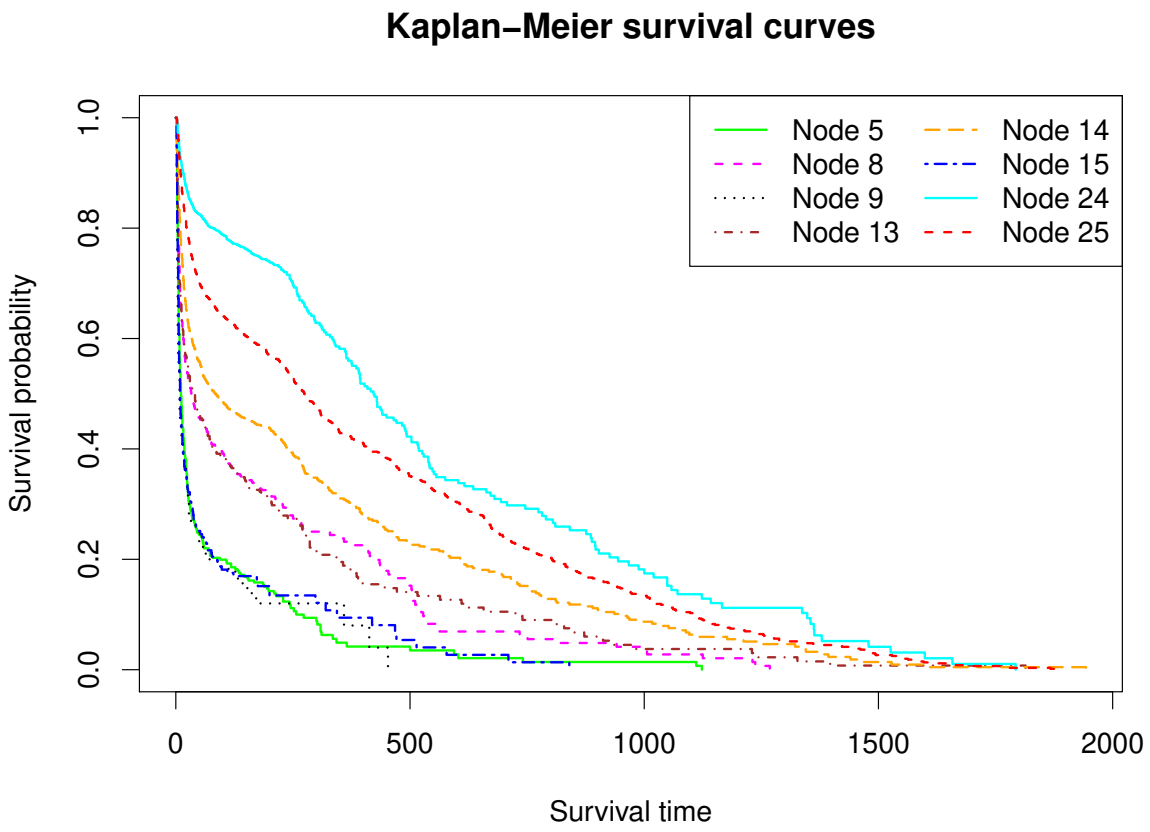

Figure 22: Kaplan-Meier survival curves for data in terminal nodes of Figure 21

10.2 Restricted mean event time

The mean survival time is not estimable if there is censoring. But given a pre-specified 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.

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 ...
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 categorical variable values ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
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       3443       11        0        0       20
    #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: 51
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
20 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
      :
     61  race       c                                 3
     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       3443       11        0        0       20
    #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: 51
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: 0.5000

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  0.1122E+03   cat1
     2  0.6234E+02   aps1
     :
    48  0.1196E+00   amihx
    49  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       74   1.162E+05   3.396E+03   2.369E+03   1.156E+05   4.469E+03
   2       73   1.162E+05   3.396E+03   2.370E+03   1.156E+05   4.476E+03
   :
  41+      15   1.147E+05   3.320E+03   2.362E+03   1.119E+05   4.304E+03
  42       14   1.147E+05   3.321E+03   2.313E+03   1.128E+05   3.580E+03
  43       12   1.140E+05   3.237E+03   2.398E+03   1.126E+05   3.461E+03
  44        9   1.144E+05   3.240E+03   2.526E+03   1.129E+05   3.686E+03
  45*       6   1.132E+05   3.122E+03   2.119E+03   1.128E+05   2.455E+03
  46        4   1.136E+05   2.844E+03   8.966E+02   1.132E+05   6.506E+02
  47**      2   1.142E+05   2.774E+03   1.026E+03   1.135E+05   1.352E+03
  48        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 15 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 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   cat1
          2T     905      905     1  1.649E+02  7.926E+04   scoma1
          3T    2827     2827     1  3.514E+02  2.011E+05   cat1

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 aps1

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

 Node 1: cat1 = "Cirrhosis", "Coma", "Lung Cancer", "MOSF w/Malignancy"
   Node 2: survtime-mean = 164.93170
 Node 1: cat1 /= "Cirrhosis", "Coma", "Lung Cancer", "MOSF w/Malignancy"
   Node 3: survtime-mean = 351.36539

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

 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 cat1 = "Cirrhosis", "Coma", "Lung Cancer",
 "MOSF w/Malignancy"
 cat1 mode = "ARF"
 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: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      164.9        17.62       0.7772E-15
 survtime mean = 164.932
 ----------------------------
Node 3: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      351.4        41.66        0.000
 survtime mean = 351.365
```

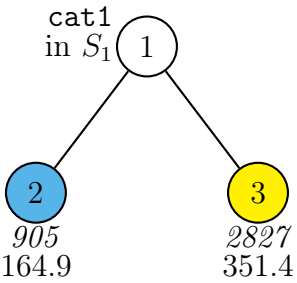

Figure 23: GUIDE v.36.2 0.50-SE piecewise constant regression tree for mean `survtime` restricted to less than 972.000. Tree constructed with 3732 observations (excluding observations with non-positive weight or missing values in `D`, `T`, `R` or `Z` variables). Maximum number of split levels is 13 and minimum node sample size is 37. At each split, an observation goes to the left branch if and only if the condition is satisfied. Set $S_1 = \{$`Cirrhosis`, `Coma`, `Lung Cancer`, `MOSF w/Malignancy`$\}$. Sample size (*in italics*) and restricted mean of `survtime` printed below nodes. Terminal nodes with means above and below value of 314.38 at root node are colored yellow and skyblue, respectively. Second best split variable at root node is `aps1`.

```
----------------------------
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 Subgroups: randomized trials

The goal of subgroup identification is to find patient subgroups who do or do not benefit from the treatment. Causal effects of a treatment are best studied in a randomized experiment where the treatment is assigned randomly to subjects. We illustrate some ways to do this here when there is a censored survival time. See Loh et al. (2019a, 2016, 2015, 2019c) and Loh and Zhou (2020) for more details. As in the previous section on the RHC data, the treatment variable is assumed to be categorical (i.e., it takes nominal values) and the response is an uncensored or censored event time (e.g., survival time). The key points are:

1. The treatment variable is designated in the description file as `R` or `r` (for "Rx").

2. The binary event indicator (such as "death") is designated as the dependent variable `D` or `d`. It is equal to 1 if an event occurred and 0 otherwise (right censored).

3. The event time (e.g., survival time) is designated as `T` or `t` (for "time").

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).

We demonstrate the capabilities with a data set from a randomized controlled breast cancer trial (Schmoor et al., 1996). The data are in the file `cancerdata.txt`; it can also be obtained from the `TH.data` R package (Hothorn, 2017). In the description file `cancerdsc.txt` below, the treatment variable is hormone therapy, `horTh`. The variable `time` is (censored) time to recurrence of cancer and `event` = 1 if the cancer recurred and = 0 if it did not. Ordinal predictor variables may be designated as "`n`" or "`s`" (with this option of no linear prognostic control, `n` variables will be automatically changed to `s` when the program is executed).

```
cancerdata.txt
NA
1
1 horTh r
2 age n
3 menostat c
4 tsize n
5 tgrade c
6 pnodes n
7 progrec n
8 estrec n
9 time t
10 event d
```

11.1 Censored response: proportional hazards

11.1.1 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 ...
Assigning integer codes to categorical variable values ...
```

```
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to categorical and missing values ...
Data checks complete
Smallest uncensored 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):
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: 0.5000

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: 5
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
   :
  29        4   1.461E+00   6.040E-02   4.355E-02   1.443E+00   4.585E-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.

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

Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix    Node       Node    Split
       label   cases      fit  rank   rel.risk   deviance  variable
          1      672      672     1  1.000E+00  1.431E+00  progrec
          2T     274      274     1  1.452E+00  1.601E+00  estrec
          3T     398      398     1  7.713E-01  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: Relative hazard (relative to root node) = 1.4519726
Node 1: progrec > 21.500000 or NA
  Node 3: Relative hazard (relative to root node) = 0.77133186

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


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 risk 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 risk 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 risk 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
```

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

$$\begin{aligned}\lambda(u,\mathbf{x}) &= \lambda_0(u)[\exp\{\hat{\beta}_1 + \hat{\gamma}_1 I(\texttt{horTh} = \texttt{yes})\} I(\texttt{progrec} \le 21.5) \\ &\quad + \exp\{\hat{\beta}_2 + \hat{\gamma}_2 I(\texttt{horTh} = \texttt{yes})\} I(\texttt{progrec} > 21.5)]\end{aligned}$$

with $\hat{\beta}_1 = 0.37292$, $\hat{\gamma}_1 = -0.11404$, $\hat{\beta}_2 = -0.25964$, and $\hat{\gamma}_2 = -0.64531$.

Figure 24 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 25 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")
```

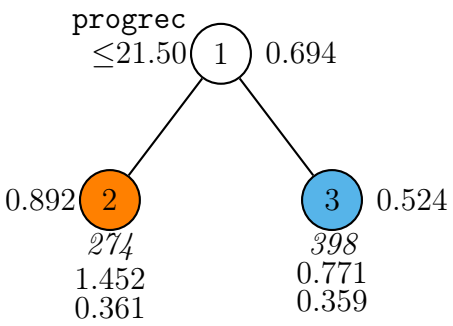

Figure 24: GUIDE v.36.2 0.50-SE proportional hazards regression tree using Gi option for `time` and event indicator `death` without linear prognostic effects. Tree constructed with 672 observations (excluding observations with non-positive weight or missing values in `D`, `T`, `R` or `Z` variables). Maximum number of split levels is 10, minimum node sample size is 6 and minimum treatment fraction is 0.180. 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*), relative baseline hazard (relative to that at root node), and proportion of `horTh` = `yes` printed below nodes. Terminal nodes with hazard ratios above and below value at root node are colored orange and skyblue, respectively. Second best split variable at root node is `estrec`.

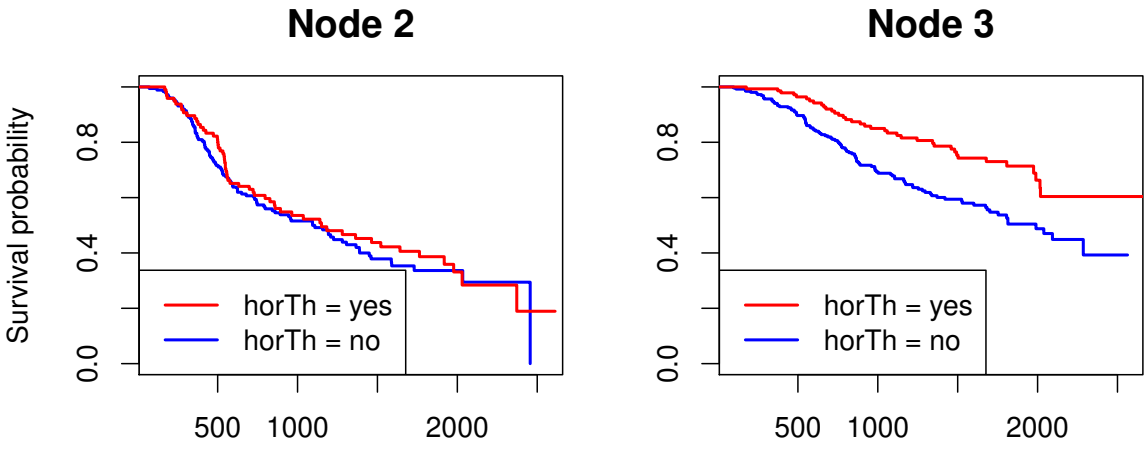

Figure 25: Estimated survival probability functions for breast cancer data

```
leg.col <- c("red","blue")
leg.lty <- 1:2
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, estimated survival time, log baseline cumulative hazard, relative risk (relative to the average for the data, ignoring covariates), survival probability, median survival time, and treatment effect (regression coefficient of treatment indicator) of each observation in the training sample (`cancerdata.txt`). The results for the first few observations are shown below. A plus (+) sign at the end of a value in the last column indicates that the observed survival time is censored.

train	node	survivaltime	logbasecumhaz	relativerisk	survivalprob	mediansurvtime	horTh.yes
y	3	1.814000E+03	-3.356226E-01	7.713319E-01	5.761313E-01	2.275550E+03	-6.453111E-01
y	3	2.018000E+03	-2.103084E-01	7.713319E-01	7.204845E-01	2.275550E+03	-6.453111E-01
y	3	7.120000E+02	-1.284520E+00	7.713319E-01	8.940654E-01	2.275550E+03	-6.453111E-01
y	3	1.807000E+03	-3.581910E-01	7.713319E-01	7.536968E-01	2.275550E+03	-6.453111E-01
y	3	7.720000E+02	-1.162320E+00	7.713319E-01	7.856518E-01	2.275550E+03	-6.453111E-01
y	2	4.480000E+02	-2.083218E+00	1.451973E+00	8.345918E-01	1.173368E+03	-1.140416E-01

11.1.2 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 ...
Assigning integer codes to categorical variable values ...
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to categorical and missing values ...
```

```
Data checks complete
Smallest uncensored 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):
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 there are no splits. The best linear predictor at the root node is the prognostic

variable pnodes.

```
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: 0.5000

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: 5
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
 * 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.

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

 Cases fit give the number of cases used to fit node
 Deviance is mean residual deviance for all cases in node
        Node    Total    Cases Matrix    Node       Node    Split
        label   cases      fit  rank   rel.risk   deviance  variable
           1T     672      672     3  1.000E+00  1.343E+00  estrec
 Best split at root node is estrec <= 4.5000

 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: Relative hazard (relative to root node) = 1.0000000

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


Node 1: Terminal node
 Coefficients of log-relative risk 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
 ----------------------------
```

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

11.2 Censored response: restricted mean

11.2.1 Without linear prognostic control

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 ...
Assigning integer codes to categorical variable values ...
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to categorical and missing values ...
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):
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: 0.5000

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
```

11.2.2 With linear prognostic control

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 ...
Assigning integer codes to categorical variable values ...
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to categorical and missing values ...
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):
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: 0.5000

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
 ----------------------------
 LaTeX code for tree is in rest-lin-gi.tex
```

11.3 Uncensored response

If response variable is uncensored (e.g., survival time with no censoring or an event indicator such as 1=abstinent, 0=smoking relapse), designate it with `D` in the description file and do not designate any variable with `T`.

12 Subgroups: 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 x
26 das2d3pc x
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 x
60 urin1 n
61 race c
62 income c
63 ptid x
64 survtime t
```

12.1 Censored response: proportional hazards

12.1.1 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 ...
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 categorical variable values ...
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
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       3443       11        0        0       20
    #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: .649
Number of cases used for training: 5735
Number of split variables: 50
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):
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
```

12.1.2 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
20 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
      :
     58  ortho       c                                 2
     60  urin1       s    0.000        9000.                  3028
     61  race        c                                 3
     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       3443       11        0        0       20
   #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: 50
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: 0.5000

No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
```

```
Maximum number of split levels: 15
Minimum node sample size: 57
Minimum fraction of cases per treatment at each node: 0.076
Number of iterations: 5
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
     5  0.5987E+01   aps1
     :
    32  0.1497E-01   sod1
    33  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       74   1.392E+00   1.709E-02   9.794E-03   1.378E+00   1.105E-02
   2       73   1.392E+00   1.710E-02   9.800E-03   1.378E+00   1.115E-02
   :
  34       15   1.365E+00   1.644E-02   8.102E-03   1.357E+00   9.238E-03
  35++     14   1.358E+00   1.609E-02   5.827E-03   1.354E+00   5.338E-03
  36        9   1.359E+00   1.546E-02   4.822E-03   1.358E+00   4.053E-03
  37--      8   1.359E+00   1.545E-02   4.839E-03   1.358E+00   4.072E-03
  38**      5   1.361E+00   1.548E-02   4.810E-03   1.359E+00   5.155E-03
  39        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 14 terminal nodes
0-SE tree based on median is marked with + and has 14 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.

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

Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix    Node       Node    Split
       label   cases      fit  rank   rel.risk   deviance  variable
          1    5735     5735     1  1.000E+00  1.367E+00  ph1
```

```
          2     1411     1411     1  1.007E+00  1.454E+00  cat2
          4T    1307     1307     1  9.401E-01  1.416E+00  paco21
          5T     104      104     1  2.227E+00  1.636E+00  -
          3     4324     4324     1  9.979E-01  1.334E+00  resp1
          6     3341     3341     1  1.001E+00  1.333E+00  paco21
         12T     687      687     1  1.351E+00  1.531E+00  income
         13T    2654     2654     1  9.314E-01  1.265E+00  paco21
          7T     983      983     1  9.886E-01  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: Relative hazard (relative to root node) = 0.94005821
  Node 2: cat2 /= "MOSF w/Sepsis", "NA"
    Node 5: Relative hazard (relative to root node) = 2.2265831
Node 1: ph1 > 7.3344730 or NA
  Node 3: resp1 <= 38.500000 or NA
    Node 6: paco21 <= 29.498050
      Node 12: Relative hazard (relative to root node) = 1.3506462
    Node 6: paco21 > 29.498050 or NA
      Node 13: Relative hazard (relative to root node) = 0.93141525
  Node 3: resp1 > 38.500000
    Node 7: Relative hazard (relative to root node) = 0.98856914

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

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 risk 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 risk 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 risk 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 risk 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 risk 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 risk 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
```

Figure 26 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-001,0.406689682597)
   } 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-001)
     } else {
       nodeid <- 13
       predict <- c(-0.710500703708E-001,0.593672033426E-002)
     }
   } else {
     nodeid <- 7
     predict <- c(-0.114966933127E-001,0.355516696179)
   }
 }
 return(c(nodeid,predict))
}
## end of function
##
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")
```

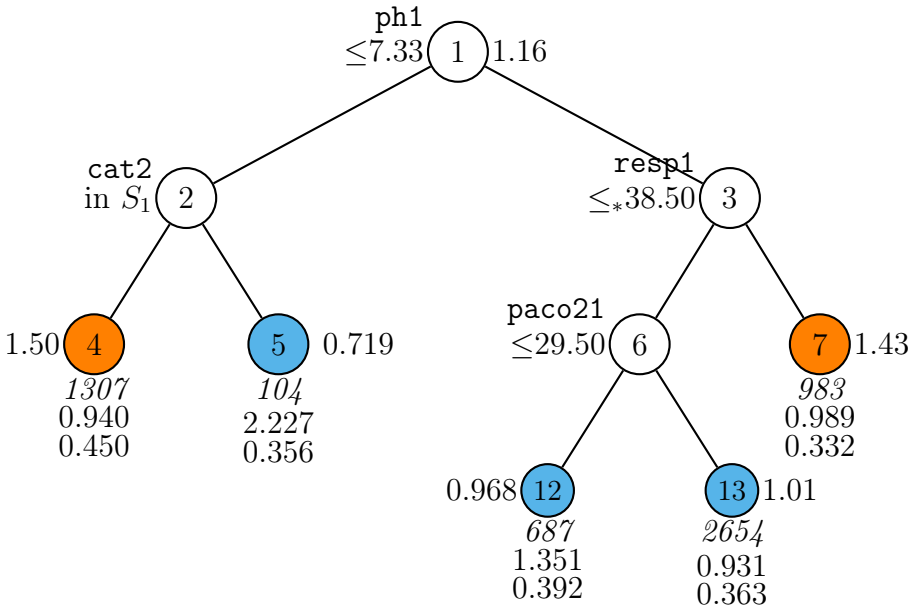

Figure 26: GUIDE v.36.2 0.50-SE proportional hazards regression tree using Gi option for `survtime` and event indicator `death` without linear prognostic effects. Tree constructed with 5735 observations. Maximum number of split levels is 15, minimum node sample size is 57 and minimum treatment fraction 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 '$\leq$ or missing'. Set S_1 = {`MOSF w/Sepsis`, `NA`}. Treatment `swang1` hazard ratio of level `RHC` to `NoRHC` beside nodes. Sample size (*in italics*), relative baseline hazard (relative to that at root node), and proportion of `swang1` = `RHC` printed below nodes. Terminal nodes with hazard ratios above and below value at root node are colored orange and skyblue, respectively. Second best split variable at root node is `resp1`.

```
## 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])
}
```

12.1.3 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 ...
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 categorical variable values ...
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
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       3443       11        0        0       20
    #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: .649
Number of cases used for training: 5735
Number of split variables: 50
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):
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
```

12.1.4 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
20 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
      :
     61  race        c                                 3
     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       3443       11        0        0       20
   #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: 50
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: 0.5000

No nodewise interaction tests
```

```
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 15
Minimum node sample size: 57
Minimum fraction of cases per treatment at each node: 0.076
Number of iterations: 5
Top-ranked variables and chi-squared values at root node
     1  0.2041E+03   ca
     2  0.1863E+03   malighx
     3  0.1857E+03   age
     4  0.1566E+03   cat1
     5  0.1340E+03   aps1
     :
    32  0.8147E+00   sex
    33  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       78   1.342E+00   1.837E-02   1.114E-02   1.358E+00   1.985E-02
   2       77   1.342E+00   1.837E-02   1.105E-02   1.358E+00   1.970E-02
   :
  44       20   1.304E+00   1.673E-02   1.095E-02   1.317E+00   1.350E-02
  45*      19   1.303E+00   1.671E-02   1.085E-02   1.316E+00   1.299E-02
  46       17   1.305E+00   1.674E-02   1.126E-02   1.321E+00   1.432E-02
  47       13   1.305E+00   1.661E-02   1.131E-02   1.318E+00   1.405E-02
  48++     12   1.305E+00   1.661E-02   1.131E-02   1.318E+00   1.405E-02
  49**     11   1.310E+00   1.650E-02   1.248E-02   1.326E+00   1.814E-02
  50        8   1.327E+00   1.648E-02   1.122E-02   1.343E+00   1.638E-02
  51        7   1.329E+00   1.647E-02   1.074E-02   1.343E+00   1.569E-02
  52        6   1.329E+00   1.639E-02   1.055E-02   1.343E+00   1.487E-02
  53        5   1.336E+00   1.631E-02   1.234E-02   1.345E+00   1.642E-02
  54        4   1.344E+00   1.642E-02   1.487E-02   1.344E+00   2.134E-02
  55        3   1.351E+00   1.647E-02   1.405E-02   1.344E+00   1.833E-02
  56        2   1.366E+00   1.610E-02   1.266E-02   1.368E+00   1.168E-02
  57        1   1.408E+00   1.578E-02   6.412E-03   1.398E+00   1.033E-02

0-SE tree based on mean is marked with * and has 19 terminal nodes
0-SE tree based on median is marked with + and has 19 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.
```

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

Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node
```

Node label	Total cases	Cases fit	Matrix rank	Node rel.risk	Node deviance	Split variable
1	5735	5735	1	1.000E+00	1.408E+00	ca
2	4379	4379	1	8.635E-01	1.408E+00	age
4	1577	1577	1	6.319E-01	1.392E+00	cat1
8T	227	227	1	1.267E+00	1.760E+00	crea1
9	1350	1350	1	5.195E-01	1.273E+00	aps1
18T	613	613	1	3.746E-01	9.228E-01	immunhx
19	737	737	1	6.956E-01	1.487E+00	bili1
38T	617	617	1	6.083E-01	1.374E+00	age
39T	120	120	1	1.531E+00	1.636E+00	pot1
5	2802	2802	1	9.937E-01	1.382E+00	urin1
10T	1507	1507	1	9.315E-01	1.372E+00	age
11	1295	1295	1	1.096E+00	1.383E+00	urin1
22T	318	318	1	1.941E+00	1.483E+00	scoma1
23	977	977	1	9.256E-01	1.273E+00	cat1
46T	102	102	1	2.268E+00	1.544E+00	-
47T	875	875	1	8.091E-01	1.181E+00	age
3	1356	1356	1	1.491E+00	1.268E+00	cat1
6T	521	521	1	1.929E+00	1.160E+00	aps1
7	835	835	1	1.288E+00	1.258E+00	cat2
14T	524	524	1	1.009E+00	1.179E+00	ninsclas
15T	311	311	1	1.966E+00	1.221E+00	pot1

```
Number of terminal nodes of final tree: 11
Total number of nodes of final tree: 21
Second best split variable (based on curvature test) at root node is malighx

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

Node 1: ca = "No"
  Node 2: age <= 56.241975
    Node 4: cat1 = "Cirrhosis", "Coma"
      Node 8: Relative hazard (relative to root node) = 1.2670055
    Node 4: cat1 /= "Cirrhosis", "Coma"
      Node 9: aps1 <= 50.500000
        Node 18: Relative hazard (relative to root node) = 0.37458886
      Node 9: aps1 > 50.500000 or NA
        Node 19: bili1 <= 7.2998047
          Node 38: Relative hazard (relative to root node) = 0.60830182
```

```
        Node 19: bili1 > 7.2998047 or NA
          Node 39: Relative hazard (relative to root node) = 1.5307482
  Node 2: age > 56.241975 or NA
    Node 5: urin1 = NA
      Node 10: Relative hazard (relative to root node) = 0.93147869
    Node 5: urin1 /= NA
      Node 11: urin1 <= 1002.0000
        Node 22: Relative hazard (relative to root node) = 1.9413899
      Node 11: urin1 > 1002.0000 or NA
        Node 23: cat1 = "Coma"
          Node 46: Relative hazard (relative to root node) = 2.2683538
        Node 23: cat1 /= "Coma"
          Node 47: Relative hazard (relative to root node) = 0.80908406
Node 1: ca /= "No"
  Node 3: cat1 = "Cirrhosis", "Coma", "Lung Cancer", "MOSF w/Malignancy"
    Node 6: Relative hazard (relative to root node) = 1.9286907
  Node 3: cat1 /= "Cirrhosis", "Coma", "Lung Cancer", "MOSF w/Malignancy"
    Node 7: cat2 = "Cirrhosis", "NA"
      Node 14: Relative hazard (relative to root node) = 1.0087240
    Node 7: cat2 /= "Cirrhosis", "NA"
      Node 15: Relative hazard (relative to root node) = 1.9659896

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

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 ca = "No"
ca mode = "No"
Coefficients of log-relative risk function:
Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
Constant       0.000
swang1.RHC    0.1514        4.525       0.6165E-05    0.000       0.3808        1.000
----------------------------
```

```
Node 2: Intermediate node
 A case goes into Node 4 if age <= 56.241975
 age mean = 60.883990
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if cat1 = "Cirrhosis", "Coma"
 cat1 mode = "ARF"
 ----------------------------
Node 8: Terminal node
 Coefficients of log-relative risk function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.2367
 swang1.RHC    0.5577        2.695       0.7561E-02    0.000       0.1674        1.000
 ----------------------------
Node 9: Intermediate node
 A case goes into Node 18 if aps1 <= 50.500000
 aps1 mean = 53.948148
 ----------------------------
Node 18: Terminal node
 Coefficients of log-relative risk function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -0.9819
 swang1.RHC    0.2715        1.848       0.6515E-01    0.000       0.3409        1.000
 ----------------------------
Node 19: Intermediate node
 A case goes into Node 38 if bili1 <= 7.2998047
 bili1 mean = 4.4573584
 ----------------------------
Node 38: Terminal node
 Coefficients of log-relative risk function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -0.4971
 swang1.RHC    0.2940        2.468       0.1385E-01    0.000       0.4814        1.000
 ----------------------------
Node 39: Terminal node
 Coefficients of log-relative risk function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.4258
 swang1.RHC    0.3461        1.622       0.1075        0.000       0.6000        1.000
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if urin1 = NA
 urin1 mean = 2003.6963
 ----------------------------
Node 10: Terminal node
 Coefficients of log-relative risk function:
```

```
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -0.7098E-01
 swang1.RHC    0.1575        2.526       0.1163E-01    0.000       0.3736        1.000
 ----------------------------
Node 11: Intermediate node
 A case goes into Node 22 if urin1 <= 1002.0000
 urin1 mean = 2003.6963
 ----------------------------
Node 22: Terminal node
 Coefficients of log-relative risk function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.6634
 swang1.RHC    0.9913E-01   0.7906       0.4297        0.000       0.4560        1.000
 ----------------------------
Node 23: Intermediate node
 A case goes into Node 46 if cat1 = "Coma"
 cat1 mode = "ARF"
 ----------------------------
Node 46: Terminal node
 Coefficients of log-relative risk function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.8191
 swang1.RHC    0.1260       0.4917       0.6240        0.000       0.2157        1.000
 ----------------------------
Node 47: Terminal node
 Coefficients of log-relative risk function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -0.2119
 swang1.RHC    0.2923        3.226       0.1301E-02    0.000       0.4354        1.000
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if cat1 = "Cirrhosis", "Coma", "Lung Cancer",
 "MOSF w/Malignancy"
 cat1 mode = "ARF"
 ----------------------------
Node 6: Terminal node
 Coefficients of log-relative risk function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.6568
 swang1.RHC    0.2651        2.743       0.6300E-02    0.000       0.3474        1.000
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if cat2 = "Cirrhosis", "NA"
 cat2 mode = "NA"
 ----------------------------
Node 14: Terminal node
```

```
 Coefficients of log-relative risk function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.8686E-02
 swang1.RHC    0.1045       0.9408       0.3472        0.000       0.3454        1.000
 ----------------------------
Node 15: Terminal node
 Coefficients of log-relative risk function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.6760
 swang1.RHC   -0.1350       -1.039       0.2997        0.000       0.3055        1.000
 ----------------------------
 Observed and fitted values are stored in surv-gs.fit
 LaTeX code for tree is in surv-gs.tex
```

Figure 27 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 survivaltime logbasecumhaz  relativehaz survivalprob mediansurvtime   swang1.RHC
    y   14 2.400000E+02 -2.470827E-01 1.008724E+00 4.548033E-01   1.797160E+02 1.045482E-01
    y   47 4.500000E+01 -7.650555E-01 8.090841E-01 6.039415E-01   2.710733E+02 2.922802E-01
    y    6 3.170000E+02 -4.143456E-02 1.928691E+00 8.961693E-02   2.470234E+01 2.651369E-01
```

The column definitions are

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

node: terminal node label of observation.

survivaltime: observed survival time t.

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

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 14 with $\beta_0 = 0.0086862$ and $\beta_1 = 0.10455$ (see towards the end of surv-gs.out). Thus its relativehaz $= \exp(\boldsymbol{\beta}'\mathbf{x}) = \exp(0.0086862 + 0.10455 \times I(\text{swang1} = \text{RHC})) = 1.008724$.

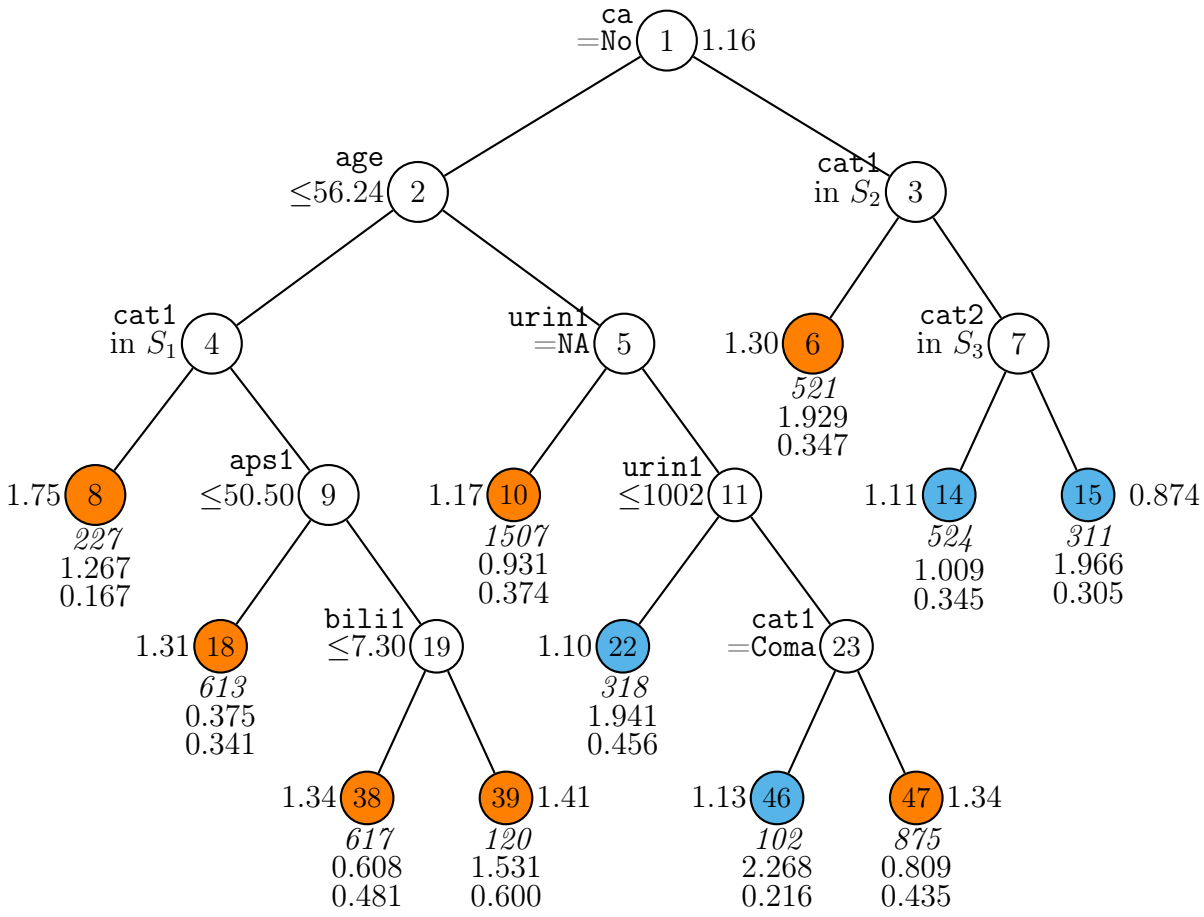

Figure 27: GUIDE v.36.2 0.50-SE proportional hazards regression tree using Gs option for `survtime` and event indicator `death` without linear prognostic effects. Tree constructed with 5735 observations. Maximum number of split levels is 15, minimum node sample size is 57 and minimum treatment fraction is 0.076. At each split, an observation goes to the left branch if and only if the condition is satisfied. Set S_1 = {`Cirrhosis`, `Coma`}. Set S_2 = {`Cirrhosis`, `Coma`, `Lung Cancer`, `MOSF w/Malignancy`}. Set S_3 = {`Cirrhosis`, `NA`}. Treatment `swang1` hazard ratio of level `RHC` to `NoRHC` beside nodes. Sample size (*in italics*), relative baseline hazard (relative to that at root node), and proportion of `swang1` = `RHC` printed below nodes. Terminal nodes with hazard ratios above and below value at root node are colored orange and skyblue, respectively. Second best split variable at root node is `malighx`.

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

$$\begin{aligned}\exp\{-\Lambda_0(t)\exp(\boldsymbol{\beta}'\mathbf{x})\} &= \exp\{-\exp(\texttt{logbasecumhaz})\times\texttt{relativehaz}\}\\ &= \exp(-\exp(-0.2470827)\times 1.008724)\\ &= 0.4548033.\end{aligned}$$

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

swang1.RHC: estimated treatment effect β_1 for level RHC of swang1. The relative hazard of RHC vs NoRHC is exp(swang1.RHC). For the first observation this is $\exp(0.1045482) = 1.110209$ (value is printed beside node 14 in Figure 27).

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. We show the Gs option here because the Gi option yields no splits after pruning.

12.2.1 Gs input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: rest-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: rest-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): 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 ...
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 categorical variable values ...
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
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): 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
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):
default value is just a suggestion; input your own value
```

```
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       3443       11        0        0       20
    #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: 50
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 1 for LaTeX tree code, 2 to skip it ([1:2], <cr>=1):

Input file name to store LaTeX code (use .tex as suffix): rest-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: rest-gs.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=1): 2
Input file name: rest-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 < rest-gs.in
```

12.2.2 Contents of `rest-gs.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
20 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
      :
     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       3443       11        0        0       20
    #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: 50
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: 0.5000
```

```
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: 37
Minimum fraction of cases per treatment at each node: 0.080
Rank of top variable to split root node is 2
Top-ranked variables and chi-squared values at root node
     1  0.1741E+03   cat1
     2  0.1008E+03   scoma1
     3  0.8505E+02   aps1
     4  0.7316E+02   chfhx
     5  0.4174E+02   urin1
     :
    34  0.7371E+00   age
    35  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       75   1.182E+05   3.529E+03   2.906E+03   1.186E+05   4.960E+03
   2       74   1.182E+05   3.529E+03   2.909E+03   1.186E+05   4.969E+03
   :
  47        3   1.114E+05   2.980E+03   1.815E+03   1.104E+05   2.460E+03
  48**      2   1.104E+05   2.940E+03   1.750E+03   1.095E+05   2.738E+03
  49        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 cat1
          3T     639      639     2  1.333E+02  4.975E+04  0.0016 cat1
```

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

 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:
```

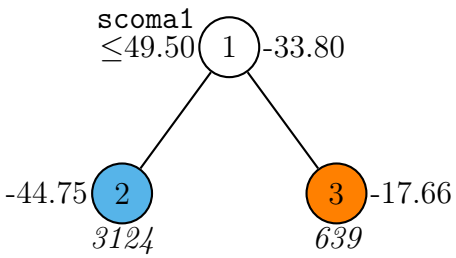

Figure 28: GUIDE v.36.2 0.50-SE regression tree using Gs option for mean `survtime` restricted to less than 622.00 without linear prognostic effects. Root node split with 2nd highest ranked variable. Tree constructed with 3763 observations (excluding observations with non-positive weight or missing values in `D`, `T`, `R` or `Z` variables). Maximum number of split levels is 13, minimum node sample size is 37 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 effects above and below value at root node are colored orange and skyblue, respectively.

```
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
----------------------------
LaTeX code for tree is in rest-gs.tex
R code is stored in rest-gs.fit
```

Figure 28 shows the Gs restricted mean event time tree.

13 Multi-response: public health 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. But 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 use the first option in this section. The second option is used in Section 14.

The data set `phs.dat` is from a public health survey of about 120,000 respondents. There are three `D` variables, namely, total restricted activity days in the past 2 week (`raday`), number of doctor visits in the past 12 months (`visit`), and number of short-stay hospital days in the past 12 months (`hda12`). Only 60000 respondents have values for all three response variables. The description files `phs.dsc` given below lists 6 numeric and 9 categorical variables.

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

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: phs.dsc
Reading data description file ...
Training sample file: phs.dat
Missing value code: NA
Records in data file start on line 1
6 N variables changed to S
Number of D variables; 3
D variables are:
raday
visit
hda12
Multivariate or univariate split variable selection:
Choose multivariate if there is an order among the D variables;
choose univariate otherwise or if item response
Input 1 for multivariate, 2 for univariate ([1:2], <cr>=2):
Input 1 to normalize D variables, 2 for no normalization ([1:2], <cr>=1):
Input 1 for equal, 2 for unequal weighting of D variables ([1:2], <cr>=1):
Reading data file ...
Number of records in data file: 119579
Length of longest entry in data file: 17
Checking for missing values ...
Missing values found in D variables
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 categorical variable values ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
Finished processing 10000 of 119579 observations
Finished processing 15000 of 119579 observations
:
Finished processing 110000 of 119579 observations
Finished processing 115000 of 119579 observations
Data checks complete
```

```
Normalizing data
Creating missing value indicators ...
Some D variables have missing values
Rereading data ...
PCA can be used for variable selection
Do not use PCA if differential item functioning (DIF) scores are wanted
Input 1 to use PCA, 2 otherwise ([1:2], <cr>=2):

#cases w/ miss. D = number of cases with all D values missing
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
    119579          0      30722        2        0        0        6
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        9        0
Number of cases used for training: 60000
Number of split variables: 15
Number of cases excluded due to 0 weight or missing D: 59579
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):
Input rank of top variable to split root node ([1:15], <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: phs.dsc
Training sample file: phs.dat
Missing value code: NA
Records in data file start on line 1
6 N variables changed to S
Number of D variables: 3
Univariate split variable selection method
Mean-squared errors (MSE) are calculated from normalized D variables
D variables equally weighted
Piecewise constant model
Number of records in data file: 119579
```

```
Length of longest entry in data file: 17
Missing values found in D variables
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Model fitted to subset of observations with complete D values
Neither LDA nor PCA used

Summary information for training sample of size 60000 (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  phone      c                                 4
      2  sex        c                                 2
      3  age        s    0.000        99.00
      4  race       c                                 3
      5  marstat    c                                 7       352
      6  educ       s    0.000        18.00                   5575
      7  income     s    0.000        26.00                  10499
      8  poverty    c                                 2      5420
      9  famsize    s    1.000        26.00
     10  condlist   c                                 6       288
     11  health     s    1.000        5.000                   305
     12  latotal    s    1.000        4.000
     13  wkclass    c                                 8     31764
     14  indus      c                                14     31912
     15  occup      c                                14     31917
     16  raday      d    0.000        14.00
     17  visit      d    0.000        637.0
     19  hda12      d    0.000        268.0

#cases w/ miss. D = number of cases with all D values missing
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
    119579          0      30722        2        0        0        6
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        9        0
Number of cases used for training: 60000
Number of split variables: 15
Number of cases excluded due to 0 weight or missing D: 59579
```

```
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: 0.5000

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 30
Minimum node sample size: 3000
Top-ranked variables and chi-squared values at root node
     1  0.1137E+05   latotal
     2  0.9500E+04   health
     3  0.1590E+04   marstat
     4  0.1527E+04   age
     5  0.8407E+03   indus
     6  0.7986E+03   occup
     7  0.7975E+03   wkclass
     8  0.7366E+03   sex
     9  0.5143E+03   income
    10  0.4982E+03   educ
    11  0.3819E+03   famsize
    12  0.2525E+03   poverty
    13  0.8300E+02   race
    14  0.2538E+02   phone
    15  0.4599E+01   condlist

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       14   6.709E+16   5.956E+14   4.803E+15   7.190E+16   1.152E+16
   2       12   6.709E+16   5.956E+14   4.803E+15   7.190E+16   1.152E+16
   3       11   2.919E+16   4.156E+14   1.064E+16   1.086E+16   1.921E+16
   4        9   1.208E+16   2.738E+14   3.718E+15   1.086E+16   1.008E+16
   5        8   1.208E+16   2.738E+14   3.718E+15   1.086E+16   1.008E+16
   6        7   1.208E+16   2.738E+14   3.718E+15   1.086E+16   1.008E+16
   7        6   7.437E+05   1.240E+04   1.092E+04   7.434E+05   1.998E+04
   8        5   7.437E+05   1.240E+04   1.092E+04   7.434E+05   1.998E+04
   9**      3   1.593E+00   7.349E-02   7.342E-02   1.592E+00   7.691E-02
  10        2   1.659E+00   7.352E-02   7.950E-02   1.677E+00   8.602E-02
  11        1   1.899E+00   7.710E-02   7.345E-02   1.917E+00   5.913E-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.

 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    60000    60000  1.310E+00   latotal
          2T    5936     5936  7.660E+00   -
          3    54064    54064  5.212E-01   health
          6T   50875    50875  4.330E-01   educ
          7T    3189     3189  1.787E+00   -

 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:

 Node 1: latotal <= 2.5000000
   Node 2: Mean cost = 3.8429747
 Node 1: latotal > 2.5000000 or NA
   Node 3: health <= 3.5000000 or NA
     Node 6: Mean cost = 0.21726310
   Node 3: health > 3.5000000
     Node 7: Mean cost = 0.89637215

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

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

Node 1: Intermediate node
 A case goes into Node 2 if latotal <= 2.5000000
 latotal mean = 3.7126500
 Means of raday, visit, and hda12
   6.1067E-01   4.1094E+00   6.0488E-01
 ----------------------------
Node 2: Terminal node
 Means of raday, visit, and hda12
   2.8630E+00   1.2474E+01   3.0076E+00
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if health <= 3.5000000 or NA
 health mean = 1.9395511
 ----------------------------
```

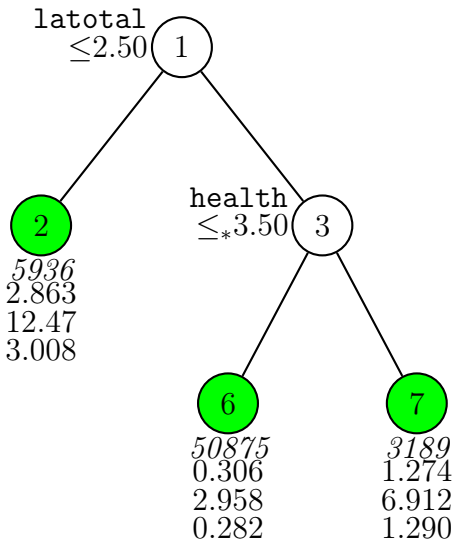

Figure 29: GUIDE v.36.0 0.50-SE regression tree for predicting response variables raday, visit, and hda12, without using PCA at each node. Tree constructed with 60000 observations (excluding observations with non-positive weight or missing values in D, T, R or Z variables). Maximum number of split levels is 30 and minimum node sample size is 3000. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. Sample size (*in italics*) and predicted values of raday, visit, and hda12 printed below nodes. Second best split variable at root node is health.

```
Node 6: Terminal node
 Means of raday, visit, and hda12
   3.0630E-01   2.9577E+00   2.8157E-01
 ----------------------------
Node 7: Terminal node
 Means of raday, visit, and hda12
   1.2738E+00   6.9116E+00   1.2904E+00
 ----------------------------
 Case and node IDs are in file: mult.nid
 Node fitted values are in file: mult.fit
 LaTeX code for tree is in mult.tex
```

The tree is shown in Figure 29. The file mult.fit saves the mean values of the dependent variables in each terminal node:

```
  node    raday         visit         hda12
     2  0.28630E+01  0.12474E+02  0.30076E+01
```

```
6  0.30630E+00  0.29577E+01  0.28157E+00
7  0.12738E+01  0.69116E+01  0.12904E+01
```

The file `mult.nid` gives the terminal node number for each observation, including those that are not used to construct the tree (indicated by the letter "`n`" in the `train` column of the file).

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; 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
 Choose non-default option here to allow change from default 0.5 SE to 0.0 SE below.
 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
 1 N variables 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 ...
Missing values found in D variables
Assigning integer codes to categorical variable values ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
#cases w/ miss. D = number of cases with all D values missing
     Total  #cases w/   #missing
```

```
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       888          0          0       40        0        0        1
    #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>=0.50): 0
This is where default 0.5 SE is changed to 0.0 SE.
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
1 N variables changed to S
Number of D variables: 13
Number of D variables: 13
D variables are:
wage1
wage2
wage3
wage4
wage5
wage6
wage7
wage8
wage9
wage10
wage11
wage12
wage13
T variables are:
exper1
exper2
exper3
exper4
exper5
exper6
exper7
exper8
exper9
exper10
exper11
```

```
exper12
exper13
Number of records in data file: 888
Length of longest entry in data file: 16
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  -0.1798+309    50.40                   38
     31  wage3      d  -0.1798+309    34.50                   77
     32  wage4      d  -0.1798+309    33.15                  124
     33  wage5      d  -0.1798+309    49.30                  159
     34  wage6      d  -0.1798+309    74.00                  233
     35  wage7      d  -0.1798+309    47.28                  325
     36  wage8      d  -0.1798+309    37.71                  428
     37  wage9      d  -0.1798+309    46.11                  551
     38  wage10     d  -0.1798+309    56.54                  678
     39  wage11     d  -0.1798+309    22.20                  791
     40  wage12     d  -0.1798+309    46.20                  856
     41  wage13     d  -0.1798+309    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 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      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 30 shows the tree and Figure 31 plots lowess-smoothed curves of mean

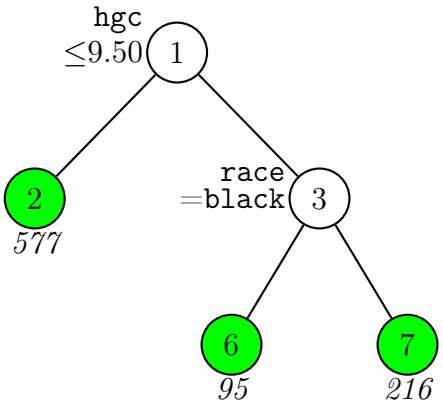

Figure 30: GUIDE v.36.0 0.00-SE regression tree for predicting longitudinal variables wage1, wage2, etc. Tree constructed with 888 observations. Maximum number of split levels is 10 and minimum node sample size is 44. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size *(in italics)* printed below nodes. Second best split variable at root node is race.

wage in the two terminal nodes. The figure is produced by the following R code.

```
z <- read.table("widewage.txt",header=FALSE)
names(z) <- c("id","hgc","exper1","exper2","exper3","exper4","exper5","exper6",
              "exper7","exper8","exper9","exper10","exper11","exper12","exper13",
              "postexp1","postexp2","postexp3","postexp4","postexp5","postexp6",
              "postexp7","postexp8","postexp9","postexp10","postexp11","postexp12",
              "postexp13","wage1","wage2","wage3","wage4","wage5","wage6","wage7",
              "wage8","wage9","wage10","wage11","wage12","wage13","ged1","ged2",
              "ged3","ged4","ged5","ged6","ged7","ged8","ged9","ged10","ged11",
              "ged12","ged13","uerate1","uerate2","uerate3","uerate4","uerate5",
              "uerate6","uerate7","uerate8","uerate9","uerate10","uerate11",
              "uerate12","uerate13","race")
exper <- c(z$exper1,z$exper2,z$exper3,z$exper4,z$exper5,z$exper6,z$exper7,
           z$exper8,z$exper9,z$exper10,z$exper11,z$exper12,z$exper13)
wage <- c(z$wage1,z$wage2,z$wage3,z$wage4,z$wage5,z$wage6,z$wage7,z$wage8,
          z$wage9,z$wage10,z$wage11,z$wage12,z$wage13)
xr <- range(exper,na.rm=TRUE)
yr <- range(wage,na.rm=TRUE)

guide.fit <- read.table("wage.fit",header=TRUE)
g.node <- guide.fit$node
g.start <- guide.fit$t.start
g.end <- guide.fit$t.end
n <- length(g.node)
```

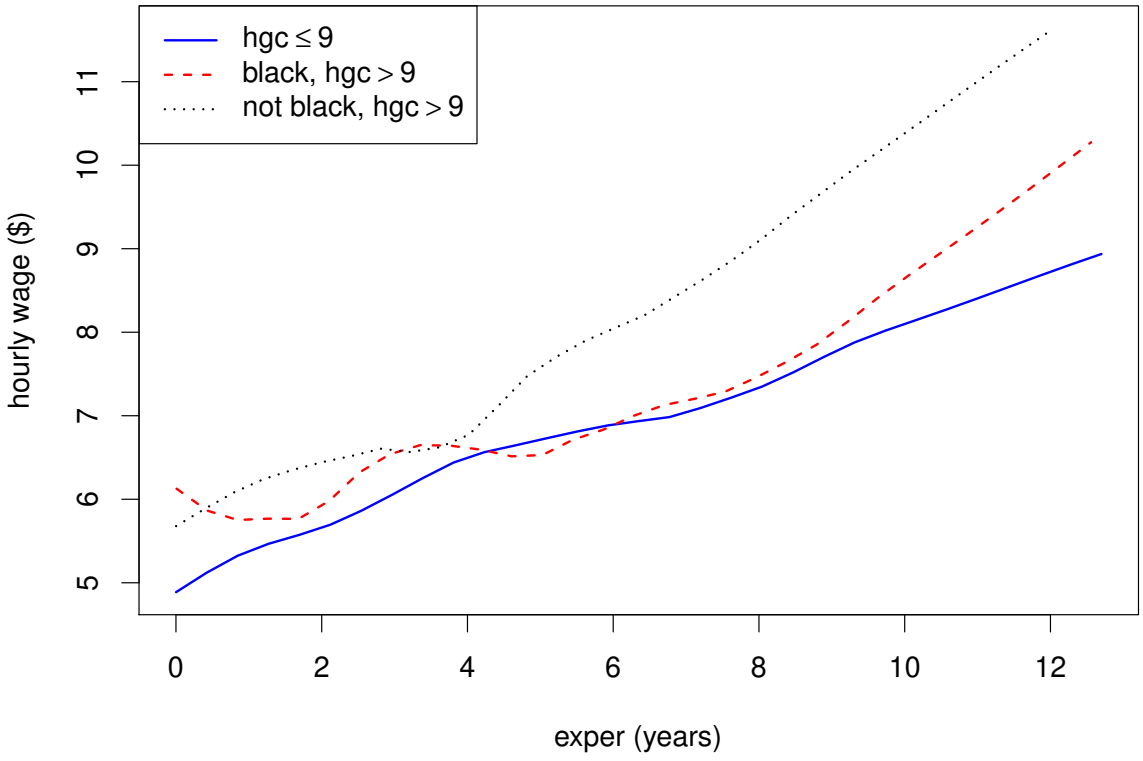

Figure 31: Lowess-smoothed mean wage curves in the terminal nodes of Figure 30.

```
m <- dim(guide.fit)[2]
npts <- m-3  # number of time points for plotting

xvals <- guide.fit[,2:3]
xvals <- as.numeric(unlist(xvals))
yvals <- guide.fit[,4:m]
yvals <- as.numeric(unlist(yvals))
plot(range(xvals),range(yvals),type="n",xlab="exper (years)",ylab="hourly wage ($)")
leg.col <- c("blue","red","black")
leg.lty <- c(1,2,3)
for(i in 1:n){
    node <- g.node[i]
    start <- g.start[i]
    end <- g.end[i]
    gap <- (end-start)/(npts-1)
    x <- start+(0:(npts-1))*gap
    y <- as.numeric(guide.fit[i,4:m])
    lines(x,y,col=leg.col[i],lty=leg.lty[i])
}
leg.txt <- c(expression(paste("hgc" <= 9)),
             expression(paste("black, hgc" > 9)),
             expression(paste("not black, hgc" > 9))
             )
legend("topleft",legend=leg.txt,lty=leg.lty,col=leg.col,lwd=2)
```

The plotting values are obtained from the result file `wage.fit` whose contents are given below. The first column gives the node number and the next two columns the start and end of the times at which fitted values are computed. The other columns give the fitted values equally spaced between the start and end times.

node	t.start	t.end	fitted1	fitted2	fitted3	fitted4	fitted5	fitted6	fitted7	fitted8	fitted9	fitted10
2	0.10000E-02	0.12700E+02	0.48875E+01	0.51221E+01	0.53241E+01	0.54668E+01	0.55738E+01					
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 on the NHTSA data using the data and description files `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
Option 7 selects logistic regression
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 ...
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 categorical variable values ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
     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
      :
    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: 0.5000

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
     5  0.2128E+03   CTRL2
     :
    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
** 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
         5T     357      357     2  3.361E-01  1.192E+00 TRANSM   +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: 9
Total number of nodes of final tree: 17
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: HIC2 proportion of 1s = 0.33613445
 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-001
       Node 13: BARSHP /= "LCB"
         Node 27: HIC2 proportion of 1s = 0.24725275E-001
   Node 3: BARSHP /= "LCB", "POL", "US2", "US3"
     Node 7: HIC2 proportion of 1s = 0.87124879E-002

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

 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-001
 ----------------------------
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: Terminal node
```

```
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -40.62       -2.609       0.9472E-02
 VEHSPD       0.7117        2.575       0.1044E-01    39.60        55.47        57.10
 Proportion of ones in variable HIC2 = 0.336134
 ----------------------------
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-001
 ----------------------------
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-001
 ----------------------------
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-002
 ----------------------------
 Observed and fitted values are stored in logits.fit
 LaTeX code for tree is in logits.tex
 R code is stored in logits.r
```

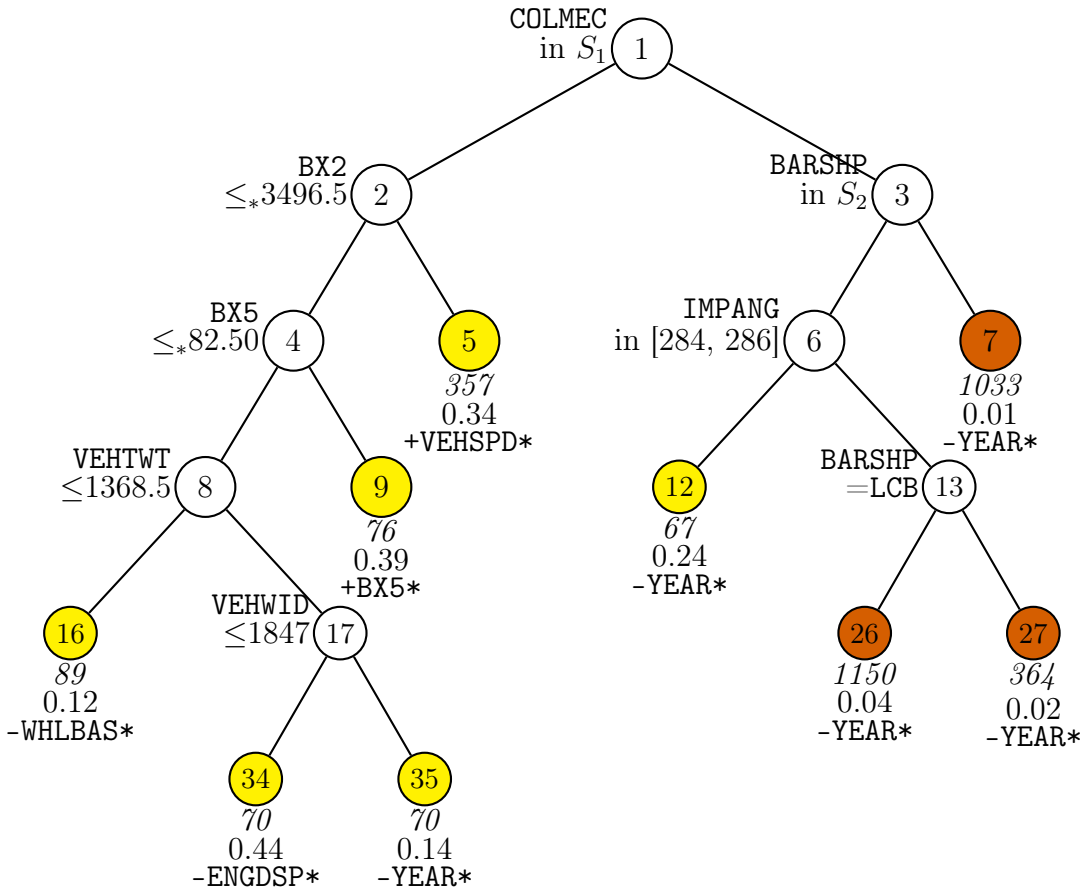

Figure 32: GUIDE v.36.2 0.50-SE piecewise simple linear logistic regression tree for predicting HIC2. Tree constructed with 3276 observations (excluding observations with non-positive weight or missing values in D, T, R or Z variables). Maximum number of split levels is 13 and minimum node sample size 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 = {LCB, POL, US2, US3}. Sample size (*in italics*), proportion of 1s in HIC2, and sign and 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. An asterisk at end of name of regressor indicates its slope is significant at the 0.05 level (unadjusted for post selection). Second best split variable at root node is BMPENG.

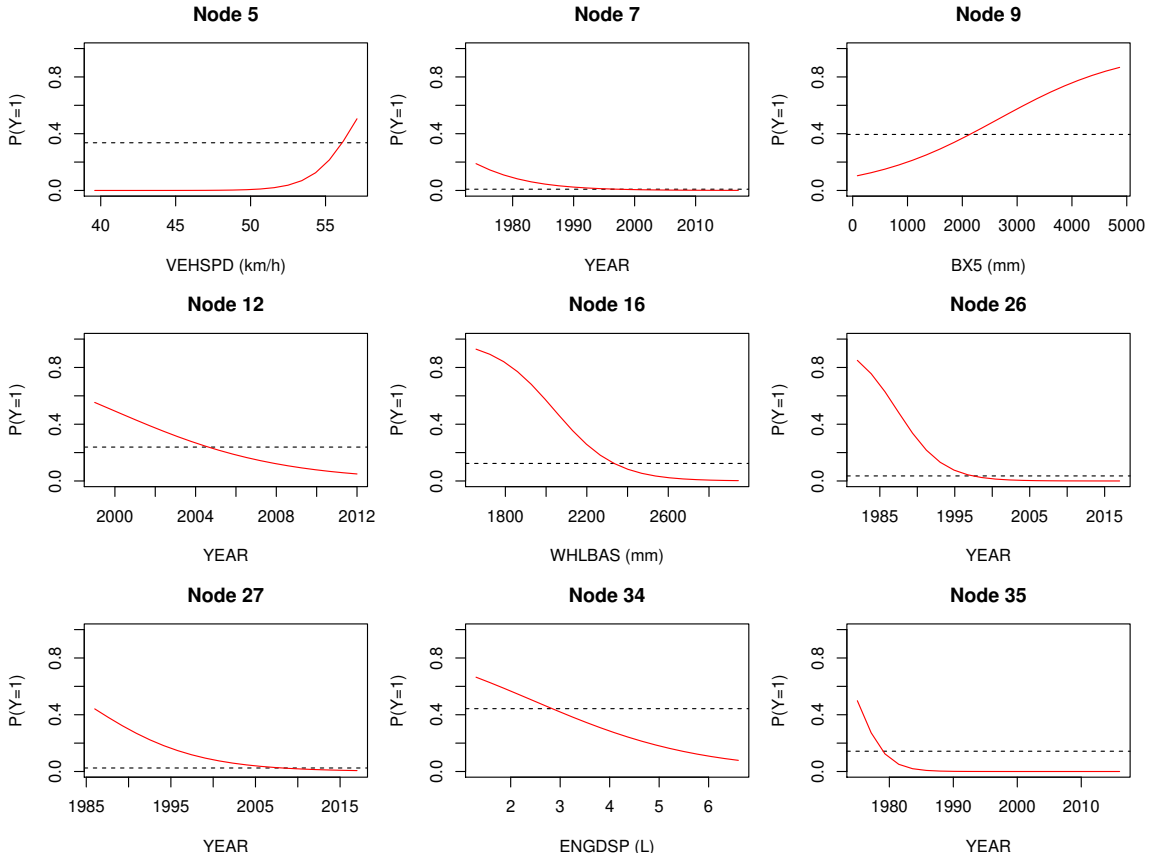

Figure 33: Estimated logistic regression curves in terminal nodes of tree in Figure 32. Horizontal dashed line marks proportion of head injury in node.

```
z1 <- read.csv("nhtsadata.csv",header=TRUE)
z2 <- read.table("logits.fit",header=TRUE)
par(mfrow=c(3,3),mar=c(4,4,3,1),cex=0.9)
nvarid <- 1:dim(z1)[2]
nodes <- unique(sort(z2$node))
xnames <-c("VEHSPD","YEAR","BX5","YEAR","WHLBAS","YEAR","YEAR","ENGDSP","YEAR")
xlabs <-c("VEHSPD␣(km/h)","YEAR","BX5␣(mm)","YEAR","WHLBAS␣(mm)",
          "YEAR","YEAR","ENGDSP␣(L)","YEAR")
titles.txt <- paste("Node",nodes)
i <- 0
for(node in nodes){
    i <- i+1
    tmp <- names(z1) %in% xnames[i]
    xid <- nvarid[tmp]
    gp <- z2$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 34: R code for Figure 33

Figure 32 shows the logistic regression tree and Figure 33 shows the fitted logistic curves in the terminal nodes. The R code for the plots is given in Figure 34.

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
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 categorical variable values ...
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       3443       13        0        0       20
    #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: 50
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
 2.287E+01     1.00  cat1
 2.234E+01     2.00  aps1
 2.012E+01     3.00  crea1
 1.924E+01     4.00  pafi1
 1.855E+01     5.00  meanbp1
 1.219E+01     6.00  alb1
 1.196E+01     7.00  neuro
 1.066E+01     8.00  card
 1.021E+01     9.00  hema1
 1.010E+01    10.00  cat2
 9.315E+00    11.00  wtkilo1
 7.929E+00    12.00  seps
 6.409E+00    13.00  resp
 6.073E+00    14.00  dnr1
 6.029E+00    15.00  bili1
 5.410E+00    16.00  paco21
 4.260E+00    17.00  hrt1
 3.979E+00    18.00  transhx
 3.860E+00    19.00  chrpulhx
 3.860E+00    20.00  resp1
 3.311E+00    21.00  ninsclas
 3.183E+00    22.00  dementhx
 2.969E+00    23.00  ph1
 2.163E+00    24.00  psychhx
 2.029E+00    25.00  renal
 1.913E+00    26.00  gastr
 1.896E+00    27.00  income
 1.748E+00    28.00  cardiohx
 1.412E+00    29.00  urin1
 1.386E+00    30.00  trauma
 1.247E+00    31.00  age
 1.194E+00    32.00  sex
--------- variables above this line are highly important -----
 1.188E+00    33.00  edu
 1.176E+00    34.00  sod1
 1.056E+00    35.00  immunhx
--------- variables below this line are unimportant ----------
 9.441E-01    36.00  malighx
 9.116E-01    37.00  wblc1
 8.732E-01    38.00  ca
 8.616E-01    39.00  amihx
 8.071E-01    40.00  scoma1
 6.766E-01    41.00  chfhx
 5.859E-01    42.00  gibledhx
```

```
 4.112E-01    43.00  renalhx
 4.095E-01    44.00  pot1
 3.971E-01    45.00  ortho
 3.431E-01    46.00  liverhx
 3.412E-01    47.00  hema
 3.280E-01    48.00  meta
 2.586E-01    49.00  temp1
 1.296E-01    50.00  race

Variables with scores above 1.19 are highly important
Variables with scores between 1.0 and 1.19 are likely important
Variables with scores below 1.0 are unimportant
No. highly important, likely important, and unimportant split variables: 32, 3, 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 partial 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      2.287E+01     cat1
 H      2.234E+01     aps1
 H      2.012E+01     crea1
 H      1.924E+01     pafi1
 H      1.855E+01     meanbp1
 H      1.219E+01     alb1
 H      1.196E+01     neuro
 :
 H      1.194E+00     sex
 L      1.188E+00     edu
 L      1.176E+00     sod1
 L      1.056E+00     immunhx
 U      9.441E-01     malighx
 U      9.116E-01     wblc1
 :
 U      2.586E-01     temp1
 U      1.296E-01     race
```

Figure 35 shows a barplot of the scores. It is made by the following R code.

```
par(mar=c(5,6,2,1),las=1)
leg.col <- c("yellow","magenta","white")
```

```
leg.txt <- c("highly important","likely important","unimportant")
x <- read.table("imp.scr",header=TRUE)
score <- x$Score
vars <- x$Variable
type <- x$Type
barcol <- rep("yellow",length(vars))
barcol[type == "L"] <- "magenta"
barcol[type == "U"] <- "white"
barplot(rev(score),names.arg=rev(vars),col=rev(barcol),horiz=TRUE,
        xlab="GUIDE importance scores")
abline(v=1,col="red",lty=2)
legend("bottomright",legend=leg.txt,fill=leg.col,cex=1.2)
```

Figure 36 shows the classification tree from `imp.tex` that produced the scores. It is an unpruned tree with four levels of splits.

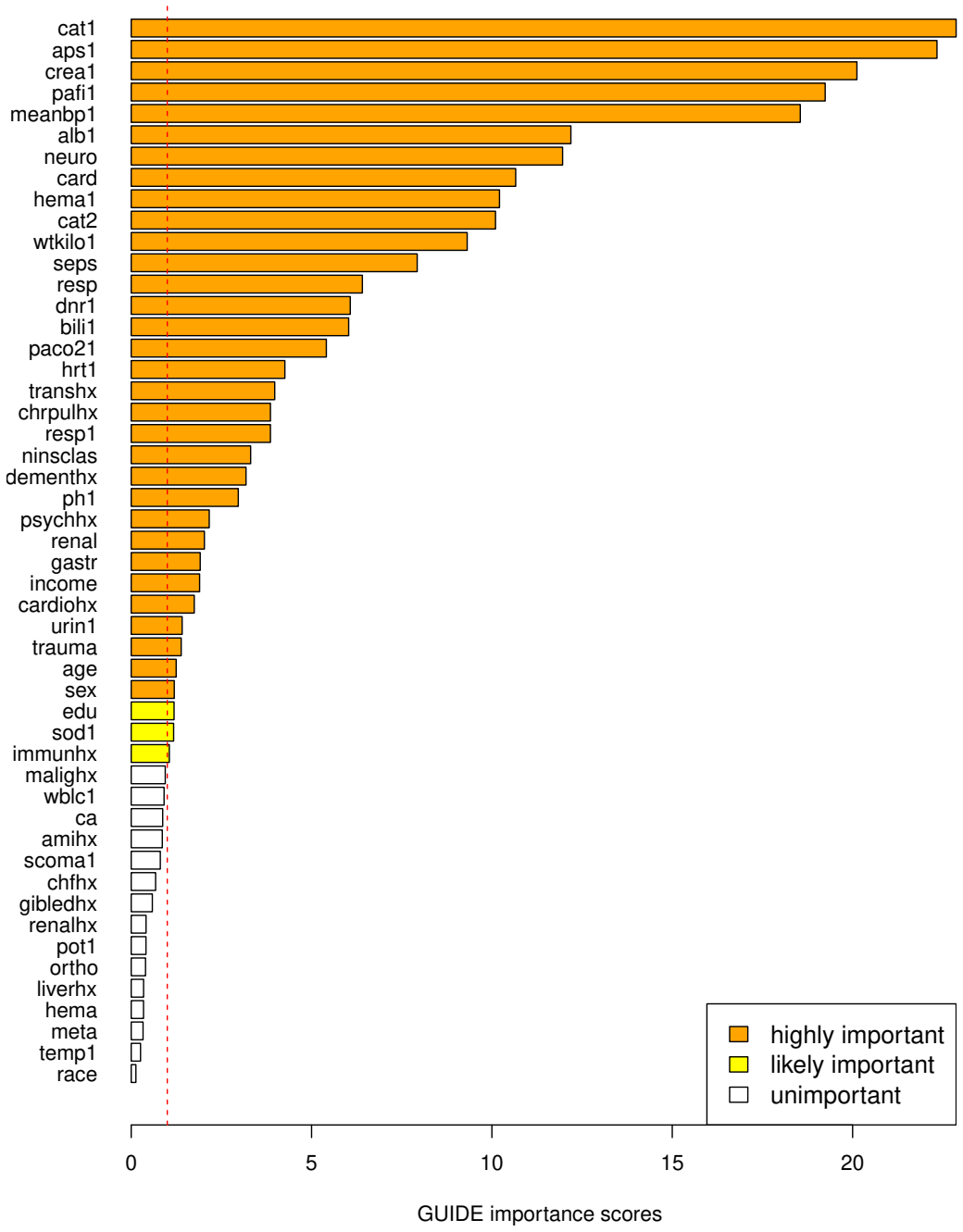

Figure 35: Importance scores for predicting `swang1`; variables with bars shorter than 1.0 (indicated by the red dashed line) are considered unimportant.

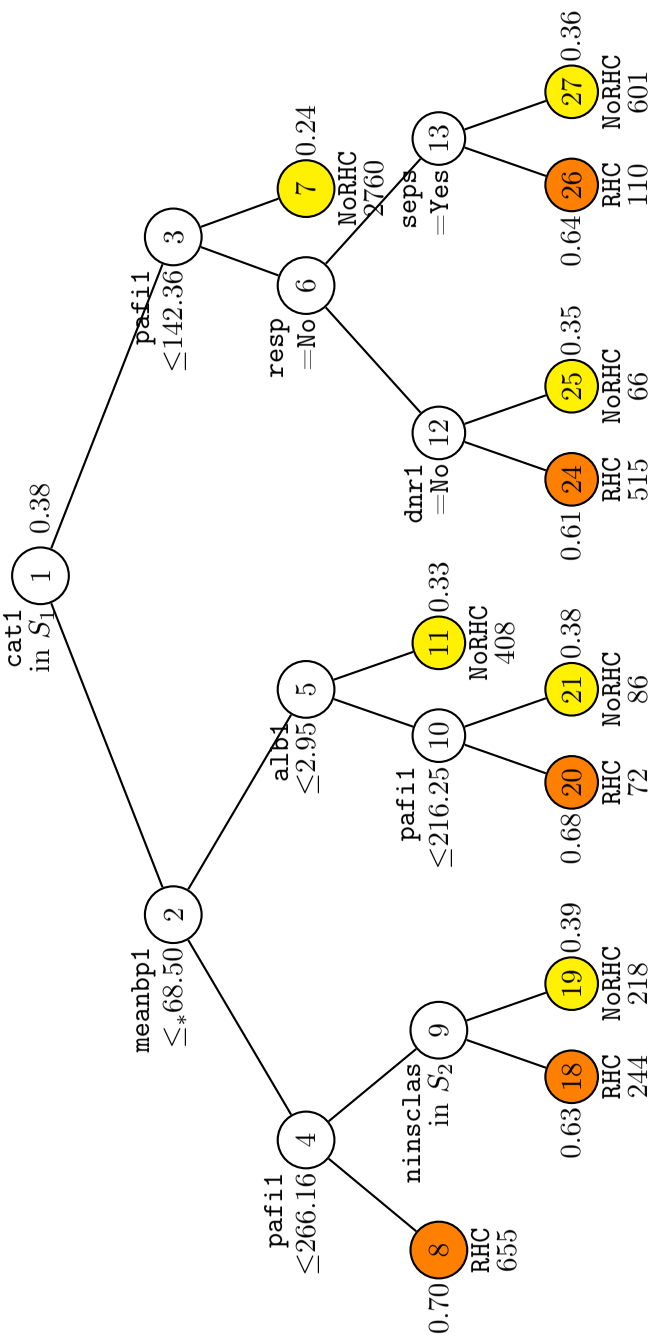

Figure 36: GUIDE v.36.2 importance scoring classification tree for predicting `swang1` using estimated priors and unit misclassification costs. Tree constructed with 5735 observations. Maximum number of split levels is 4 and minimum node sample size is 57. At each split, an observation goes to the left 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.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 ...
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 categorical variable values ...
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
```

```
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
treatment variable is not randomized
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       3443       11        0        0       20
    #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: .649
Number of cases used for training: 5735
Number of split variables: 50
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 following output shows that there is only one important variable. Figure 37 shows the tree that produced the scores.

```
Scaled importance scores of predictor variables
 (F, I and R variables are excluded)
     Score     Rank  Variable
 1.091E+00     1.00  dnr1
--------- variables below this line are unimportant ----------
 9.095E-01     2.00  ph1
 8.341E-01     3.00  chrpulhx
 7.199E-01     4.00  paco21
 6.665E-01     5.00  resp1
 5.124E-01     6.00  liverhx
 4.448E-01     7.00  pot1
 3.899E-01     8.00  age
 3.894E-01     9.00  gastr
 3.694E-01    10.00  cat2
 3.392E-01    11.00  gibledhx
 3.206E-01    12.00  pafi1
 3.154E-01    13.00  hrt1
 3.102E-01    14.00  aps1
 2.766E-01    15.00  amihx
 2.372E-01    16.00  edu
 2.342E-01    17.00  malighx
 2.067E-01    18.00  income
 1.900E-01    19.00  meanbp1
 1.883E-01    20.00  scoma1
 1.644E-01    21.00  ortho
 1.599E-01    22.00  alb1
 1.510E-01    23.00  crea1
 1.498E-01    24.00  urin1
 1.498E-01    25.00  neuro
 1.469E-01    26.00  sex
 1.467E-01    27.00  hema1
 1.434E-01    28.00  ninsclas
 1.356E-01    29.00  temp1
 1.355E-01    30.00  race
 1.232E-01    31.00  psychhx
 1.201E-01    32.00  renalhx
 1.183E-01    33.00  meta
 1.146E-01    34.00  chfhx
```

```
 1.084E-01    35.00  ca
 1.079E-01    36.00  wblc1
 1.050E-01    37.00  hema
 9.448E-02    38.00  seps
 9.144E-02    39.00  wtkilo1
 8.568E-02    40.00  trauma
 7.713E-02    41.00  cardiohx
 6.760E-02    42.00  cat1
 6.077E-02    43.00  resp
 5.724E-02    44.00  immunhx
 5.450E-02    45.00  sod1
 5.230E-02    46.00  card
 5.123E-02    47.00  dementhx
 4.201E-02    48.00  renal
 3.128E-02    49.00  bili1
 3.019E-02    50.00  transhx

Variables with scores above 2.27 are highly important
Variables with scores between 1.0 and 2.27 are likely important
Variables with scores below 1.0 are unimportant
No. highly important, likely important, and unimportant split variables: 0, 1, 49
LaTeX code for tree is in imp_surv.tex
Importance scores are stored in imp_surv.scr
```

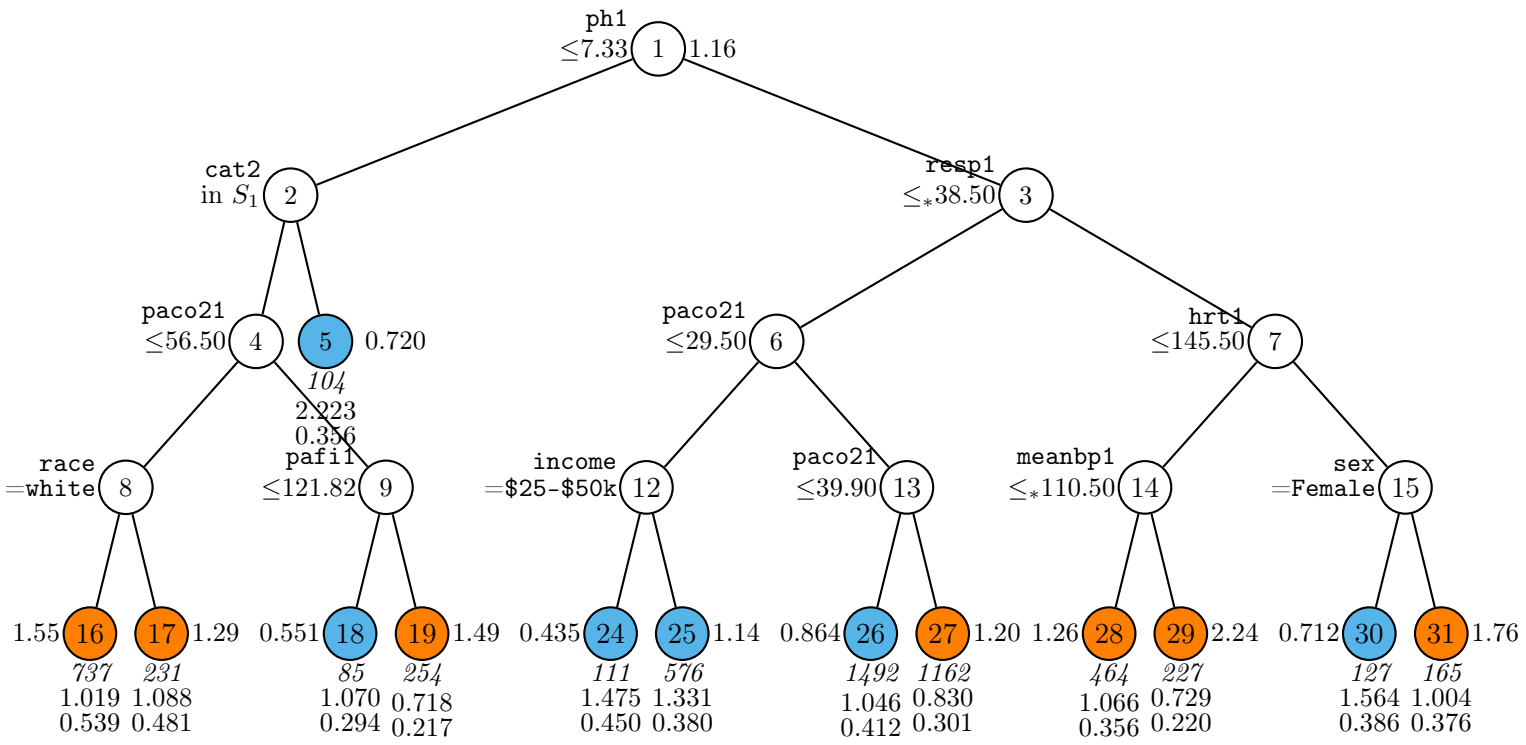

Figure 37: GUIDE v.36.2 importance scoring proportional hazards regression tree using Gi option for `survtime` and event indicator `death` without linear prognostic effects. Tree constructed with 5735 observations. Maximum number of split levels is 4, minimum node sample size is 57 and minimum treatment 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 '$\leq$ or missing'. Set S_1 = {`MOSF w/Sepsis`, `NA`}. Treatment `swang1` hazard ratio of level `RHC` to `NoRHC` beside nodes. Sample size (*in italics*), relative baseline hazard (relative to that at root node), and proportion of `swang1` = `RHC` printed below nodes. Terminal nodes with hazard ratios above and below value at root node are colored orange and skyblue, respectively. Second best split variable at root node is `chrpulhx`.

Figure 38 plots the scores if the Gs method is used. The reason there are many important variables now is due to the Gs method picking up prognostic as well as predictive variables.

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 $\pi(x) = P(Z = 1 \,|\, X = x)$. 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)*

$$n^{-1} \sum_{i=1}^{n} \left\{ \frac{Z_i Y_i}{\hat{\pi}(X_i)} - \frac{(1 - Z_i) Y_i}{1 - \hat{\pi}(X_i)} \right\}$$

or the *Hájek inverse probability estimate (IPW)*

$$\frac{\sum_i Z_i Y_i / \hat{\pi}(X_i)}{\sum_i Z_i / \hat{\pi}(X_i)} - \frac{\sum_i (1 - Z_i) Y_i / (1 - \hat{\pi}(X_i))}{\sum_i (1 - Z_i) / (1 - \hat{\pi}(X_i))}$$

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*

$$n^{-1} \sum_t n_t \hat{\beta}_t$$

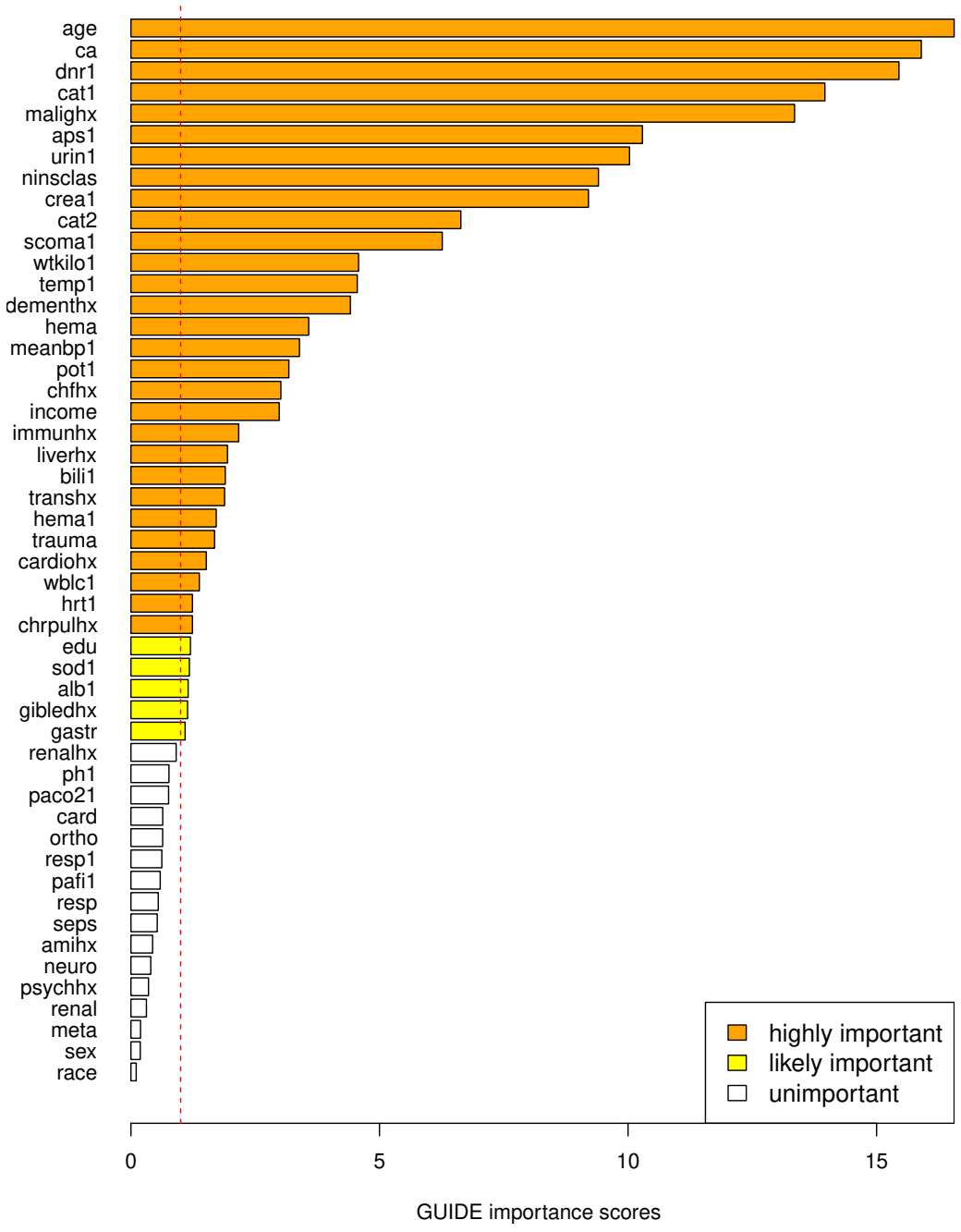

Figure 38: Importance scores for relative risk using Gs method; variables with bars shorter than 1.0 (indicated by the red dashed line) are considered unimportant.

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) and `death` (1=death within 6 months). 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 ...
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 categorical variable values ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
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       3443       12        0        0       32
    #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: 50
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
```

```
32 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
     10  cardiohx   s    0.000        1.000
      :
     24  edu        s    0.000        30.00
     28  dth30      d    0.000        1.000
     29  aps1       s    3.000        147.0
      :
     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       3443       12        0        0       32
    #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: 50
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: 0.5000

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: 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
     6  0.1482E+03   neuro
     7  0.1247E+03   alb1
     8  0.1178E+03   card
     9  0.1077E+03   hema1
    10  0.9651E+02   wtkilo1
     :
    48  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       77   3.288E-01   6.446E-03   5.749E-03   3.254E-01   9.153E-03
   2       76   3.288E-01   6.446E-03   5.749E-03   3.254E-01   9.153E-03
   :
  37       30   3.237E-01   6.391E-03   5.073E-03   3.196E-01   7.756E-03
  38*      27   3.233E-01   6.384E-03   4.903E-03   3.196E-01   6.791E-03
  39+      26   3.238E-01   6.388E-03   4.683E-03   3.196E-01   6.147E-03
  40       25   3.239E-01   6.374E-03   5.178E-03   3.204E-01   6.668E-03
  41       20   3.237E-01   6.340E-03   5.721E-03   3.204E-01   6.875E-03
  42++     19   3.240E-01   6.365E-03   5.096E-03   3.221E-01   6.470E-03
  43**     17   3.257E-01   6.370E-03   4.672E-03   3.241E-01   6.086E-03
  44       15   3.275E-01   6.373E-03   4.927E-03   3.252E-01   5.232E-03
   :
  53        4   3.690E-01   6.337E-03   7.280E-03   3.705E-01   9.859E-03
  54        2   4.131E-01   5.710E-03   3.745E-03   4.112E-01   3.751E-03
  55        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 27 terminal nodes
0-SE tree based on median is marked with + and has 26 terminal nodes
```

```
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as -- tree

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

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

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

Node label	Total cases	Train cases	Predicted class	Node cost	Split variables	Interacting variable
1	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	resp	
34	231	231	RHC	4.772E-01	paco21	
68T	63	63	RHC	3.298E-01	-	
69	168	168	NoRHC	4.616E-01	resp1	
138T	58	58	RHC	4.139E-01	-	
139T	110	110	NoRHC	3.895E-01	-	
35T	222	222	NoRHC	3.432E-01	bili1 :aps1	
9T	1395	1395	RHC	3.044E-01	resp1	
5	2354	2354	NoRHC	4.682E-01	cat1	
10	1076	1076	RHC	4.030E-01	meanbp1	
20T	798	798	RHC	3.358E-01	bili1	
21	278	278	NoRHC	3.753E-01	resp1	
42T	64	64	RHC	4.260E-01	-	
43T	214	214	NoRHC	3.081E-01	scoma1 :age	
11	1278	1278	NoRHC	3.462E-01	cat2	
22	291	291	RHC	4.813E-01	wtkilo1	
44T	108	108	NoRHC	3.287E-01	-	
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	-	
15	196	196	NoRHC	4.635E-01	ph1 :paco21	
30	124	124	RHC	4.682E-01	resp	
60T	62	62	RHC	3.657E-01	-	
61T	62	62	NoRHC	4.181E-01	-	
31T	72	72	NoRHC	3.345E-01	-	

```
Number of terminal nodes of final tree: 17
Total number of nodes of final tree: 33
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: resp = "No"
          Node 34: paco21 <= 32.199220
            Node 68: RHC
          Node 34: paco21 > 32.199220 or NA
            Node 69: resp1 <= 14.000000
              Node 138: RHC
            Node 69: resp1 > 14.000000 or NA
              Node 139: NoRHC
        Node 17: resp /= "No"
          Node 35: NoRHC
    Node 4: crea1 > 1.2498779 or NA
      Node 9: RHC
  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: resp1 <= 21.000000
          Node 42: RHC
        Node 21: resp1 > 21.000000 or NA
          Node 43: 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: ph1 <= 7.3997060
         Node 30: resp = "No"
           Node 60: RHC
         Node 30: resp /= "No"
           Node 61: NoRHC
       Node 15: ph1 > 7.3997060 or NA
         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 61: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.4419        5.730       0.3452E-06
 swang1.RHC    0.8446E-01   0.6063       0.5466
 Number of observations in node = 62
 ----------------------------
Node 31: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.4909        7.255       0.4275E-09
 swang1.RHC    0.1561        1.121       0.2660
 Number of observations in node = 72
 ----------------------------
 Regression estimates are weighted means over terminal nodes
 Regressor     Coefficient  z-stat       p-value
 Constant      0.3085        37.65        0.000
 swang1.RHC    0.6151E-01    4.239       0.2243E-04

 Average treatment effect of swang1 level "RHC" vs level "NoRHC" =  6.1511E-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 39. The pair of numbers beside each terminal node are the proportions of observations with `swang1 = NoRHC` ($Z = 0$) and `swang1 = RHC` ($Z = 1$), respectively. 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 is alphabetically before "`RHC`".

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      9   "NoRHC"  "RHC"      0.41577E+00    0.58423E+00
     y     20   "RHC"    "RHC"      0.45113E+00    0.54887E+00
```

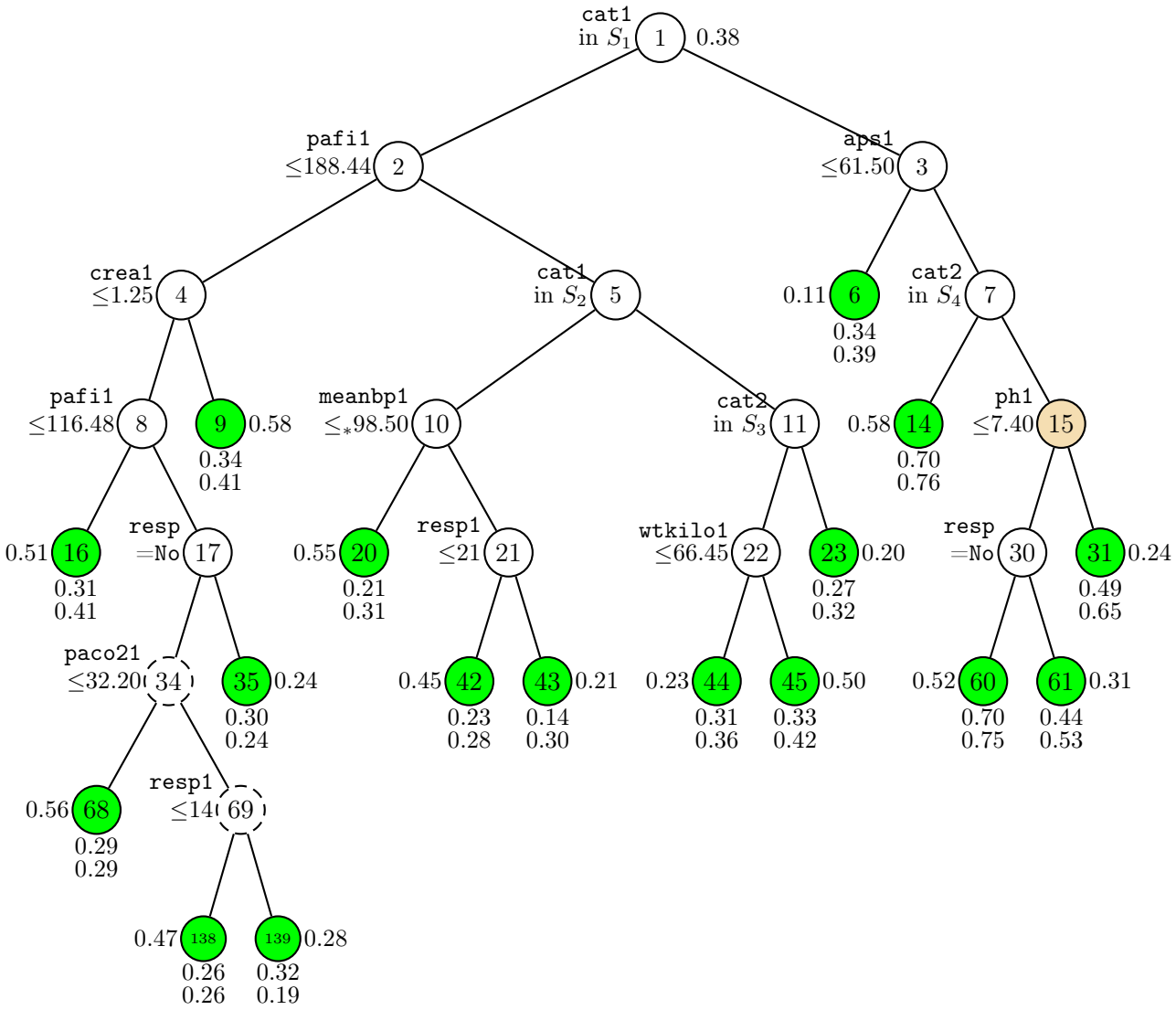

Figure 39: GUIDE v.36.2 0.50-SE tree for propensity score grouping and estimation of effects of `swang1` on `dth30`. Tree constructed with 5735 observations. Maximum number of split levels is 15 and minimum node sample size is 57. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. Set S_1 = {`ARF`, `CHF`, `MOSF w/Malignancy`, `MOSF w/Sepsis`}. Set S_2 = {`CHF`, `MOSF w/Sepsis`}. Set S_3 = {`MOSF w/Sepsis`}. Set S_4 = {`MOSF w/Sepsis`}. Circles with dashed lines are nodes with no significant split variables. Intermediate nodes with splits due to interaction are in wheat color. Sample means of `dth30` for `swang1` levels `NoRHC` and `RHC`, respectively, printed below nodes. Sample proportions of the corresponding levels of `swang1` 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
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 ...
Missing values found in D variables
Assigning integer codes to categorical variable values ...
Re-checking data ...
```

```
Allocating missing value information ...
Assigning codes to categorical and missing values ...
Data checks complete
Creating missing value indicators ...
Some D variables have missing values
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 40 shows the tree.

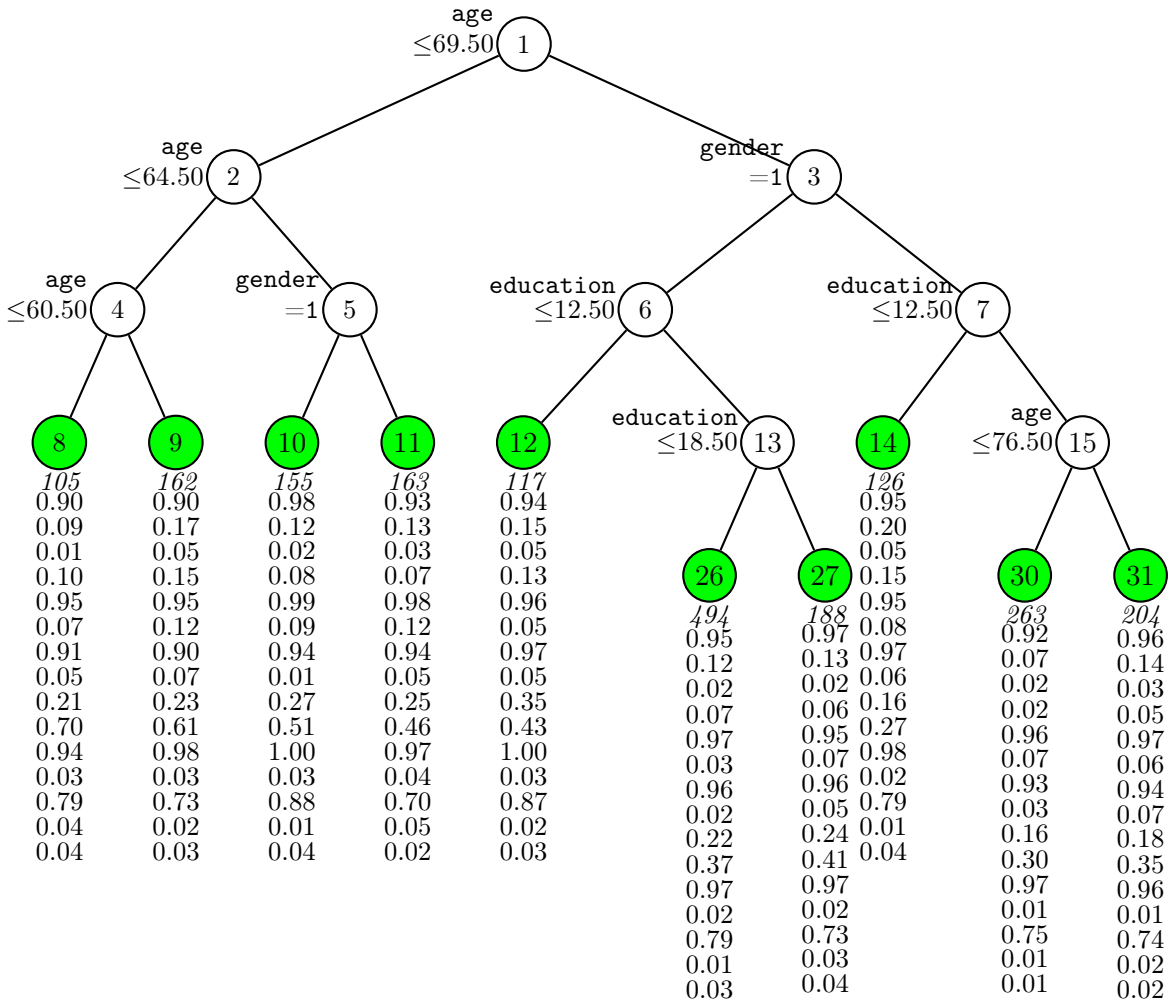

Figure 40: GUIDE v.36.2 importance scoring or DIF regression tree for predicting response variables `satis`, `drop`, `empty`, `bored`, `spirit`, `afraid`, `happy`, `help`, `home`, `memory`, `alive`, `worth`, `energy`, `hope`, and `better`, without using PCA at each node. Tree constructed with 1977 observations (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)
```

```
47              sd <- sqrt(p*(1-p)/(ntot+1))
48          } else {
49              sd <- sqrt(estp*(1-estp)/ntot)
50          }
51          gp <- bfit$node == unodes[j] & bfit$train == "n"  ## real data
52          n0 <- sum(bfit$observed[gp] != class.name)
53          n1 <- sum(bfit$observed[gp] == class.name)
54          realp <- n1/(n0+n1)
55          zstat[i] <- max(zstat[i],abs(realp-estp)/sd)
56      }
57      if(i %% 100 == 0){
58          sd.err <- sqrt(var(err.test[1:i])) ## linear interpolation
59          q <- quantile(zstat[1:i],probs=probs,type=4)
60          write(c(i,q,bias/i,sd.err),"results.txt",append=TRUE,ncol=7)
61      }
62  }
63  ### find calibrated z.alpha
64  write(paste("No. bootstraps =",nboot),"results.txt",append=TRUE)
65  write(c("Calibrated z at levels",probs),file="results.txt",ncol=5,
66        append=TRUE)
67  q <- quantile(zstat,probs=probs,type=4) ## linear interpolation
68  write(q,"results.txt",append=TRUE,ncol=4)
69  write(paste("Bootstrap estimate of bias of error rate =",bias/nboot),
70        "results.txt",append=TRUE)
71  write(paste("Bootstrap estimate of SD of error rate =",
72        sqrt(var(err.test))),"results.txt",append=TRUE)
73  ### fit real data
74  system("rm -f log.txt real.out real.fit")
75  system("guide < real.in > log.txt")
76  realfit <- read.table("real.fit",header=TRUE)
77  train <- realfit$train == "y"
78  err.obs <- sum(realfit$observed[train] != realfit$predicted[train])/nobs
79  write(paste("Real data observed error rate =",err.obs),"results.txt",
80        append=TRUE)
81  k <- 3      ## 95% level
82  z0 <- q[k] ## 95% z value
83  write(c("Simultaneous intervals at level",probs[k]),
84        file="results.txt",ncol=2,append=TRUE)
85  write(paste0("Node  N   P(",class.name,") halfwid   left   right"),
86        "results.txt", append=TRUE)
87  unodes <- unique(sort(realfit$node))
88  for(j in 1:length(unodes)){
89      gp <- realfit$node == unodes[j] & realfit$train == "y"
90      n0 <- sum(realfit$observed[gp] != class.name)
91      n1 <- sum(realfit$observed[gp] == class.name)
92      ntot <- n0+n1
```

```
93      if(n1 == 0 | n0 == 0){
94          p <- (n1+0.5)/(ntot+1)
95          sd <- sqrt(p*(1-p)/(ntot+1))
96      } else {
97          p <- n1/ntot
98          sd <- sqrt(p*(1-p)/(ntot))
99      }
100     p <- n1/ntot
101     halfwid <- z0*sd
102     left <- p-halfwid
103     rght <- p+halfwid
104     write(c(unodes[j],ntot,p,halfwid,left,rght),"results.txt",
105           append=TRUE,ncol=6)
106 }
107 ## write(sort(zstat),"zstat.txt",ncol=1) ## output sorted zstat values
```

Figure 41 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 42.

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.

Figure 41: Contents of `results.txt`

```
Bootstrap simultaneous intervals by linear interpolation of z
trials  z80    z90    z95     z98      bias.err     sd.err
100 3.862114 4.306968 5.099053 6.159524 -0.02995466 0.006104871
200 4.016427 4.443295 5.045085 5.615615 -0.03068963 0.006230428
300 4.027601 4.418647 5.045085 5.491243 -0.03084394 0.006367497
400 3.973859 4.364386 4.745885 5.337637 -0.03097777 0.00634391
500 3.973859 4.371885 4.791225 5.449224 -0.03100366 0.006192969
600 3.961602 4.361018 4.716976 5.491243 -0.03099826 0.006224448
700 3.943059 4.340042 4.68289 5.449224 -0.03097646 0.00629261
800 3.949144 4.342243 4.690964 5.337637 -0.03089712 0.006283988
900 3.938293 4.313407 4.68289 5.337637 -0.0307771 0.00633395
1000 3.961722 4.325215 4.690964 5.337637 -0.03080593 0.006287672
No. bootstraps =  1000
Calibrated z at levels 0.8 0.9 0.95 0.98
3.961722 4.325215 4.690964 5.337637
Bootstrap estimate of bias of error rate =  -0.0308059285091543
Bootstrap estimate of SD of error rate =  0.00628767196598997
Real data observed error rate =  0.296251089799477
Simultaneous intervals at level 0.95
Node  N   P(RHC) halfwid   left   right
5 566 0.3816254 0.09578518 0.2858403 0.4774106
7 2760 0.2355072 0.03788754 0.1976197 0.2733948
8 655 0.6961832 0.08429641 0.6118868 0.7804796
18 244 0.6270492 0.1452258 0.4818234 0.772275
19 218 0.3853211 0.1546213 0.2306998 0.5399424
25 66 0.3484848 0.275134 0.07335081 0.6236189
26 110 0.6363636 0.2151553 0.4212083 0.8515189
27 601 0.3627288 0.09199792 0.2707309 0.4547267
48 438 0.6552511 0.1065321 0.5487191 0.7617832
49 77 0.3506494 0.2550897 0.09555968 0.605739
```

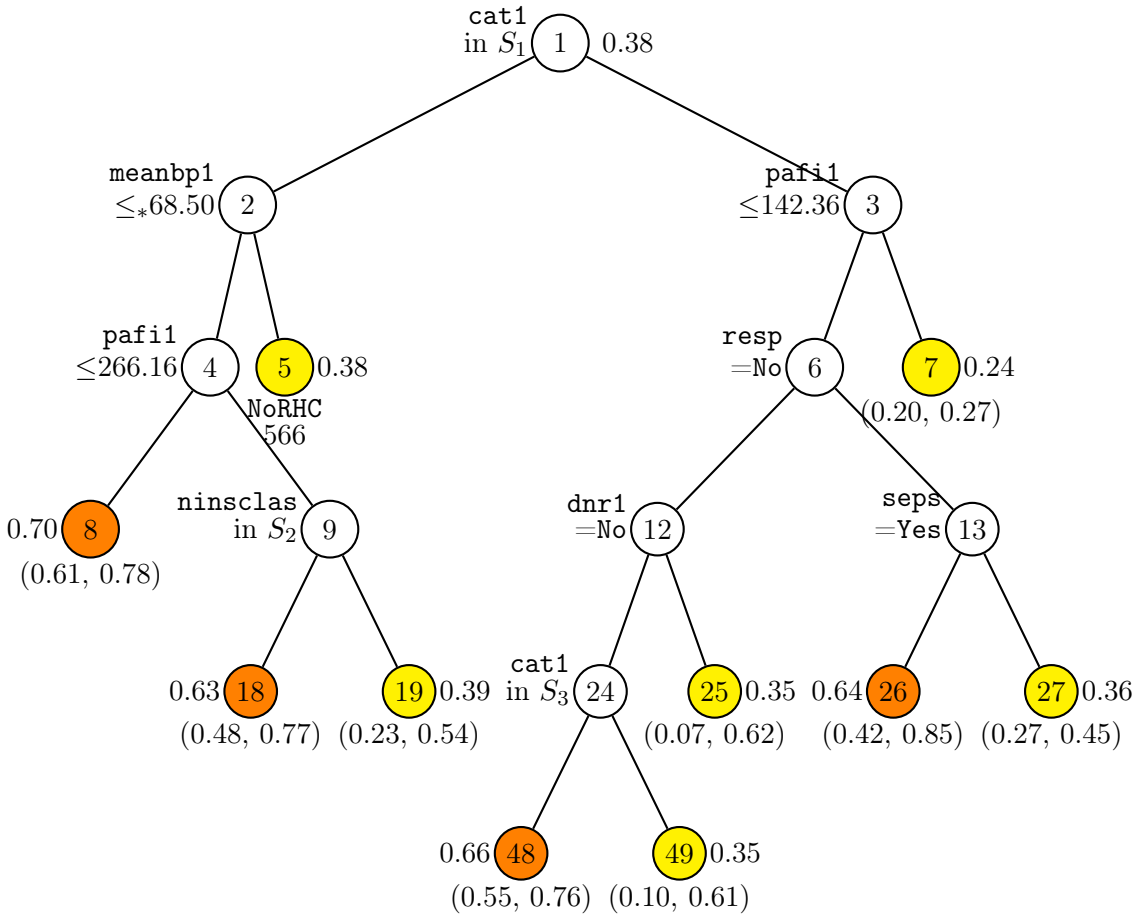

Figure 42: GUIDE v.36.2 0.50-SE classification tree for predicting `swang1` using estimated priors and unit misclassification costs. Tree constructed with 5735 observations. Maximum number of split levels is 15 and minimum node sample size is 57. At each split, an observation goes to the left branch 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`}. 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
423 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 ...
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
Finding number of levels of M variables associated with C variables ...
```

```
Assigning integer codes to categorical variable values ...
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: variable MISC2PQ is constant
Warning: variable MISC2CQ is constant
:
Warning: variable ROTHRFLC is constant
Warning: variable WELFREBX has all values missing
Warning: variable OTRINCBX has all values missing
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       15        0        0      423
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      172        0       41        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
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
weighted observations are treated equally for classification
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
423 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: variable MISC2PQ is constant
Warning: variable MISC2CQ is constant
:
Warning: variable ROTHRFLC is constant
Warning: variable WELFREBX has all values missing
Warning: variable OTRINCBX has all values missing
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
      :
     50  FINLWT21   w    1351.       0.7027E+05
     51  FJSSDEDX   s    0.000       0.3042E+05
     52  FJSS_EDX   m                                 0
      :
    507  FSMPFRMX   s  -0.4000E+06   0.1090E+07
    508  FSMP_RMX   m                                 0
```

```
    514  INTRDVX_   d                                  3
    522  IRAB       s    1.000        6.000                  4514
    523  IRAB_      m                                  2
      :
    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       15        0        0      423
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      172        0       41        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
Simple node models
Estimated priors
Unit misclassification costs
Warning: All positive weights treated as 1
weighted observations are treated equally for classification
Univariate split highest priority
No interaction 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:    139.0

Classification matrix for training sample:
Predicted       True class
class              C         D         T
C               1281        71         7
D                490      2767        77
T                  0         0         0
Total           1771      2838        84

Number of cases used for tree construction: 4693
Number misclassified: 645
Resubstitution estimate of mean misclassification cost: .1374

Number of OOB cases: 4693
Number OOB misclassified: 1034
OOB estimate of mean misclassification cost: .2203
```

```
 Mean number of trees per OOB observation: 183.82

 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   0.24037E+00  0.74350E+00  0.16130E-01    "D"    "D"
y   0.28351E+00  0.70804E+00  0.84491E-02    "D"    "D"
y   0.14221E+00  0.85411E+00  0.36831E-02    "D"    "D"
y   0.18758E+00  0.80421E+00  0.82022E-02    "D"    "D"
y   0.13307E+00  0.85283E+00  0.14099E-01    "D"    "D"
y   0.17520E+00  0.74845E+00  0.76346E-01    "D"    "D"
y   0.58929E+00  0.39981E+00  0.10908E-01    "C"    "C"
y   0.44573E+00  0.52668E+00  0.27590E-01    "D"    "D"
y   0.20441E+00  0.78793E+00  0.76684E-02    "D"    "D"
y   0.13832E+00  0.85621E+00  0.54729E-02    "D"    "D"
y   0.55575E+00  0.43043E+00  0.13820E-01    "C"    "C"
y   0.28918E+00  0.69421E+00  0.16614E-01    "D"    "D"
y   0.43310E+00  0.56198E+00  0.49221E-02    "D"    "D"
y   0.33338E+00  0.63876E+00  0.27861E-01    "D"    "D"
y   0.23832E+00  0.73837E+00  0.23305E-01    "D"    "D"
y   0.22788E+00  0.60925E+00  0.16287E+00    "D"    "T"
y   0.61780E+00  0.38146E+00  0.74131E-03    "C"    "D"
```

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
423 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 ...
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
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to categorical variable values ...
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to categorical and missing values ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: variable MISC2PQ is constant
Warning: variable MISC2CQ is constant
Warning: variable TCARTRKP is constant
Warning: variable TCARTRKC is constant
Warning: variable TOTHVHRP is constant
Warning: variable TOTHVHRC is constant
Warning: variable VMISCHEP is constant
Warning: variable VMISCHEC is constant
Warning: variable ROTHRFLP is constant
Warning: variable ROTHRFLC is constant
Warning: variable WELFREBX has all values missing
Warning: variable OTRINCBX has all values missing
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       15        0        0      423
    #P-var   #M-var   #B-var   #C-var   #I-var
```

```
         0      172        0       41        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
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
weighted observations are treated equally for classification
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
423 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: variable MISC2PQ is constant
Warning: variable MISC2CQ is constant
Warning: variable TCARTRKP is constant
Warning: variable TCARTRKC is constant
Warning: variable TOTHVHRP is constant
Warning: variable TOTHVHRC is constant
Warning: variable VMISCHEP is constant
Warning: variable VMISCHEC is constant
Warning: variable ROTHRFLP is constant
Warning: variable ROTHRFLC is constant
Warning: variable WELFREBX has all values missing
Warning: variable OTRINCBX has all values missing
```

```
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
      :
    507  FSMPFRMX   s  -0.4000E+06   0.1090E+07
    508  FSMP_RMX   m                                 0
    514  INTRDVX_   d                                 3
    522  IRAB       s    1.000        6.000                  4514
    523  IRAB_      m                                 2
      :
    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       15        0        0      423
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      172        0       41        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: 0.5000

Simple node models
```

```
Estimated priors
Unit misclassification costs
Warning: All positive weights treated as 1
Univariate split highest priority
Interaction and linear splits 2nd and 3rd priorities
Fraction of cases used for splitting each node: .0213
Maximum number of split levels: 19
Minimum node sample size: 46
Mean number of terminal nodes:    20.29

Classification matrix for training sample:
Predicted       True class
class              C         D         T
C                792       142         3
D                979      2696        81
T                  0         0         0
Total           1771      2838        84

Number of cases used for tree construction: 4693
Number misclassified: 1205
Resubstitution estimate of mean misclassification cost: .2568

Number of OOB cases: 4693
Number OOB misclassified: 1296
OOB estimate of mean misclassification cost: .2762
Mean number of trees per OOB observation: 36.86

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   0.25963E+00  0.72060E+00  0.19768E-01    "D"    "D"
y   0.26725E+00  0.71431E+00  0.18437E-01    "D"    "D"
y   0.20397E+00  0.78366E+00  0.12371E-01    "D"    "D"
y   0.23679E+00  0.75144E+00  0.11773E-01    "D"    "D"
y   0.21028E+00  0.76701E+00  0.22713E-01    "D"    "D"
y   0.24253E+00  0.72801E+00  0.29464E-01    "D"    "D"
y   0.50851E+00  0.47644E+00  0.15050E-01    "C"    "C"
y   0.46757E+00  0.49579E+00  0.36635E-01    "D"    "D"
y   0.26252E+00  0.71672E+00  0.20766E-01    "D"    "D"
y   0.22087E+00  0.76699E+00  0.12137E-01    "D"    "D"
y   0.47395E+00  0.50389E+00  0.22162E-01    "D"    "C"
y   0.27556E+00  0.70485E+00  0.19591E-01    "D"    "D"
y   0.45610E+00  0.53424E+00  0.96535E-02    "D"    "D"
y   0.39900E+00  0.57347E+00  0.27531E-01    "D"    "D"
```

```
y   0.25603E+00  0.72344E+00  0.20529E-01    "D"    "D"
y   0.27631E+00  0.68635E+00  0.37332E-01    "D"    "T"
y   0.81393E+00  0.18534E+00  0.73288E-03    "C"    "D"
```

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 $\texttt{wtgain}^2$ 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.

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.

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.

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.