User Manual for GUIDE ver. 41.1*

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

April 10, 2023

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, USDA Economic Research Service, and Eli Lilly & Co. Work on precursors to GUIDE additionally supported by IBM Research and Pfizer.

1 Warranty disclaimer

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. Provision for multiple missing-value codes and missing-value flag variables.
5. Optional automatic creation of missing-value indicator variables for regression.
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).

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

The GUIDE algorithm is documented in Loh (2002) for regression trees and Loh (2009) for classification trees. Reviews of the subject may be found in Loh (2008a, 2011, 2014). Advanced features of the algorithm are reported in Chaudhuri and Loh (2002), Loh (2006b, 2008b), Kim et al. (2007), Loh et al. (2007, 2019b, 2016, 2015, 2019c), and Loh and Zhou (2021). For third-party applications of GUIDE and predecessors, see `http://www.stat.wisc.edu/~loh/apps.html`. This manual demonstrates use of the GUIDE software and interpretation of the results.

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** or **gfortran**. Both versions are compiled in Ubuntu 20.0. Unzip the file with “`gunzip guide.gz`” and, if necessary, make it executable by typing “`chmod a+x guide`” in a `Terminal` window. To execute, type “`./guide`”.

macOS: There are five versions to choose from. The **NAG** versions do not require additional software to be installed; the **gfortran** versions require **Xcode** as explained below. Download the desired `guide.gz` file and double-click it to gunzip. The make it executable by typing the command “`chmod a+x guide`” in a `Terminal` application in the folder where the file is located. If this still does not allow you to run the app, carry out these steps:

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

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

NAG Fortran for Apple M1. This is the only native version for M1 Macs. It is compiled with macOS Monterey 12.4.

NAG Fortran for Intel. This is version is for Intel Macs, but it works for M1 Macs too. It is compiled with macOS Catalina 10.15.7 but also works on Big Sur.

`https://github.com/fxcoudert/gfortran-for-macOS/releases`

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 if no missing values	Yes	No	No	Yes if no missing values
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
Missing values in splits	Missing as observed	Node mean or mode imputation	Surrogate splits	Surrogate splits	Weights	Random splits[d]
Missing-value flag variables	Yes	No	No	No	No	No
Pruning	Yes	Yes	Yes	Yes	No	No
Tree diagrams	Text and LaTeX			R	Text	R
Bagging	Yes	No	No	No	No	No
Forests	Yes	No	No	No	No	cforest
Importance scores	Yes	No	No	Yes	No	Yes

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

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

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

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

Monterey 12.6.2 with gfortran 12.1. This version is compiled with **Xcode 14.2**.

1. Install **Xcode** from `https://developer.apple.com/xcode/downloads/`.
2. Download the `gfortran` disk image from
   `https://github.com/fxcoudert/gfortran-for-macOS/releases/download/12.1-monterey/gfortran-Intel-12.1-Monte`
3. Double-click the disk image to install `gfortran`.

Big Sur 11.7.1 with gfortran 11.2. This version is compiled with **Xcode 13.2.1**.

1. Install **Xcode** from `https://developer.apple.com/xcode/downloads/`.
2. Download the `gfortran` disk image from
   `https://github.com/fxcoudert/gfortran-for-macOS/releases/tag/11.2-bigsur-intel`
3. Double-click the disk image to install `gfortran`.

gfortran for High Sierra. This version is for Mac that cannot be upgrade above macOS High Sierra. It is compiled with **Xcode 10.1** and **gfortran 5.1**. 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 two 64-bit executables to choose from: **gfortran** or **Intel**. Both versions are compiled under Windows 10. The **gfortran** version is preferable because the `Intel` version sometimes crashes due to a compiler bug. Download the file `guide.zip` and unzip it (right-click on file icon and select "Extract all"). The resulting file `guide.exe` may be placed in one of three places:

1. Top level of your `C` drive. Type "`C:\guide`" in a **Command Prompt** window to execute—see Section 3.1.
2. A folder that contains your data files. Type "`guide`" in that folder to execute.
3. A folder on your search path. Type "`guide`" anywhere to execute.

2.2 LATEX

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

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

Mac: MacTeX http://tug.org/mactex/ or
MikTeX https://miktex.org/howto/install-miktex-mac. Both include the **TeXShop** GUI frontend.

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

The LATEX files produced by GUIDE can be edited to change colors, node sizes, etc., in the trees; see *pstricks manual* (http://tug.org/PSTricks/main.cgi/). There are two ways to generate pdf figures of the tree diagrams. In the following, assume that the LATEX file is named `diagram.tex`.

1. **Terminal window (simplest).** Type these three commands in the **Terminal** (Linux or Mac) or **Command Prompt** (Win) window where the LATEX file (say, `diagram.tex`) was produced.

   (a) `latex diagram`

   (b) `dvips diagram`

   (c) `ps2pdf diagram.ps`

   The first command produces a file called `diagram.dvi`. The second command converts the latter to postscript file called `diagram.ps` (which can be edited with any postscript app). The third command turns it into a pdf file with name `diagram.pdf`.

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

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

3 Program operation

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

Linux. Any terminal program will do.

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

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

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

After the terminal window is opened, change to the folder where the data and program files are stored. Mac and Windows users are unfamiliar with terminal commands may consult
`https://wiredpen.com/resources/basic-unix-commands-for-osx/`
and `https://cmdref.net/os/windows/command/index.html`, respectively.

3.1 Required files

GUIDE requires two text files to begin.

Data file: This file contains the data from the training sample. Each data record consists of observations on the dependent variable, the predictor (i.e., X or independent) variables, and optional weight, missing value flag, time, offset, periodic, and event indicator (for censored responses) variables. Entries in each record are comma, space, or tab delimited (multiple spaces are treated as one space, but not for commas). A record can occupy more than one line in the file, but each record must begin on a new line.

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

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

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

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

```
rhcdata.txt
NA
2
1 X x
2 cat1 c
3 cat2 c
4 ca c
5 sadmdte x
6 dschdte x
7 dthdte x
8 lstctdte x
9 death x
10 cardiohx c
11 chfhx c
12 dementhx c
13 psychhx c
14 chrpulhx c
15 renalhx c
16 liverhx c
17 gibledhx c
18 malighx c
19 immunhx c
20 transhx c
21 amihx c
22 age n
```

```
23 sex c
24 edu n
25 surv2md1 n
26 das2d3pc n
27 t3d30 x
28 dth30 x
29 aps1 n
30 scoma1 n
31 meanbp1 n
32 wblc1 n
33 hrt1 n
34 resp1 n
35 temp1 n
36 pafi1 n
37 alb1 n
38 hema1 n
39 bili1 n
40 crea1 n
41 sod1 n
42 pot1 n
43 paco21 n
44 ph1 n
45 swang1 d
46 wtkilo1 n
47 dnr1 c
48 ninsclas c
49 resp c
50 card c
51 neuro c
52 gastr c
53 renal c
54 meta c
55 hema c
56 seps c
57 trauma c
58 ortho c
59 adld3p n
60 urin1 n
```

```
61 race c
62 income c
63 ptid x
64 survtime x
```

The 1st line gives the name of the data file. If the file is not in the current folder, its full path must be given (e.g., `"c:\data\rhcdata.txt"` for Windows users or `"~/Data/rhcdata.txt"` for Mac users) 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 text output (but not in R 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 **E**stimated 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 a **c**, **n** or **s** variable in the description file. Missing value flag variables associated with any other variable type (including **b** and **p**) should be specified as **c**.

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. 8 for an example.

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!).

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

```
> guide
GUIDE Classification and Regression Trees and Forests
Version 41.0 (Build date: March 3, 2023)
Compiled with GFortran 12.1 on macOS Monterey 12.6.3
Copyright (c) 1997-2023 Wei-Yin Loh. All rights reserved.
Software based upon work partially supported by the U.S. Army Research Office,
National Science Foundation, National Institutes of Health,
Bureau of Labor Statistics, USDA Economic Research Service, and Eli Lilly.

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

4 Classification: RHC data

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

To construct a classification tree for predicting `swang1`, we need to generate an input file from the description file `rhcdsc1.txt`, which specifies `swang1` as a `d` variable and `dth30` and `death` both as `x`. When GUIDE prompts for a selection, there is usually range of permissible values given within square brackets and a default choice (indicated by the symbol `<cr>=`). The default may be selected by pressing the `ENTER` or `RETURN` key.

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

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

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

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

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

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

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
```

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

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

```
23 N variables changed to S
D variable is swang1
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 2
Assigning integer codes to values of 30 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
Creating missing value indicators ...
Rereading data ...
Class  #Cases    Proportion
NoRHC    3551    0.61918047
RHC      2184    0.38081953
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157       10        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       30        0
Number of cases used for training: 5735
Number of split variables: 53
Number of cases excluded due to 0 W or missing D variable: 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
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: classfit.txt
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: classpred.r
Input rank of top variable to split root node ([1:53], <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)
  41.0      (version of GUIDE that generated this file)
 1          (1=model fitting, 2=importance or DIF scoring, 3=data conversion)
"classout.txt"  (name of output file)
 1          (1=one tree, 2=ensemble)
 1          (1=classification, 2=regression, 3=propensity score grouping)
 1          (1=simple model, 2=nearest-neighbor, 3=kernel)
 1          (0=linear 1st, 1=univariate 1st, 2=skip linear, 3=skip linear and interaction)
 1          (0=tree with fixed no. of nodes, 1=prune by CV, 2=by test sample, 3=no pruning)
"rhcdsc1.txt"  (name of data description file)
        10  (number of cross-validations)
 1          (1=mean-based CV tree, 2=median-based CV tree)
     0.250  (SE number for pruning)
 1          (1=estimated priors, 2=equal priors, 3=other priors)
 1          (1=unit misclassification costs, 2=other)
 2          (1=split point from quantiles, 2=use exhaustive search)
 1          (1=default max. number of split levels, 2=specify no. in next line)
 1          (1=default min. node size, 2=specify min. value in next line)
 2          (0=no LaTeX code, 1=tree without node numbers, 2=tree with node numbers)
"class.tex" (latex file name)
 1          (1=color terminal nodes, 2=no colors)
 2          (0=#errors, 1=sample sizes, 2=sample proportions, 3=posterior probs, 4=nothing)
 1          (1=no storage, 2=store fit and split variables, 3=store split variables and values)
 2          (1=do not save fitted values and node IDs, 2=save in a file)
"classfit.txt" (file name for fitted values and node IDs)
 2          (1=do not write R function, 2=write R function)
"classpred.r" (R code file)
 1          (rank of top variable to split root node)
```

4.1.3 Contents of `classout.txt`

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

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

Summary information for training sample of size 5735
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                      #Codes/
                                                      Levels/
 Column  Name             Minimum      Maximum    Periods   #Missing
      2  cat1       c                                   9
      3  cat2       c                                   6     4535
      4  ca         c                                   3
     10  cardiohx   c                                   2
     11  chfhx      c                                   2
     12  dementhx   c                                   2
     13  psychhx    c                                   2
     14  chrpulhx   c                                   2
     15  renalhx    c                                   2
     16  liverhx    c                                   2
     17  gibledhx   c                                   2
     18  malighx    c                                   2
     19  immunhx    c                                   2
```

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

```
         0         0         0        30         0
Number of cases used for training: 5735
Number of split variables: 53
Number of cases excluded due to 0 W or missing D variable: 0

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

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

Size and CV mean cost and SE of subtrees:
 Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   1       68   3.236E-01   6.178E-03   3.960E-03   3.284E-01   6.780E-03
   2       67   3.236E-01   6.178E-03   3.960E-03   3.284E-01   6.780E-03
   3       66   3.236E-01   6.178E-03   3.960E-03   3.284E-01   6.780E-03
   4       65   3.236E-01   6.178E-03   3.960E-03   3.284E-01   6.780E-03
   :
  37       18   3.180E-01   6.150E-03   2.945E-03   3.217E-01   3.907E-03
  38+      12   3.198E-01   6.159E-03   3.064E-03   3.182E-01   3.105E-03
  39**     10   3.180E-01   6.150E-03   2.127E-03   3.188E-01   3.098E-03
  40        8   3.219E-01   6.169E-03   3.105E-03   3.217E-01   5.293E-03
  41        6   3.240E-01   6.180E-03   3.474E-03   3.249E-01   6.673E-03
  42        5   3.228E-01   6.174E-03   3.471E-03   3.249E-01   5.539E-03
  43        3   3.325E-01   6.221E-03   3.956E-03   3.365E-01   6.220E-03
  44        2   3.751E-01   6.393E-03   4.248E-03   3.801E-01   3.186E-03
  45        1   3.808E-01   6.412E-03   2.782E-04   3.805E-01   4.832E-04
Above shows that the largest tree has 68 terminal nodes.
0-SE tree based on mean is marked with * and has 10 terminal nodes
0-SE tree based on median is marked with + and has 12 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
```

```
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as ++ tree
** tree same as -- tree
++ tree same as -- tree
* tree same as ** tree
* tree same as ++ tree
* tree same as -- tree
Pruned tree has 10 terminal nodes and is marked by two asterisks.
Following tree is based on mean CV with naive SE estimate (**)

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

Node cost is node misclassification cost divided by number of training cases
       Node     Total     Train     Predicted        Node     Split          Interacting
      label     cases     cases     class            cost     variables      variable
          1      5735      5735     NoRHC          3.808E-01 cat1
          2      1683      1683     RHC            4.599E-01 meanbp1
          4      1117      1117     RHC            3.796E-01 pafi1
         8T       655       655     RHC            3.038E-01 resp1
          9       462       462     RHC            4.870E-01 ninsclas
        18T       244       244     RHC            3.730E-01 bili1
        19T       218       218     NoRHC          3.853E-01 card
         5T       566       566     NoRHC          3.816E-01 alb1
          3      4052      4052     NoRHC          3.147E-01 pafi1
          6      1292      1292     NoRHC          4.837E-01 resp
         12       581       581     RHC            4.200E-01 dnr1
         24       515       515     RHC            3.903E-01 cat1
        48T       438       438     RHC            3.447E-01 meanbp1
        49T        77        77     NoRHC          3.506E-01 -
        25T        66        66     NoRHC          3.485E-01 -
         13       711       711     NoRHC          4.051E-01 seps
        26T       110       110     RHC            3.636E-01 -
        27T       601       601     NoRHC          3.627E-01 adld3p
         7T      2760      2760     NoRHC          2.355E-01 aps1
Above gives the number of observations in each node (terminal
 node marked with a T), its predicted class, and the split variable.

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

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

```
 Node 1: cat1 = "CHF", "MOSF w/Sepsis"
   Node 2: meanbp1 <= 68.500000 or NA
     Node 4: pafi1 <= 266.15625
       Node 8: RHC
     Node 4: pafi1 > 266.15625 or NA
       Node 9: ninsclas = "No insurance", "Private", "Private & Medicare"
         Node 18: RHC
       Node 9: ninsclas /= "No insurance", "Private", "Private & Medicare"
         Node 19: NoRHC
   Node 2: meanbp1 > 68.500000
     Node 5: NoRHC
 Node 1: cat1 /= "CHF", "MOSF w/Sepsis"
   Node 3: pafi1 <= 142.35938
     Node 6: resp = "No"
       Node 12: dnr1 = "No"
         Node 24: cat1 = "ARF", "Lung Cancer", "MOSF w/Malignancy"
           Node 48: RHC
         Node 24: cat1 /= "ARF", "Lung Cancer", "MOSF w/Malignancy"
           Node 49: NoRHC
       Node 12: dnr1 /= "No"
         Node 25: NoRHC
     Node 6: resp /= "No"
       Node 13: seps = "Yes"
         Node 26: RHC
       Node 13: seps /= "Yes"
         Node 27: NoRHC
   Node 3: pafi1 > 142.35938 or NA
     Node 7: NoRHC

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

Node 1: Intermediate node
 A case goes into Node 2 if cat1 = "CHF", "MOSF w/Sepsis"
 cat1 mode = "ARF"
 Class      Number   Posterior
 NoRHC        3551  0.6192E+00
 RHC          2184  0.3808E+00
 Number of training cases misclassified = 2184
 Predicted class is NoRHC
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if meanbp1 <= 68.500000 or NA
 meanbp1 mean = 72.674985
 Class      Number   Posterior
 NoRHC         774  0.4599E+00
```

```
 RHC           909  0.5401E+00
 Number of training cases misclassified = 774
 Predicted class is RHC
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if pafi1 <= 266.15625
 pafi1 mean = 241.37331
 Class      Number   Posterior
 NoRHC         424  0.3796E+00
 RHC           693  0.6204E+00
 Number of training cases misclassified = 424
 Predicted class is RHC
 ----------------------------
Node 8: Terminal node
 Class      Number   Posterior
 NoRHC         199  0.3038E+00
 RHC           456  0.6962E+00
 Number of training cases misclassified = 199
 Predicted class is RHC
 ----------------------------
Node 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
Elapsed time in seconds: 14.489
```

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

4.1.4 Contents of `classfit.txt`

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

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

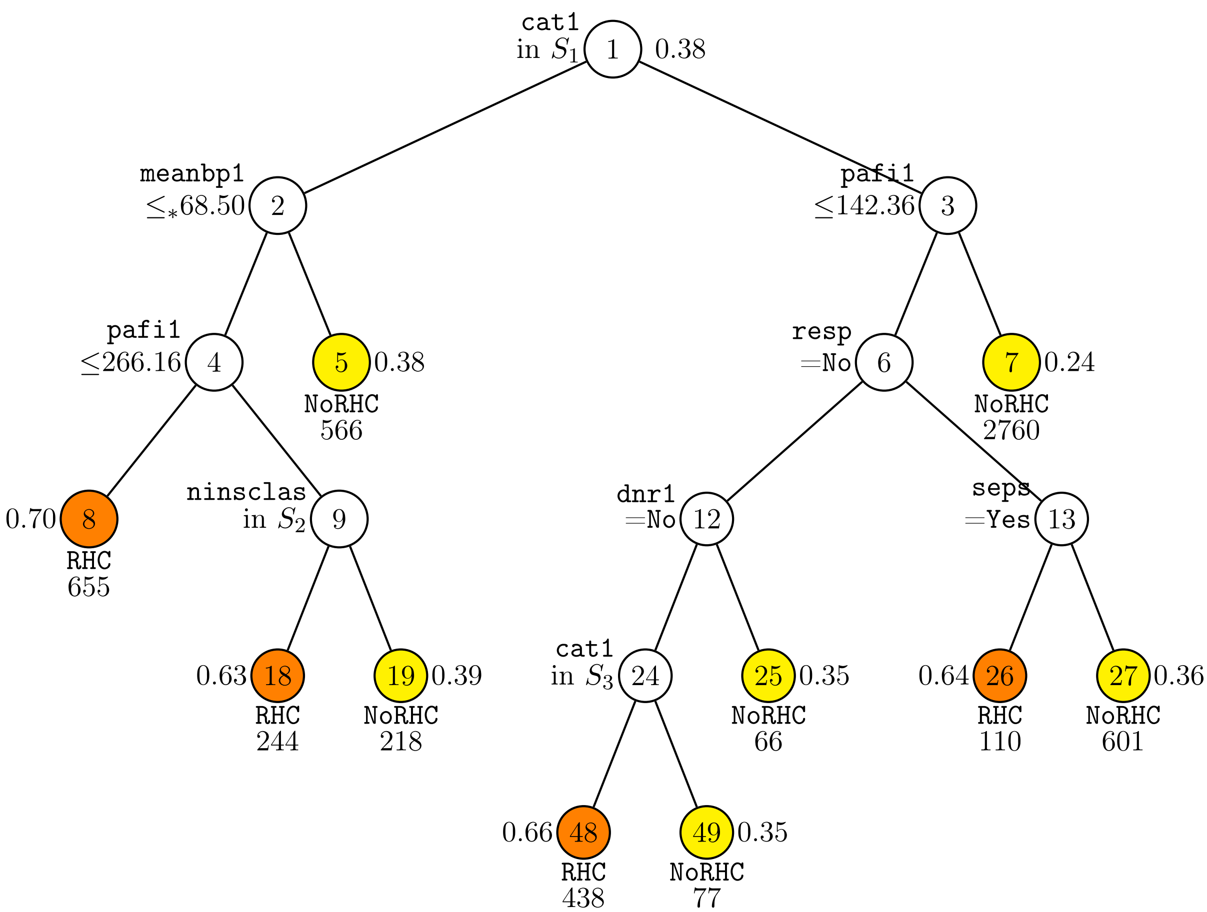

Figure 1: GUIDE v.41.0 0.250-SE classification tree for predicting `swang1` using estimated priors and unit misclassification costs. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. S_1 = {`CHF`, `MOSF w/Sepsis`}. S_2 = {`No insurance`, `Private`, `Private & Medicare`}. S_3 = {`ARF`, `Lung Cancer`, `MOSF w/Malignancy`}. Predicted classes and sample sizes (in *italics*) printed below terminal nodes; class sample proportion for `swang1` = `RHC` beside nodes. Second best split variable at root node is `aps1`.

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
```

```
##
##
## If desired, replace "rhcdata.txt" with name of file containing new data
## New file must have at least the same variables with same names
## (but not necessarily the same order) as in the training data file
## Missing value code is converted to NA if not already NA
newdata <- read.table("rhcdata.txt",header=TRUE,colClasses="character")
## node contains terminal node ID of each case
## pred.class contains predicted class
## prob contains predicted posterior probabilities
node <- NULL
pred.class <- NULL
prob <- 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])
    prob <- rbind(prob,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): 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
23 N variables changed to S
D variable is swang1
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 2
Assigning integer codes to values of 30 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
Creating missing value indicators ...
Rereading data ...
Class  #Cases    Proportion
NoRHC    3551    0.61918047
RHC      2184    0.38081953
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157       10        0        0       23
```

```
     #P-var   #M-var   #B-var   #C-var   #I-var
          0        0        0       30        0
 Number of cases used for training: 5735
 Number of split variables: 53
 Number of cases excluded due to 0 W or missing D variable: 0
 Finished reading data file
 Default number of cross-validations:           10
 Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
 Best tree may be chosen based on mean or median CV estimate
 Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
 Input number of SEs for pruning ([0.00:1000.00], <cr>=0.25):
 Choose 1 for estimated priors, 2 for equal priors, 3 to input priors from a file
 Input 1, 2, or 3 ([1:3], <cr>=1):
 Choose 1 for unit misclassification costs, 2 to input costs from a file
 Input 1 or 2 ([1:2], <cr>=1):
 Choose a split point selection method for numerical variables:
 Choose 1 to use faster method based on sample quantiles
 Choose 2 to use exhaustive search
 Input 1 or 2 ([1:2], <cr>=2):
 Default max. number of split levels: 15
 Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
 Default minimum node sample size is 57
 Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
 Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
 Input file name to store LaTeX code (use .tex as suffix): linear.tex
 Input 1 to color terminal nodes, 2 otherwise ([1:2], <cr>=1):
 Choose amount of detail in nodes of LaTeX tree diagram:
 Input 0 for #errors, 1 for sample sizes, 2 for sample proportions, 3 for posterior probs,
 4 for nothing
 Input your choice ([0:4], <cr>=2):
 You can store the variables and/or values used to split and fit in a file
 Choose 1 to skip this step, 2 to store split variables and their values
 Input your choice ([1:2], <cr>=1):
 Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
 Input name of file to store node ID and fitted value of each case: linearfit.txt
 Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
 Input file name: linearpred.r
 Input rank of top variable to split root node ([1:53], <cr>=1):
 Input file is created!
 Run GUIDE with the command: guide < linearin.txt
  Press ENTER or RETURN to quit
```

4.2.2 Contents of `linearin.txt`

```
GUIDE       (do not edit this file unless you know what you are doing)
```

```
  41.0      (version of GUIDE that generated this file)
 1          (1=model fitting, 2=importance or DIF scoring, 3=data conversion)
"linearout.txt"  (name of output file)
 1          (1=one tree, 2=ensemble)
 1          (1=classification, 2=regression, 3=propensity score grouping)
 1          (1=simple model, 2=nearest-neighbor, 3=kernel)
 0          (0=linear 1st, 1=univariate 1st, 2=skip linear, 3=skip linear and interaction)
 1          (0=tree with fixed no. of nodes, 1=prune by CV, 2=by test sample, 3=no pruning)
"rhcdsc1.txt"  (name of data description file)
        10  (number of cross-validations)
 1          (1=mean-based CV tree, 2=median-based CV tree)
     0.250  (SE number for pruning)
 1          (1=estimated priors, 2=equal priors, 3=other priors)
 1          (1=unit misclassification costs, 2=other)
 2          (1=split point from quantiles, 2=use exhaustive search)
 1          (1=default max. number of split levels, 2=specify no. in next line)
 1          (1=default min. node size, 2=specify min. value in next line)
 2          (0=no LaTeX code, 1=tree without node numbers, 2=tree with node numbers)
"linear.tex" (latex file name)
 1          (1=color terminal nodes, 2=no colors)
 2          (0=#errors, 1=sample sizes, 2=sample proportions, 3=posterior probs, 4=nothing)
 1          (1=no storage, 2=store split variables and values)
 2          (1=do not save fitted values and node IDs, 2=save in a file)
"linearfit.txt" (file name for fitted values and node IDs)
 2          (1=do not write R function, 2=write R function)
"linearpred.r" (R code file)
 1          (rank of top variable to split root node)
```

4.2.3 Contents of linearout.txt

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

```
Class  #Cases     Proportion
NoRHC    3551     0.61918047
RHC      2184     0.38081953

Summary information for training sample of size 5735
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                 #Codes/
                                                 Levels/
 Column  Name           Minimum      Maximum    Periods   #Missing
      2  cat1       c                                  9
      3  cat2       c                                  6     4535
      4  ca         c                                  3
     10  cardiohx   c                                  2
     11  chfhx      c                                  2
     12  dementhx   c                                  2
     13  psychhx    c                                  2
     14  chrpulhx   c                                  2
     15  renalhx    c                                  2
     16  liverhx    c                                  2
     17  gibledhx   c                                  2
     18  malighx    c                                  2
     19  immunhx    c                                  2
     20  transhx    c                                  2
     21  amihx      c                                  2
     22  age        s    18.04        101.8
     23  sex        c                                  2
     24  edu        s    0.000        30.00
     25  surv2md1   s    0.000       0.9620
     26  das2d3pc   s    11.00        33.00
     29  aps1       s    3.000        147.0
     30  scoma1     s    0.000        100.0
     31  meanbp1    s    10.00        259.0                    80
     32  wblc1      s    0.000        192.0
     33  hrt1       s    8.000        250.0                   159
     34  resp1      s    2.000        100.0                   136
     35  temp1      s    27.00        43.00
     36  pafi1      s    11.60        937.5
     37  alb1       s   0.3000        29.00
     38  hema1      s    2.000        66.19
     39  bili1      s   0.9999E-01    58.20
     40  crea1      s   0.9999E-01    25.10
     41  sod1       s    101.0        178.0
     42  pot1       s    1.100        11.90
```

```
      43  paco21     s    1.000       156.0
      44  ph1        s    6.579       7.770
      45  swang1     d                                  2
      46  wtkilo1    s    19.50       244.0                    515
      47  dnr1       c                                  2
      48  ninsclas   c                                  6
      49  resp       c                                  2
      50  card       c                                  2
      51  neuro      c                                  2
      52  gastr      c                                  2
      53  renal      c                                  2
      54  meta       c                                  2
      55  hema       c                                  2
      56  seps       c                                  2
      57  trauma     c                                  2
      58  ortho      c                                  2
      59  adld3p     s    0.000       7.000                   4296
      60  urin1      s    0.000       9000.                   3028
      61  race       c                                  3
      62  income     c                                  4

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

 Constant fitted to cases with missing values in regressor variables
 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: 0.2500

 Simple node models
 Estimated priors
 Unit misclassification costs
 Linear split highest priority
 Interaction and linear splits 2nd and 3rd priorities
 Split values for N and S variables based on exhaustive search
 Maximum number of split levels: 15
 Minimum node sample size: 57
 Top-ranked variables and 1-df 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
     :
    48  0.1861E+01   temp1
    49  0.1376E+01   renalhx
    50  0.1052E+01   meta
    51  0.6357E+00   race

 Size and CV mean cost and SE of subtrees:
  Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
    1       59   3.085E-01   6.099E-03   7.419E-03   3.139E-01   8.732E-03
    2       58   3.085E-01   6.099E-03   7.419E-03   3.139E-01   8.732E-03
    3       57   3.085E-01   6.099E-03   7.419E-03   3.139E-01   8.732E-03
    :
   29       17   3.060E-01   6.085E-03   7.366E-03   3.078E-01   8.293E-03
   30**     16   3.050E-01   6.079E-03   7.354E-03   3.025E-01   8.394E-03
   31       12   3.085E-01   6.099E-03   7.055E-03   3.072E-01   7.716E-03
   32        9   3.083E-01   6.098E-03   6.862E-03   3.069E-01   7.082E-03
   33        6   3.158E-01   6.138E-03   6.474E-03   3.191E-01   1.028E-02
   34        3   3.425E-01   6.266E-03   7.205E-03   3.479E-01   1.195E-02
   35        1   3.808E-01   6.412E-03   2.782E-04   3.805E-01   4.832E-04

 0-SE tree based on mean is marked with * and has 16 terminal nodes
 0-SE tree based on median is marked with + and has 16 terminal nodes
 Selected-SE tree based on mean using naive SE is marked with **
 Selected-SE tree based on mean using bootstrap SE is marked with --
 Selected-SE tree based on median and bootstrap SE is marked with ++
 * tree, ** tree, + tree, and ++ tree all the same

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

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

 Node cost is node misclassification cost divided by number of training cases
        Node    Total    Train    Predicted       Node    Split          Interacting
       label    cases    cases    class           cost    variables      variable
           1     5735     5735    NoRHC        3.808E-01 cat1
           2     1683     1683    RHC          4.599E-01 meanbp1 +pafi1
           4     1174     1174    RHC          3.705E-01 resp1 +surv2md1
          8T      229      229    RHC          1.790E-01 sod1 :wtkilo1
           9      945      945    RHC          4.169E-01 ninsclas
         18T      321      321    RHC          3.084E-01 -
          19      624      624    RHC          4.728E-01 dnr1
          38      554      554    RHC          4.495E-01 adld3p +edu
         76T      479      479    RHC          4.071E-01 -
         77T       75       75    NoRHC        2.800E-01 -
```

```
         39T      70       70    NoRHC          3.429E-01 -
          5T     509      509    NoRHC          3.340E-01 resp1 +adld3p
          3     4052     4052    NoRHC          3.147E-01 pafi1 +adld3p
          6     3330     3330    NoRHC          3.526E-01 aps1 +hema1
         12T    1092     1092    NoRHC          1.795E-01 pafi1 +scoma1
         13     2238     2238    NoRHC          4.370E-01 pafi1 +resp1
         26T     390      390    RHC            3.000E-01 cat2
         27     1848     1848    NoRHC          3.815E-01 aps1 +adld3p
         54T      74       74    NoRHC          2.432E-01 -
         55     1774     1774    NoRHC          3.873E-01 aps1 +wtkilo1
        110T     607      607    NoRHC          2.636E-01 card
        111     1167     1167    NoRHC          4.516E-01 meanbp1 +pafi1
        222      602      602    RHC            4.485E-01 paco21 +wtkilo1
        444T      94       94    RHC            2.340E-01 -
        445      508      508    RHC            4.882E-01 scoma1
        890      260      260    RHC            4.269E-01 bili1 +pot1
       1780T     155      155    RHC            3.226E-01 resp
       1781T     105      105    NoRHC          4.190E-01 -
        891T     248      248    NoRHC          4.476E-01 sex
        223T     565      565    NoRHC          3.451E-01 crea1 +pafi1
          7T     722      722    NoRHC          1.399E-01 card

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

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

Node 1: cat1 = "CHF", "MOSF w/Sepsis"
  Node 2: 0.24316737 * pafi1 + meanbp1 <= 153.28329 or NA
    Node 4: 48.127695 * surv2md1 + resp1 <= 43.437797 or NA
      Node 8: RHC
    Node 4: 48.127695 * surv2md1 + resp1 > 43.437797
      Node 9: ninsclas = "No insurance", "Private"
        Node 18: RHC
      Node 9: ninsclas /= "No insurance", "Private"
        Node 19: dnr1 = "No"
          Node 38: -23.826398 * edu + adld3p <= -282.91678 or NA
            Node 76: RHC
          Node 38: -23.826398 * edu + adld3p > -282.91678
            Node 77: NoRHC
        Node 19: dnr1 /= "No"
          Node 39: NoRHC
  Node 2: 0.24316737 * pafi1 + meanbp1 > 153.28329
    Node 5: NoRHC
```

```
 Node 1: cat1 /= "CHF", "MOSF w/Sepsis"
   Node 3: 11.508773 * adld3p + pafi1 <= 149.35252 or NA
     Node 6: -1.3120163 * hema1 + aps1 <= 0.84337055
       Node 12: NoRHC
     Node 6: -1.3120163 * hema1 + aps1 > 0.84337055 or NA
       Node 13: 4.0975611 * resp1 + pafi1 <= 207.99333
         Node 26: RHC
       Node 13: 4.0975611 * resp1 + pafi1 > 207.99333 or NA
         Node 27: -23.161068 * adld3p + aps1 <= 66.838932
           Node 54: NoRHC
         Node 27: -23.161068 * adld3p + aps1 > 66.838932 or NA
           Node 55: 1.0116045 * wtkilo1 + aps1 <= 121.69374 or NA
             Node 110: NoRHC
           Node 55: 1.0116045 * wtkilo1 + aps1 > 121.69374
             Node 111: 0.35358803 * pafi1 + meanbp1 <= 134.65949 or NA
               Node 222: -0.42185873 * wtkilo1 + paco21 <= -7.0243280
                 Node 444: RHC
               Node 222: -0.42185873 * wtkilo1 + paco21 > -7.0243280 or NA
                 Node 445: scoma1 <= 4.5000000
                   Node 890: 5.8542561 * pot1 + bili1 <= 25.404949
                     Node 1780: RHC
                   Node 890: 5.8542561 * pot1 + bili1 > 25.404949 or NA
                     Node 1781: NoRHC
                 Node 445: scoma1 > 4.5000000 or NA
                   Node 891: NoRHC
             Node 111: 0.35358803 * pafi1 + meanbp1 > 134.65949
               Node 223: NoRHC
   Node 3: 11.508773 * adld3p + pafi1 > 149.35252
     Node 7: NoRHC

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

Node 1: Intermediate node
 A case goes into Node 2 if cat1 = "CHF", "MOSF w/Sepsis"
 cat1 mode = "ARF"
 Class      Number   Posterior
 NoRHC        3551  0.6192E+00
 RHC          2184  0.3808E+00
 Number of training cases misclassified = 2184
 Predicted class is NoRHC
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if 0.24316737 * pafi1 + meanbp1 <= 153.28329
 Linear combination mean = 133.36641
 Class      Number   Posterior
```

```
 NoRHC         774  0.4599E+00
 RHC           909  0.5401E+00
 Number of training cases misclassified = 774
 Predicted class is RHC
 ----------------------------
:
:
Node 891: Terminal node
 Class      Number   Posterior
 NoRHC         137  0.5524E+00
 RHC           111  0.4476E+00
 Number of training cases misclassified = 111
 Predicted class is NoRHC
 ----------------------------
Node 223: Terminal node
 Class      Number   Posterior
 NoRHC         370  0.6549E+00
 RHC           195  0.3451E+00
 Number of training cases misclassified = 195
 Predicted class is NoRHC
 ----------------------------
Node 7: Terminal node
 Class      Number   Posterior
 NoRHC         621  0.8601E+00
 RHC           101  0.1399E+00
 Number of training cases misclassified = 101
 Predicted class is NoRHC
 ----------------------------

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

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

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

The LaTeX tree is shown in Figure 2, where each node that is split on a pair of

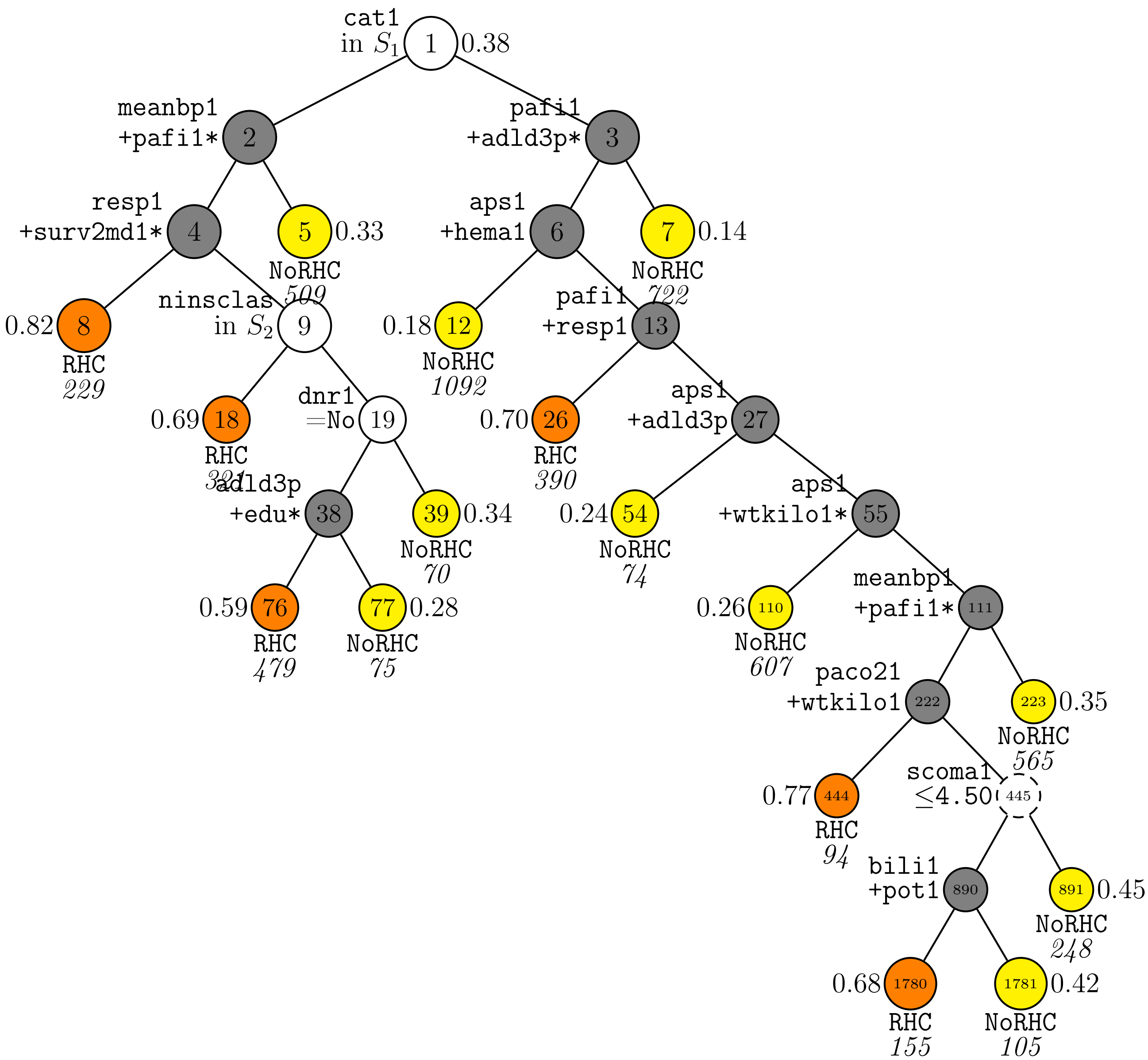

Figure 2: GUIDE v.41.0 0.250-SE classification tree for predicting `swang1` using linear split priority, estimated priors and unit misclassification costs. At each split, an observation goes to the left branch if and only if the condition is satisfied. An asterisk at a bivariate split indicates that missing values in either variable go to the left node. S_1 = {`CHF`, `MOSF w/Sepsis`}. S_2 = {`No insurance`, `Private`}. Circles with dashed lines are nodes with no significant split variables. Intermediate nodes in gray indicate linear splits. Predicted classes and sample sizes (in *italics*) printed below terminal nodes; class sample proportion for `swang1` = `RHC` beside nodes. Second best split variable at root node is `aps1`.

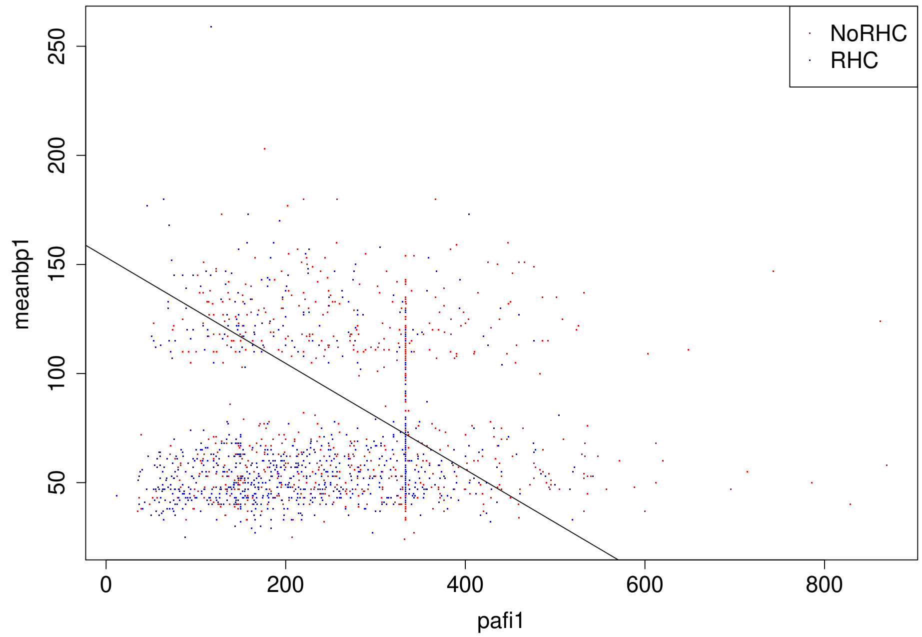

Figure 3: Plot of `meanbp1` vs `pafi1` for data and split in node 2 of tree in Figure 2

ordinal variables is painted 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. The R code for making the plot is below. It reads `linearfit.txt` to extract the observations in the node.

4.2.4 R code for plot

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

```
leg.pch <- c(1,4)
plot(x,y,xlab="pafi1",ylab="meanbp1",type="n")
g1 <- z0$swang1[gp] == "NoRHC"
points(x[g1],y[g1],pch=leg.pch[1],col=leg.col[1])
points(x[!g1],y[!g1],pch=leg.pch[2],col=leg.col[2])
abline(c(161.61473,-0.26651164))
legend("topright",legend=leg.txt,col=leg.col,pch=leg.pch,cex=1.5)
```

4.3 Kernel discriminant models

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

4.3.1 Input file generation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
1
 Name of batch input file: ker2.in
ker2.in
 File ker2.in exists
 Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):

 Input 1 for model fitting, 2 for importance or DIF scoring, 3 for data conversion ([1:3], <cr>=1):

 Name of batch output file: ker2.out
ker2.out
 File ker2.out exists
 Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):

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


 Input name of data description file (max 100 characters);
 enclose with matching quotes if it has spaces: rhcdsc1.txt
rhcdsc1.txt
 Reading data description file ...
 Training sample file: rhcdata.txt
 Missing value code: NA
 Records in data file start on line 2
 23 N variables changed to S
 D variable is swang1
 Reading data file ...
 Number of records in data file: 5735
 Length of longest entry in data file: 19
 Checking for missing values ...
 Finished checking
 Missing values found among categorical variables
 Separate categories will be created for missing categorical variables
 Missing values found among non-categorical variables
 Recoding D values to integers
 Finished recoding
 Number of classes: 2
 Assigning integer codes to values of 30 categorical variables
 Finished assigning codes to 10 categorical variables
 Finished assigning codes to 20 categorical variables
 Finished assigning codes to 30 categorical variables
 Re-checking data ...
 Allocating missing value information ...
 Assigning codes to missing values, if any ...
 Finished processing 5000 of 5735 observations
 Data checks complete
 Creating missing value indicators ...
 Rereading data ...
 Class  #Cases     Proportion
 NoRHC    3551     0.61918047
 RHC      2184     0.38081953
      Total  #cases w/   #missing
     #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       5735          0       5157       10        0        0       23
     #P-var   #M-var   #B-var   #C-var   #I-var
          0        0        0       30        0
 Number of cases used for training: 5735
 Number of split variables: 53
 Number of cases excluded due to 0 W or missing D variable: 0
 Finished reading data file
```

```
 Default number of cross-validations:           10
 Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):

 Best tree may be chosen based on mean or median CV estimate
 Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):

 Input number of SEs for pruning ([0.00:1000.00], <cr>=0.25):

 Choose 1 for estimated priors, 2 for equal priors, 3 to input priors from a file
 Input 1, 2, or 3 ([1:3], <cr>=1):

 Choose 1 for unit misclassification costs, 2 to input costs from a file
 Input 1 or 2 ([1:2], <cr>=1):

 Choose a split point selection method for numerical variables:
 Choose 1 to use faster method based on sample quantiles
 Choose 2 to use exhaustive search
 Input 1 or 2 ([1:2], <cr>=2):

 Default max. number of split levels: 15
 Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):

 Default minimum node sample size is 57
 Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):

 Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):

 Input file name to store LaTeX code (use .tex as suffix): ker2.tex
ker2.tex
 A file by that name exists
 Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):

 Input 1 to color terminal nodes, 2 otherwise ([1:2], <cr>=1):

 Choose amount of detail in nodes of LaTeX tree diagram:
 Input 0 for #errors, 1 for sample sizes, 2 for sample proportions, 3 for posterior probs, 4 for not
 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 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
```

```
ker2.fit
 A file by that name exists
 Input 1 to overwrite it, 2 to choose another name ([1:2], <cr>=1):

 Input rank of top variable to split root node ([1:53], <cr>=1):

 Input file is created!
 Run GUIDE with the command: guide < ker2.in
```

4.3.2 Contents of ker2.out

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

 Summary information for training sample of size 5735
 d=dependent, b=split and fit cat variable using indicator variables,
 c=split-only categorical, i=fit-only categorical (via indicators),
 s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
 m=missing-value flag variable, p=periodic variable, w=weight
                                                 #Codes/
                                                 Levels/
  Column  Name            Minimum      Maximum    Periods   #Missing
       2  cat1       c                                  9
       3  cat2       c                                  6     4535
       4  ca         c                                  3
      10  cardiohx   c                                  2
       :
      58  ortho      c                                  2
      59  adld3p     s    0.000        7.000                  4296
```

```
     60  urin1      s    0.000       9000.                  3028
     61  race       c                                 3
     62  income     c                                 4

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

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Kernel density node models
Bivariate preference
Estimated priors
Unit misclassification costs
Bivariate split highest priority
Interaction splits 2nd priority; no linear splits
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 15
Minimum node sample size: 57
Non-univariate split at root node
Size and CV mean cost and SE of subtrees:
 Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   1       76   3.170E-01   6.144E-03   7.391E-03   3.206E-01   1.024E-02
   2       75   3.170E-01   6.144E-03   7.391E-03   3.206E-01   1.024E-02
   3       74   3.170E-01   6.144E-03   7.391E-03   3.206E-01   1.024E-02
   :
  44       15   3.065E-01   6.088E-03   5.357E-03   3.075E-01   7.515E-03
  45+      14   3.039E-01   6.074E-03   4.918E-03   3.025E-01   5.966E-03
  46++      9   3.043E-01   6.076E-03   5.104E-03   3.034E-01   4.222E-03
  47**      7   3.039E-01   6.074E-03   5.098E-03   3.092E-01   7.207E-03
  48        6   3.107E-01   6.111E-03   4.164E-03   3.121E-01   4.682E-03
  49        5   3.180E-01   6.150E-03   5.979E-03   3.145E-01   8.560E-03
  50        4   3.229E-01   6.175E-03   4.475E-03   3.194E-01   6.704E-03
  51        3   3.236E-01   6.178E-03   4.577E-03   3.211E-01   7.707E-03
  52        2   3.275E-01   6.197E-03   6.713E-03   3.211E-01   7.780E-03
  53        1   3.688E-01   6.371E-03   2.637E-03   3.670E-01   2.864E-03

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

```
0-SE tree based on median is marked with + and has 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.

Node cost is node misclassification cost divided by number of training cases
       Node    Total    Train    Predicted        Node    Split variable followed by
      label    cases    cases    class            cost    (+)fit variable(s)
          1     5735     5735    NoRHC       3.643E-01 cat1 +cat1 +pafi1
          2     1683     1683    RHC         4.225E-01 adld3p +adld3p +pafi1
          4     1183     1183    RHC         3.567E-01 wtkilo1 +wtkilo1 +pafi1
          8T     452      452    NoRHC       3.540E-01 pafi1 +pafi1 +hema1
          9T     731      731    RHC         3.010E-01 pafi1 +pafi1 +meanbp1
          5      500      500    NoRHC       4.160E-01 card +card +meanbp1
         10      345      345    NoRHC       3.420E-01 pot1 +pot1 +meanbp1
         20T     181      181    RHC         2.928E-01 meanbp1 +meanbp1 +resp1
         21T     164      164    NoRHC       2.683E-01 meanbp1 +meanbp1 +edu
         11T     155      155    NoRHC       3.677E-01 resp1 +resp1
          3     4052     4052    NoRHC       2.850E-01 pafi1 +pafi1 +crea1
          6T    1281     1281    NoRHC       3.599E-01 aps1 +aps1 +resp1
          7T    2771     2771    NoRHC       2.324E-01 meanbp1 +meanbp1 +crea1

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

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

Node 1: cat1 = "CHF", "MOSF w/Sepsis"
  Node 2: adld3p = NA
    Node 4: wtkilo1 <= 70.249970
      Node 8: Mean cost = 0.35398230
    Node 4: wtkilo1 > 70.249970 or NA
      Node 9: Mean cost = 0.30095759
  Node 2: adld3p /= NA
    Node 5: card = "Yes"
      Node 10: pot1 <= 3.9499510
        Node 20: Mean cost = 0.29281768
```

```
      Node 10: pot1 > 3.9499510 or NA
        Node 21: Mean cost = 0.26829268
     Node 5: card /= "Yes"
       Node 11: Mean cost = 0.36774194
 Node 1: cat1 /= "CHF", "MOSF w/Sepsis"
   Node 3: pafi1 <= 141.85938
     Node 6: Mean cost = 0.35987510
   Node 3: pafi1 > 141.85938 or NA
     Node 7: Mean cost = 0.23240707

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

Node 1: Intermediate node
 A case goes into Node 2 if cat1 = "CHF", "MOSF w/Sepsis"
 cat1 mode = ARF
 pafi1 mean = 222.27371
                                 Bandwidth
 Class      Number   Posterior   cat1   pafi1
 NoRHC        3551  0.6192E+00               1.4868E-02
 RHC          2184  0.3808E+00               1.2981E-02
 Number of training cases misclassified = 2089
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if adld3p = NA
 adld3p mean = 1.2340000
 pafi1 mean = 249.20858
                                     Bandwidth
 Class      Number   Posterior   adld3p   pafi1   Correlation
 NoRHC         774  0.4599E+00   1.1959E+00   7.6307E+01     0.0944
 RHC           909  0.5401E+00   6.3364E-01   6.8628E+01     0.0222
 Number of training cases misclassified = 711
 If node model is inapplicable due to missing values, predicted class is "RHC"
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if wtkilo1 <= 70.249970
 wtkilo1 mean = 77.015038
 pafi1 mean = 231.38524
                                     Bandwidth
 Class      Number   Posterior   wtkilo1   pafi1   Correlation
 NoRHC         488  0.4125E+00   1.3035E+01   9.4062E+01    -0.1043
 RHC           695  0.5875E+00   1.2650E+01   7.1161E+01    -0.0544
 Number of training cases misclassified = 422
 If node model is inapplicable due to missing values, predicted class is "RHC"
 ----------------------------
```

```
Node 8: Terminal node
 pafi1 mean = 244.88658
 hema1 mean = 30.163116
                                    Bandwidth
 Class      Number   Posterior   pafi1   hema1   Correlation
 NoRHC         238  0.5265E+00   1.1248E+02   5.8918E+00    -0.1432
 RHC           214  0.4735E+00   9.2951E+01   3.9603E+00     0.0123
 Number of training cases misclassified = 160
 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
 Number of training cases misclassified = 461
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------
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
 Number of training cases misclassified = 644
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------

 Classification matrix for training sample:
 Predicted      True class
 class          NoRHC      RHC
 NoRHC           3003     1091
 RHC              548     1093
 Total           3551     2184

 Number of cases used for tree construction: 5735
 Number misclassified: 1639
 Resubstitution estimate of mean misclassification cost: 0.28578901

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

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

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

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

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

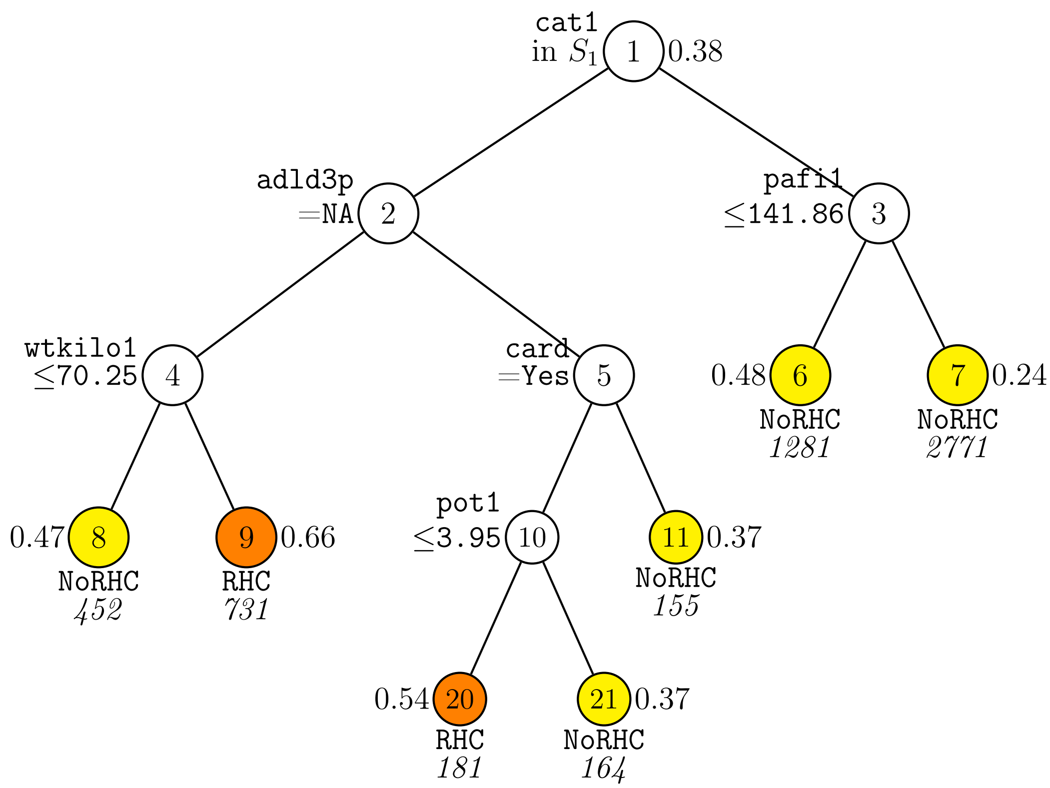

Figure 4: GUIDE v.41.0 0.250-SE classification tree for predicting `swang1` using bivariate kernel discriminant node models, estimated priors and unit misclassification costs. At each split, an observation goes to the left branch if and only if the condition is satisfied. S_1 = {`CHF`, `MOSF w/Sepsis`}. Predicted classes and sample sizes (in *italics*) printed below terminal nodes; class sample proportion for `swang1` = `RHC` beside nodes. Second best split variable (based on interaction test) at root node is `pafi1`.

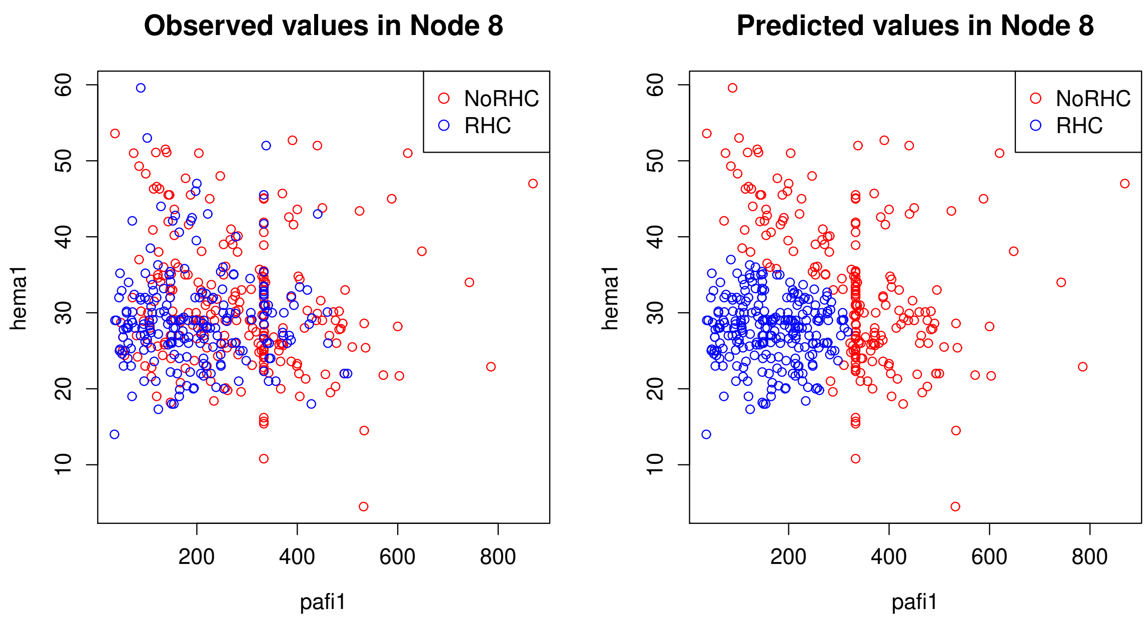

Figure 5: Plots of observed and predicted values for data in node 8 of tree in Figure 4

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
 23 N variables changed to S
 D variable is swang1
 Reading data file ...
 Number of records in data file: 5735
 Length of longest entry in data file: 19
 Checking for missing values ...
 Finished checking
 Missing values found among categorical variables
 Separate categories will be created for missing categorical variables
 Missing values found among non-categorical variables
 Recoding D values to integers
 Finished recoding
 Number of classes: 2
 Assigning integer codes to values of 30 categorical variables
 Finished assigning codes to 10 categorical variables
 Finished assigning codes to 20 categorical variables
 Finished assigning codes to 30 categorical variables
 Re-checking data ...
 Allocating missing value information ...
 Assigning codes to missing values, if any ...
 Finished processing 5000 of 5735 observations
 Data checks complete
 Creating missing value indicators ...
 Rereading data ...
 Class  #Cases     Proportion
 NoRHC    3551     0.61918047
 RHC      2184     0.38081953
      Total  #cases w/   #missing
     #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       5735          0       5157       10        0        0       23
     #P-var   #M-var   #B-var   #C-var   #I-var
          0        0        0       30        0
 Number of cases used for training: 5735
 Number of split variables: 53
 Number of cases excluded due to 0 W or missing D variable: 0
 Finished reading data file
 Default number of cross-validations:           10
 Input 1 to accept the default, 2 to change it ([1:2], <cr>=1):
 Best tree may be chosen based on mean or median CV estimate
```

```
Input 1 for mean-based, 2 for median-based ([1:2], <cr>=1):
Input number of SEs for pruning ([0.00:1000.00], <cr>=0.25):
Choose 1 for estimated priors, 2 for equal priors, 3 to input priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
Choose a split point selection method for numerical variables:
Choose 1 to use faster method based on sample quantiles
Choose 2 to use exhaustive search
Input 1 or 2 ([1:2], <cr>=2):
Default max. number of split levels: 15
Input 1 to accept this value, 2 to change it ([1:2], <cr>=1):
Default minimum node sample size is 57
Input 1 to use the default value, 2 to change it ([1:2], <cr>=1):
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): 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
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 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 ([1:53], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < nn2.in
```

4.4.2 Contents of nn2.out

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

```
Missing values found among non-categorical variables
Number of classes: 2
Training sample class proportions of D variable swang1:
Class  #Cases     Proportion
NoRHC    3551     0.61918047
RHC      2184     0.38081953

Summary information for training sample of size 5735
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                     #Codes/
                                                     Levels/
 Column  Name           Minimum      Maximum    Periods   #Missing
      2  cat1       c                                 9
      3  cat2       c                                 6     4535
      4  ca         c                                 3
     10  cardiohx   c                                 2
     11  chfhx      c                                 2
     12  dementhx   c                                 2
     13  psychhx    c                                 2
     14  chrpulhx   c                                 2
     15  renalhx    c                                 2
     16  liverhx    c                                 2
     17  gibledhx   c                                 2
     18  malighx    c                                 2
     19  immunhx    c                                 2
     20  transhx    c                                 2
     21  amihx      c                                 2
     22  age        s    18.04        101.8
     23  sex        c                                 2
     24  edu        s    0.000        30.00
     25  surv2md1   s    0.000       0.9620
     26  das2d3pc   s    11.00        33.00
     29  aps1       s    3.000        147.0
     30  scoma1     s    0.000        100.0
     31  meanbp1    s    10.00        259.0                    80
     32  wblc1      s    0.000        192.0
     33  hrt1       s    8.000        250.0                   159
     34  resp1      s    2.000        100.0                   136
     35  temp1      s    27.00        43.00
     36  pafi1      s    11.60        937.5
     37  alb1       s   0.3000        29.00
     38  hema1      s    2.000        66.19
     39  bili1      s   0.9999E-01    58.20
```

```
     40  crea1     s   0.9999E-01   25.10
     41  sod1      s    101.0       178.0
     42  pot1      s    1.100       11.90
     43  paco21    s    1.000       156.0
     44  ph1       s    6.579       7.770
     45  swang1    d                                2
     46  wtkilo1   s    19.50       244.0                  515
     47  dnr1      c                                2
     48  ninsclas  c                                6
     49  resp      c                                2
     50  card      c                                2
     51  neuro     c                                2
     52  gastr     c                                2
     53  renal     c                                2
     54  meta      c                                2
     55  hema      c                                2
     56  seps      c                                2
     57  trauma    c                                2
     58  ortho     c                                2
     59  adld3p    s    0.000       7.000                 4296
     60  urin1     s    0.000       9000.                 3028
     61  race      c                                3
     62  income    c                                4

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

 Constant fitted to cases with missing values in regressor variables
 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: 0.2500

 Nearest-neighbor node models
 Bivariate preference
 Estimated priors
 Unit misclassification costs
 Bivariate split highest priority
 Interaction splits 2nd priority; no linear splits
 Split values for N and S variables based on exhaustive search
 Maximum number of split levels: 15
```

```
Minimum node sample size: 57
Non-univariate split at root node
Size and CV mean cost and SE of subtrees:
 Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   1       76   3.151E-01   6.134E-03   6.384E-03   3.188E-01   1.012E-02
   2       75   3.151E-01   6.134E-03   6.384E-03   3.188E-01   1.012E-02
   3       74   3.151E-01   6.134E-03   6.384E-03   3.188E-01   1.012E-02
   :
  36       39   3.151E-01   6.134E-03   6.384E-03   3.188E-01   1.012E-02
  37*      38   3.151E-01   6.134E-03   6.384E-03   3.188E-01   1.012E-02
  38       36   3.153E-01   6.135E-03   5.373E-03   3.139E-01   8.226E-03
  39++     35   3.153E-01   6.135E-03   5.373E-03   3.139E-01   8.226E-03
  40       34   3.163E-01   6.141E-03   6.259E-03   3.173E-01   9.416E-03
  41       32   3.163E-01   6.141E-03   6.259E-03   3.173E-01   9.416E-03
  42       31   3.163E-01   6.141E-03   6.111E-03   3.173E-01   8.898E-03
  43**     29   3.163E-01   6.141E-03   6.111E-03   3.173E-01   8.898E-03
  44       27   3.172E-01   6.145E-03   6.350E-03   3.200E-01   9.397E-03
  45       23   3.179E-01   6.149E-03   6.020E-03   3.200E-01   9.328E-03
  46       17   3.194E-01   6.157E-03   5.933E-03   3.191E-01   9.860E-03
  47       16   3.200E-01   6.160E-03   6.248E-03   3.243E-01   9.698E-03
  48       15   3.182E-01   6.151E-03   5.834E-03   3.240E-01   8.063E-03
  49       14   3.184E-01   6.152E-03   5.997E-03   3.261E-01   8.891E-03
  50        9   3.184E-01   6.152E-03   5.997E-03   3.261E-01   8.891E-03
  51        7   3.217E-01   6.168E-03   5.279E-03   3.226E-01   6.723E-03
  52        5   3.250E-01   6.185E-03   6.166E-03   3.243E-01   1.047E-02
  53        1   3.439E-01   6.272E-03   4.168E-03   3.458E-01   7.691E-03

0-SE tree based on mean is marked with * and has 38 terminal nodes
0-SE tree based on median is marked with + and has 35 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 variable followed by
      label    cases    cases    class            cost    (+)fit variable(s)
          1     5735     5735    NoRHC         2.961E-01 cat1 +cat1 +pafi1
          2     1683     1683    RHC           4.029E-01 adld3p +adld3p +pafi1
          4     1183     1183    RHC           3.271E-01 wtkilo1 +wtkilo1 +pafi1
          8      452      452    NoRHC         2.942E-01 pafi1 +pafi1 +hema1
```

```
   16T    257     257    RHC          2.646E-01 hema1 +hema1 +ph1
   17     195     195    NoRHC        2.872E-01 age +age
   34T    137     137    NoRHC        3.139E-01 das2d3pc +das2d3pc
   35T     58      58    NoRHC        1.034E-01 -
    9     731     731    RHC          2.791E-01 pafi1 +pafi1 +meanbp1
   18     420     420    RHC          2.619E-01 sex +sex
   36     300     300    RHC          2.233E-01 resp1 +resp1 +edu
   72T     90      90    RHC          6.667E-02 -
   73     210     210    RHC          2.905E-01 edu +edu
  146T    138     138    RHC          2.826E-01 aps1 +aps1
  147T     72      72    RHC          1.806E-01 -
   37T    120     120    RHC          3.000E-01 crea1 +crea1
   19     311     311    RHC          2.990E-01 meanbp1 +meanbp1
   38     237     237    RHC          3.418E-01 resp1 +resp1
   76T     92      92    RHC          2.609E-01 -
   77T    145     145    RHC          3.103E-01 age +age
   39T     74      74    NoRHC        2.432E-01 -
    5     500     500    NoRHC        3.280E-01 card +card +meanbp1
   10     345     345    NoRHC        3.072E-01 pot1 +pot1 +meanbp1
   20T    181     181    RHC          2.652E-01 meanbp1 +meanbp1 +resp1
   21T    164     164    NoRHC        2.805E-01 meanbp1 +meanbp1 +edu
   11T    155     155    NoRHC        3.226E-01 resp1 +resp1
    3    4052    4052    NoRHC        2.848E-01 pafi1 +pafi1 +crea1
    6    1281    1281    NoRHC        3.052E-01 aps1 +aps1 +resp1
   12     855     855    NoRHC        4.234E-01 card +card +adld3p
   24T    272     272    RHC          3.125E-01 alb1 +alb1 +meanbp1
   25     583     583    NoRHC        3.585E-01 resp +resp
   50T    182     182    NoRHC        3.462E-01 edu +edu
   51T    401     401    NoRHC        2.693E-01 immunhx +immunhx +temp1
   13     426     426    RHC          3.427E-01 resp +resp +resp1
   26     224     224    RHC          3.080E-01 resp1 +resp1 +age
   52T    139     139    RHC          2.302E-01 ph1 +ph1
   53T     85      85    NoRHC        3.059E-01 -
   27     202     202    RHC          2.723E-01 paco21 +paco21
   54T     69      69    RHC          1.304E-01 -
   55T    133     133    RHC          2.857E-01 surv2md1 +surv2md1
    7    2771    2771    NoRHC        2.317E-01 meanbp1 +meanbp1 +crea1
   14    1456    1456    NoRHC        3.043E-01 adld3p +adld3p +crea1
   28    1095    1095    NoRHC        2.749E-01 wtkilo1 +wtkilo1 +aps1
   56T    316     316    NoRHC        1.677E-01 card +card +hema1
   57     779     779    NoRHC        3.389E-01 dementhx +dementhx +crea1
  114     695     695    NoRHC        3.367E-01 dnr1 +dnr1 +crea1
  228     617     617    NoRHC        2.966E-01 pafi1 +pafi1 +crea1
  456T    262     262    RHC          2.595E-01 cat2 +cat2 +crea1
  457     355     355    NoRHC        3.014E-01 paco21 +paco21 +crea1
  914     190     190    NoRHC        2.684E-01 ph1 +ph1 +crea1
```

```
       1828T     125      125    RHC            2.160E-01 crea1 +crea1 +pot1
       1829T      65       65    NoRHC          2.615E-01 -
        915T     165      165    NoRHC          2.667E-01 ph1 +ph1 +edu
        229T      78       78    NoRHC          2.692E-01 -
        115T      84       84    NoRHC          2.143E-01 -
         29T     361      361    NoRHC          1.856E-01 age +age +card
         15T    1315     1315    NoRHC          1.612E-01 hema1 +hema1 +card
Warning: tree very large, omitting node numbers in LaTeX file

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

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

Node 1: cat1 = "CHF", "MOSF w/Sepsis"
  Node 2: adld3p = NA
    Node 4: wtkilo1 <= 70.249970
      Node 8: pafi1 <= 254.50000
        Node 16: Mean cost = 0.26459144
      Node 8: pafi1 > 254.50000 or NA
        Node 17: age <= 75.961460
          Node 34: Mean cost = 0.31386861
        Node 17: age > 75.961460 or NA
          Node 35: Mean cost = 0.10344828
    Node 4: wtkilo1 > 70.249970 or NA
      Node 9: pafi1 <= 227.75000
        Node 18: sex = "Male"
          Node 36: resp1 <= 17.000000 or NA
            Node 72: Mean cost = 0.66666667E-1
          Node 36: resp1 > 17.000000
            Node 73: edu <= 12.410785
              Node 146: Mean cost = 0.28260870
            Node 73: edu > 12.410785 or NA
              Node 147: Mean cost = 0.18055556
        Node 18: sex /= "Male"
          Node 37: Mean cost = 0.30000000
      Node 9: pafi1 > 227.75000 or NA
        Node 19: meanbp1 <= 106.50000 or NA
          Node 38: resp1 <= 25.500000 or NA
            Node 76: Mean cost = 0.26086957
          Node 38: resp1 > 25.500000
            Node 77: Mean cost = 0.31034483
        Node 19: meanbp1 > 106.50000
          Node 39: Mean cost = 0.24324324
```

```
  Node 2: adld3p /= NA
    Node 5: card = "Yes"
      Node 10: pot1 <= 3.9499510
        Node 20: Mean cost = 0.26519337
      Node 10: pot1 > 3.9499510 or NA
        Node 21: Mean cost = 0.28048780
    Node 5: card /= "Yes"
      Node 11: Mean cost = 0.32258065
Node 1: cat1 /= "CHF", "MOSF w/Sepsis"
  Node 3: pafi1 <= 141.85938
    Node 6: aps1 <= 66.500000
      Node 12: card = "Yes"
        Node 24: Mean cost = 0.31250000
      Node 12: card /= "Yes"
        Node 25: resp = "No"
          Node 50: Mean cost = 0.34615385
        Node 25: resp /= "No"
          Node 51: Mean cost = 0.26932668
    Node 6: aps1 > 66.500000 or NA
      Node 13: resp = "Yes"
        Node 26: resp1 <= 41.000000
          Node 52: Mean cost = 0.23021583
        Node 26: resp1 > 41.000000 or NA
          Node 53: Mean cost = 0.30588235
      Node 13: resp /= "Yes"
        Node 27: paco21 <= 31.500000
          Node 54: Mean cost = 0.13043478
        Node 27: paco21 > 31.500000 or NA
          Node 55: Mean cost = 0.28571429
  Node 3: pafi1 > 141.85938 or NA
    Node 7: meanbp1 <= 69.500000 or NA
      Node 14: adld3p = NA
        Node 28: wtkilo1 <= 57.399995 or NA
          Node 56: Mean cost = 0.16772152
        Node 28: wtkilo1 > 57.399995
          Node 57: dementhx = "0"
            Node 114: dnr1 = "No"
              Node 228: pafi1 <= 216.15625
                Node 456: Mean cost = 0.25954198
              Node 228: pafi1 > 216.15625 or NA
                Node 457: paco21 <= 36.500000
                  Node 914: ph1 <= 7.4648440
                    Node 1828: Mean cost = 0.21600000
                  Node 914: ph1 > 7.4648440 or NA
                    Node 1829: Mean cost = 0.26153846
                Node 457: paco21 > 36.500000 or NA
```

```
                   Node 915: Mean cost = 0.26666667
             Node 114: dnr1 /= "No"
               Node 229: Mean cost = 0.26923077
           Node 57: dementhx /= "0"
             Node 115: Mean cost = 0.21428571
       Node 14: adld3p /= NA
         Node 29: Mean cost = 0.18559557
     Node 7: meanbp1 > 69.500000
       Node 15: Mean cost = 0.16121673

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

Node 1: Intermediate node
 A case goes into Node 2 if cat1 = "CHF", "MOSF w/Sepsis"
 Number of nearest neighbors = 9
 cat1 mode = ARF
 pafi1 mean = 222.27371
 Class      Number   Posterior
 NoRHC        3551  0.6192E+00
 RHC          2184  0.3808E+00
 Number of training cases misclassified = 1698
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if adld3p = NA
 Number of nearest neighbors = 8
 adld3p mean = 1.2340000 SD = 1.8633799
 pafi1 mean = 249.20858 SD = 104.96492
              correlation = 0.63530716E-1
 Class      Number   Posterior
 NoRHC         774  0.4599E+00
 RHC           909  0.5401E+00
 Number of training cases misclassified = 678
 If node model is inapplicable due to missing values, predicted class is "RHC"
 ----------------------------
 :
 :
Node 29: Terminal node
 Number of nearest neighbors = 6
 age mean = 62.145410
 card mode = No
 Class      Number   Posterior
 NoRHC         294  0.8144E+00
 RHC            67  0.1856E+00
 Number of training cases misclassified = 67
```

```
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------
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
 Number of training cases misclassified = 212
 If node model is inapplicable due to missing values, predicted class is "NoRHC"
 ----------------------------

 Classification matrix for training sample:
 Predicted      True class
 class          NoRHC       RHC
 NoRHC           3109       888
 RHC              442      1296
 Total           3551      2184

 Number of cases used for tree construction: 5735
 Number misclassified: 1330
 Resubstitution estimate of mean misclassification cost: 0.23190933

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

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

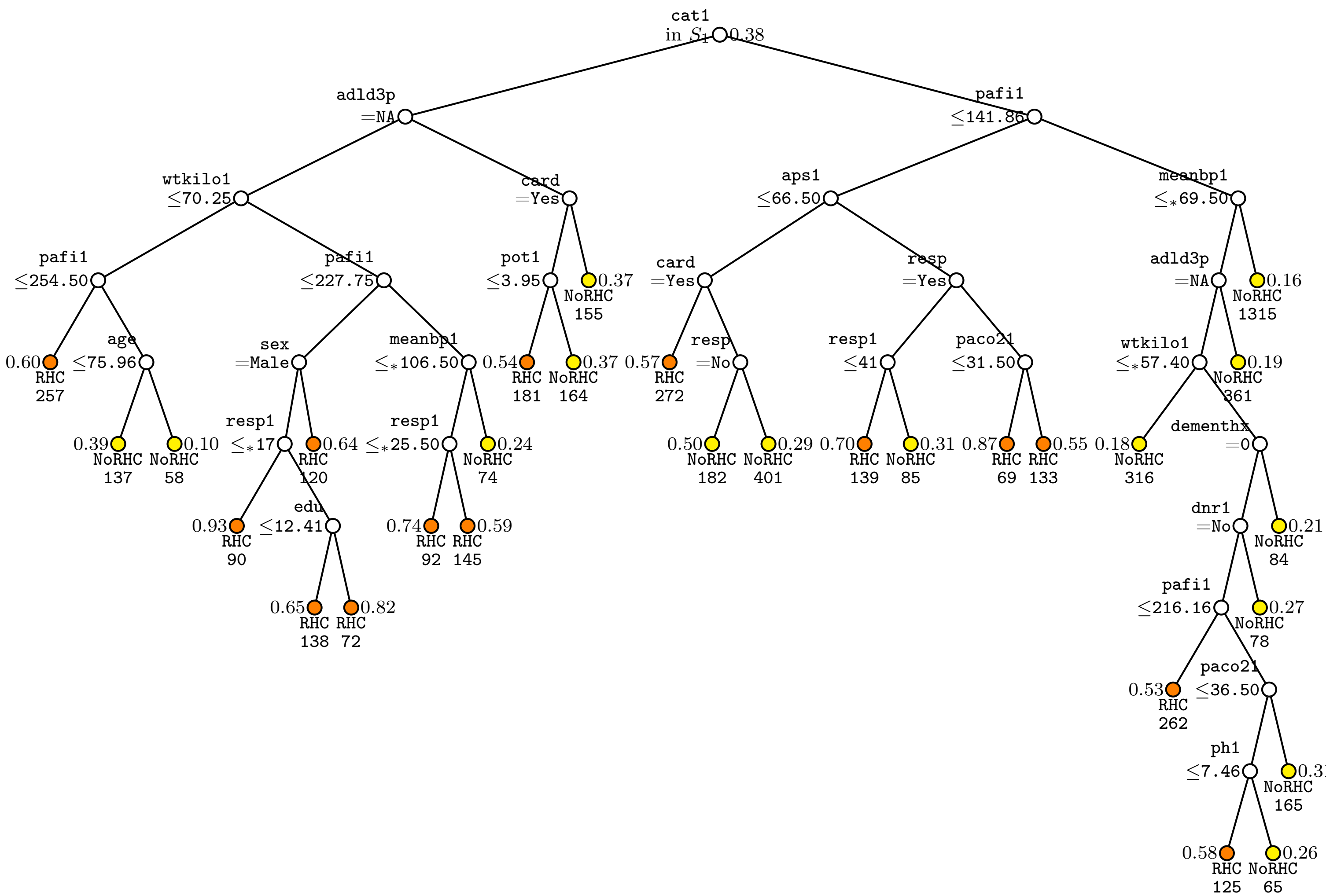

Figure 6: GUIDE v.41.0 0.250-SE classification tree for predicting `swang1` using bivariate nearest-neighbor node models, estimated priors and unit misclassification costs. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. S_1 = {`CHF`, `MOSF w/Sepsis`}. Predicted classes and sample sizes (in *italics*) printed below terminal nodes; class sample proportion for `swang1` = `RHC` beside nodes. Second best split variable (based on interaction test) at root node is `pafi1`.

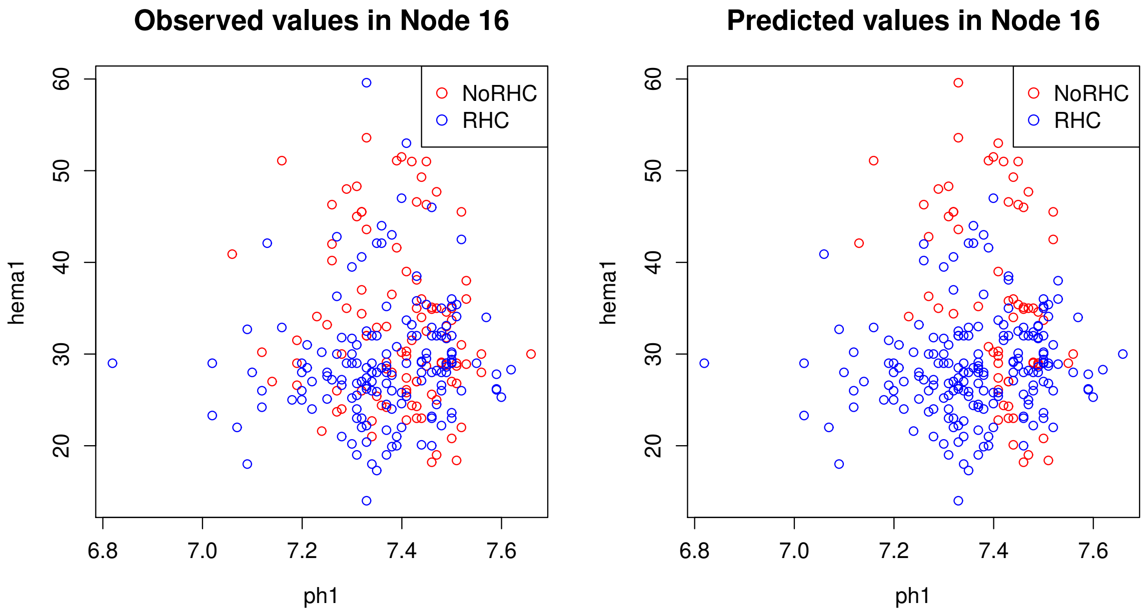

Figure 7: Plots of observed and predicted values for data in node 16 of tree in Figure 6

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

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

5 Missing-value flag variables: CE data

The Consumer Expenditure (CE) Survey is carried out by the Census Bureau for the Bureau of Labor Statistics (BLS). Conducted quarterly, the survey is a rotating panel survey that collects data on expenditures, income, and demographic characteristics of a sample of about 6000 consumer units (CUs) in the United States. After a CU is in the survey for four quarters, it is dropped and a new unit selected to replace it. The BLS defines *CU* and *reference person* of the CU as follows.

1. A CU consists of any of the following:
   (a) All members of a particular household who are related by blood, marriage, adoption, or other legal arrangements.
   (b) A person living alone or sharing a household with others or living as a roomer in a private home or lodging house or in permanent living quarters in a hotel or motel, but who is financially independent.
   (c) Two or more persons living together who use their incomes to make joint expenditure decisions. Financial independence is determined by spending behavior with regard to the three major expense categories: housing, food, and other living expenses. To be considered financially independent, the respondent must provide at least two of the three major expenditure categories, either entirely or in part.
2. A reference person of the CU is the first member mentioned by the respondent when asked "What are the names of all the persons living or staying here? Start with the name of the person or one of the persons who owns or rents the home." It is with respect to this person that the relationship of the other CU members is determined.

The data in the file `ce2021.txt` consist of 3965 observations on 550 variables. They are extracted from the second, third and fourth quarters of 2021 and the first quarter of 2022 of the Interview part of the CE survey. For the purpose of illustration and because it is not possible to link CUs between quarters, each CU in the sample is treated as unique. Table 11 in the Appendix gives the names and definitions of some of the 550 variables and their missing-value rates.

About 20% of the variables are *missing-value flags* that give the reasons for missing values. Table 8 lists the flag codes. A variable takes value `NA` (nonresponse) if its flag variable code is `A`, `B`, or `C`. The names of flag variables are typically the same as their parents, except for the addition or substitution of an underscore. For

Table 8: Codes for missing-value flag variables

A	Valid nonresponse; a response is not anticipated
B	Invalid nonresponse; nonresponse inconsistent with other data reported by CU
C	"Don't know", refusal, or other type of nonresponse
D	Valid unadjusted data value
T	Valid value topcoded or suppressed

example, INTRDVX_ is the flag variable for INTRDVX (amount of income received from interest and dividends). In this dataset, INTRDVX_ has no B codes and records with A codes are removed. Thirty-seven percent of the records (1478) have INTRDVX_ = C.

A T flag code indicates that the value of a variable 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. Usually, the reported values of the CUs in this group are replaced by their group mean. For example, below are the values of AGE2 (age of spouse) and AGE2_ in rows 112–117 of the data:

```
    AGE2 AGE2_
112   29     D
113   87     T
114   NA     A
115   57     D
116   87     T
117   NA     A
```

Respondents 113 and 116 are topcoded and have their values equal to 87, the mean of the top 3 percent of AGE2. 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 an earlier dataset.

Variable FINLWT21 is a sampling weight. For classification, GUIDE treats all observations with positive sampling weight equally; observations with non-positive weights are ignored in tree construction.

Missing-value flag variables are indicated by the letters "m" or "M" in the description file. To indicate to GUIDE to which variable is associated with which M variable, the latter must follow immediately after a B, C, N, P, or S variable in the description file. For example, the following lines from the file ce2021class.dsc indicate that DIRACC_ is the flag variable for DIRACC, AGE_REF_ is the flag variable for AGE_REF, and INCN_NW1 is the flag variable for INCNONW1.

```
ce2021.txt
NA
2
1 DIRACC n
2 DIRACC_ m
3 AGE_REF n
4 AGE_REF_ m
5 AGE2 n
6 AGE2_ m
7 AS_COMP1 n
8 AS_COMP2 n
9 AS_COMP3 n
10 AS_COMP4 n
11 AS_COMP5 n
12 BATHRMQ n
13 BATHRMQ_ m
14 BEDROOMQ n
15 BEDR_OMQ m
16 BLS_URBN n
17 BUILDING c
18 CUTENURE c
19 EARNCOMP c
20 EDUC_REF n
21 EDUCA2 n
22 EDUCA2_ m
:
50 INCNONW1 c
51 INCN_NW1 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` (which takes values `A`, `C`, `D`, and `T`) can take 7 forms:

1. `RETS_RVX = A` (only `A` flag values go left)
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 14):

1. INCNONW2 in S (where S is a subset of values of INCNONW2)

2. INCNONW2 = NA

3. INCNONW2 in S or INCN_NW2 in S^* (where S^* is a subset of flag codes)

5.1 Classification tree

This section shows how to construct a classification tree for predicting INTRDVX_ using the description file ce2021class.dsc.

5.1.1 Input file generation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: class.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: class.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
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: ce2021class.dsc
Reading data description file ...
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
384 N variables changed to S
D variable is INTRDVX_
Reading data file ...
```

```
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 3
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 47 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: S variable DIRACC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable ROTHRFLC is constant
Warning: S variable WELFREBX is constant
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
Class  #Cases     Proportion
C        1478     0.37276166
D        2431     0.61311475
T          56     0.01412358
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965          0       3965        1        0        0      384
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Number of cases used for training: 3965
Number of split variables: 431
Number of cases excluded due to 0 W or missing D variable: 0
Finished reading data file
Warning: No interaction tests; too many predictor variables
Choose 1 for estimated priors, 2 for equal priors, 3 to input priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
```

```
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
Warning: All positive weights treated as 1
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): class.tex
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: class.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: class.r
Input rank of top variable to split root node ([1:431], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < class.in
```

5.1.2 Contents of output file

```
Classification tree
Pruning by cross-validation
Data description file: ce2021class.dsc
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
384 N variables changed to S
D variable is INTRDVX_
Number of records in data file: 3965
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 DIRACC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable ROTHRFLC is constant
Warning: S variable WELFREBX is constant
Smallest and largest positive weights are  1.0725E+03 and 9.3902E+04
Training sample class proportions of D variable INTRDVX_:
Class  #Cases     Proportion
C        1478     0.37276166
D        2431     0.61311475
T          56     0.01412358
```

```
Summary information for training sample of size 3965
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     s    1.000        1.000                    170
      2  DIRACC_    m                                 2
      3  AGE_REF    s    18.00        87.00
      4  AGE_REF_   m                                 0
      5  AGE2       s    21.00        87.00                   1734
      6  AGE2_      m                                 1
      :
    407  INTRDVX_   d                                 3
    408  IRAB       s    1.000        6.000                   3831
    409  IRAB_      m                                 2
      :
    547  WHLFYR     c                                 1     3964
    548  WHLFYR_    m                                 1
    549  FFTAXOWE   s  -0.3368E+05   0.3997E+06
    550  FSTAXOWE   s   -3309.       0.7223E+05

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965          0       3965        1        0        0      384
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Number of cases used for training: 3965
Number of split variables: 431
Number of cases excluded due to 0 W or missing D variable: 0

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Warning: No interaction tests; too many predictor variables
Simple node models
Estimated priors
Unit misclassification costs
Warning: All positive weights treated as 1
Univariate split highest priority
```

```
No interaction splits
No linear splits
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 13
Minimum node sample size: 39
Top-ranked variables and 1-df chi-squared values at root node
     1  0.2367E+03   INCLASS2
     2  0.1936E+03   STATE
     3  0.1519E+03   ERANKH
     4  0.1350E+03   PSU
     :
   379  0.1031E-01   PUBTRACQ
   380  0.2768E-02   CARTKUPQ
   381  0.2626E-02   OTHLOAN
   382  0.6293E-03   TEXTILCQ

Size and CV mean cost and SE of subtrees:
 Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
   1       74   3.190E-01   7.402E-03   6.837E-03   3.161E-01   1.242E-02
   2       73   3.190E-01   7.402E-03   6.837E-03   3.161E-01   1.242E-02
   3       71   3.190E-01   7.402E-03   6.837E-03   3.161E-01   1.242E-02
   :
  29++     28   3.120E-01   7.358E-03   6.105E-03   3.052E-01   1.040E-02
  30       27   3.107E-01   7.350E-03   5.616E-03   3.086E-01   7.575E-03
  31       25   3.105E-01   7.348E-03   5.663E-03   3.086E-01   7.916E-03
  32**     21   3.072E-01   7.326E-03   4.146E-03   3.086E-01   5.017E-03
  33       14   3.115E-01   7.354E-03   6.153E-03   3.174E-01   1.043E-02
  34       11   3.102E-01   7.346E-03   7.847E-03   3.115E-01   1.483E-02
  35        8   3.185E-01   7.399E-03   6.886E-03   3.266E-01   1.041E-02
  36        3   3.359E-01   7.501E-03   9.174E-03   3.295E-01   1.160E-02
  37        2   3.417E-01   7.532E-03   1.010E-02   3.317E-01   1.779E-02
  38        1   3.869E-01   7.735E-03   8.543E-03   3.851E-01   1.446E-02

0-SE tree based on mean is marked with * and has 21 terminal nodes
0-SE tree based on median is marked with + and has 28 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          Interacting
      label    cases    cases    class           cost    variables      variable
          1     3965     3965    D          3.869E-01 INCLASS2
          2     3717     3717    D          3.613E-01 PSU
         4T      126      126    C          1.668E-01 OCCUCOD2
          5     3591     3591    D          3.448E-01 STATE
        10T     1360     1360    D          2.154E-01 INCNONW2
         11     2231     2231    D          4.236E-01 RETSURVX
         22     1609     1609    D          4.201E-01 FINDRETX
         44      968      968    D          4.928E-01 INCLASS2
         88      538      538    D          4.108E-01 STOCKX
        176      483      483    D          4.161E-01 RENTEQVX
        352      358      358    D          3.520E-01 STATE
       704T       45       45    C          2.668E-01 -
       705T      313      313    D          2.971E-01 INCNONW2
       353T      125      125    C          4.001E-01 ROOMSQ
       177T       55       55    D          3.636E-01 -
         89      430      430    C          4.395E-01 EARNCOMP
       178T       47       47    C          6.170E-01 -
        179      383      383    C          4.178E-01 REFGEN
        358      296      296    C          3.818E-01 REFGEN
        716      148      148    C          4.122E-01 DOMSRVCQ
       1432      107      107    C          3.272E-01 STATE
      2864T       41       41    D          3.659E-01 -
      2865T       66       66    C          1.365E-01 -
      1433T       41       41    D          4.146E-01 -
        717      148      148    C          3.514E-01 CASHCOCQ
      1434T      109      109    C          2.661E-01 STATE
      1435T       39       39    D          4.103E-01 -
        359       87       87    D          4.598E-01 PROPTXPQ
       718T       45       45    D          3.333E-01 -
       719T       42       42    C          4.048E-01 -
         45      641      641    D          3.105E-01 HLFBATHQ
         90      350      350    D          3.514E-01 FSTAXOWE
        180      299      299    D          2.977E-01 OCCUCOD1
       360T       43       43    C          3.489E-01 -
       361T      256      256    D          2.383E-01 ELCTRCPQ
       181T       51       51    C          3.334E-01 -
        91T      291      291    D          2.612E-01 HOUSOPPQ
         23      622      622    D          4.325E-01 RETSURVX
        46T       86       86    C          1.397E-01 STATE
        47T      536      536    D          3.638E-01 STOCKYRX
         3T      248      248    C          2.299E-01 FINCBTAX
Warning: tree very large, omitting node numbers in LaTeX file
```

```
Number of terminal nodes of final tree: 21
Total number of nodes of final tree: 41
Second best split variable (based on curvature test) at root node is STATE

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

Node 1: INCLASS2 <= 6.5000000
  Node 2: PSU = "S49F"
    Node 4: C
  Node 2: PSU /= "S49F"
    Node 5: STATE = "2", "6", "10", "11", "21", "24", "25", "27", "31", "40",
             "41", "47", "48", "49"
      Node 10: D
    Node 5: STATE /= "2", "6", "10", "11", "21", "24", "25", "27", "31", "40",
             "41", "47", "48", "49"
      Node 11: RETSURVX = NA & RETS_RVX = "A"
        Node 22: FINDRETX <= 391.50000
          Node 44: INCLASS2 <= 4.5000000
            Node 88: STOCKX = NA & STOCKX_ = "A"
              Node 176: RENTEQVX <= 1990.5000 or NA
                Node 352: STATE = "13", "17", "19", "22", "28", "32", "45", "46"
                  Node 704: C
                Node 352: STATE /= "13", "17", "19", "22", "28", "32", "45", "46"
                  Node 705: D
              Node 176: RENTEQVX > 1990.5000
                Node 353: C
            Node 88: not (STOCKX = NA & STOCKX_ = "A")
              Node 177: D
          Node 44: INCLASS2 > 4.5000000 or NA
            Node 89: EARNCOMP = "7", "8"
              Node 178: C
            Node 89: EARNCOMP /= "7", "8"
              Node 179: REFGEN <= 4.5000000
                Node 358: REFGEN <= 3.5000000
                  Node 716: DOMSRVCQ <= 12.000000
                    Node 1432: STATE = "26", "39", "42", "46", "51", "NA"
                      Node 2864: D
                    Node 1432: STATE /= "26", "39", "42", "46", "51", "NA"
                      Node 2865: C
                  Node 716: DOMSRVCQ > 12.000000 or NA
                    Node 1433: D
                Node 358: REFGEN > 3.5000000 or NA
                  Node 717: CASHCOCQ <= 30.000000
                    Node 1434: C
```

```
                   Node 717: CASHCOCQ > 30.000000 or NA
                     Node 1435: D
               Node 179: REFGEN > 4.5000000 or NA
                 Node 359: PROPTXPQ <= 287.50000
                   Node 718: D
                 Node 359: PROPTXPQ > 287.50000 or NA
                   Node 719: C
         Node 22: FINDRETX > 391.50000 or NA
           Node 45: HLFBATHQ <= 0.50000000
             Node 90: FSTAXOWE <= 10256.500
               Node 180: OCCUCOD1 = "6", "7", "11", "12", "15"
                 Node 360: C
               Node 180: OCCUCOD1 /= "6", "7", "11", "12", "15"
                 Node 361: D
             Node 90: FSTAXOWE > 10256.500 or NA
               Node 181: C
           Node 45: HLFBATHQ > 0.50000000 or NA
             Node 91: D
       Node 11: not (RETSURVX = NA & RETS_RVX = "A")
         Node 23: RETSURVX = NA
           Node 46: C
         Node 23: RETSURVX /= NA
           Node 47: 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 = 4.6266367
 Class      Number   Posterior
 C            1478  0.3728E+00
 D            2431  0.6131E+00
 T              56  0.1412E-01
 Number of training cases misclassified = 1534
 Predicted class is D
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if PSU = "S49F"
 PSU mode = "NA"
 Class      Number   Posterior
 C            1287  0.3462E+00
 D            2374  0.6387E+00
 T              56  0.1507E-01
```

```
 Number of training cases misclassified = 1343
 Predicted class is D
 ----------------------------
Node 4: Terminal node
 Class      Number   Posterior
 C             105  0.8332E+00
 D              21  0.1668E+00
 T               0  0.3561E-05
 Number of training cases misclassified = 21
 Predicted class is C
 ----------------------------
:
:
Node 47: Terminal node
 Class      Number   Posterior
 C             185  0.3451E+00
 D             341  0.6362E+00
 T              10  0.1866E-01
 Number of training cases misclassified = 195
 Predicted class is D
 ----------------------------
Node 3: Terminal node
 Class      Number   Posterior
 C             191  0.7701E+00
 D              57  0.2299E+00
 T               0  0.3561E-05
 Number of training cases misclassified = 57
 Predicted class is C
 ----------------------------

 Classification matrix for training sample:
 Predicted      True class
 class             C         D         T
 C               720       255        13
 D               758      2176        43
 T                 0         0         0
 Total          1478      2431        56

 Number of cases used for tree construction: 3965
 Number misclassified: 1069
 Resubstitution estimate of mean misclassification cost: 0.26960908

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

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

Node 1: `INCLASS2` ≤ 6.50 with missing values going right.

Node 11: `RETS_RVX` $=$ `A` (some missing values go left, all non-missing and some missing values go right).

Node 23: `RETSURVX` $=$ `NA` (all missing values go left, nonmissing go right).

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 `class.fit` are given below. The posterior probabilities of predicting class `T` are 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     4        "C"        "C"  0.83322E+00  0.16678E+00  0.35612E-05
     y    10        "D"        "D"  0.19779E+00  0.78456E+00  0.17647E-01
     y    10        "D"        "D"  0.19779E+00  0.78456E+00  0.17647E-01
     y  1434        "D"        "C"  0.73385E+00  0.26614E+00  0.35612E-05
     y    10        "D"        "D"  0.19779E+00  0.78456E+00  0.17647E-01
     y    91        "T"        "D"  0.23711E+00  0.73883E+00  0.24055E-01
     y    91        "D"        "D"  0.23711E+00  0.73883E+00  0.24055E-01
     y    10        "D"        "D"  0.19779E+00  0.78456E+00  0.17647E-01
```

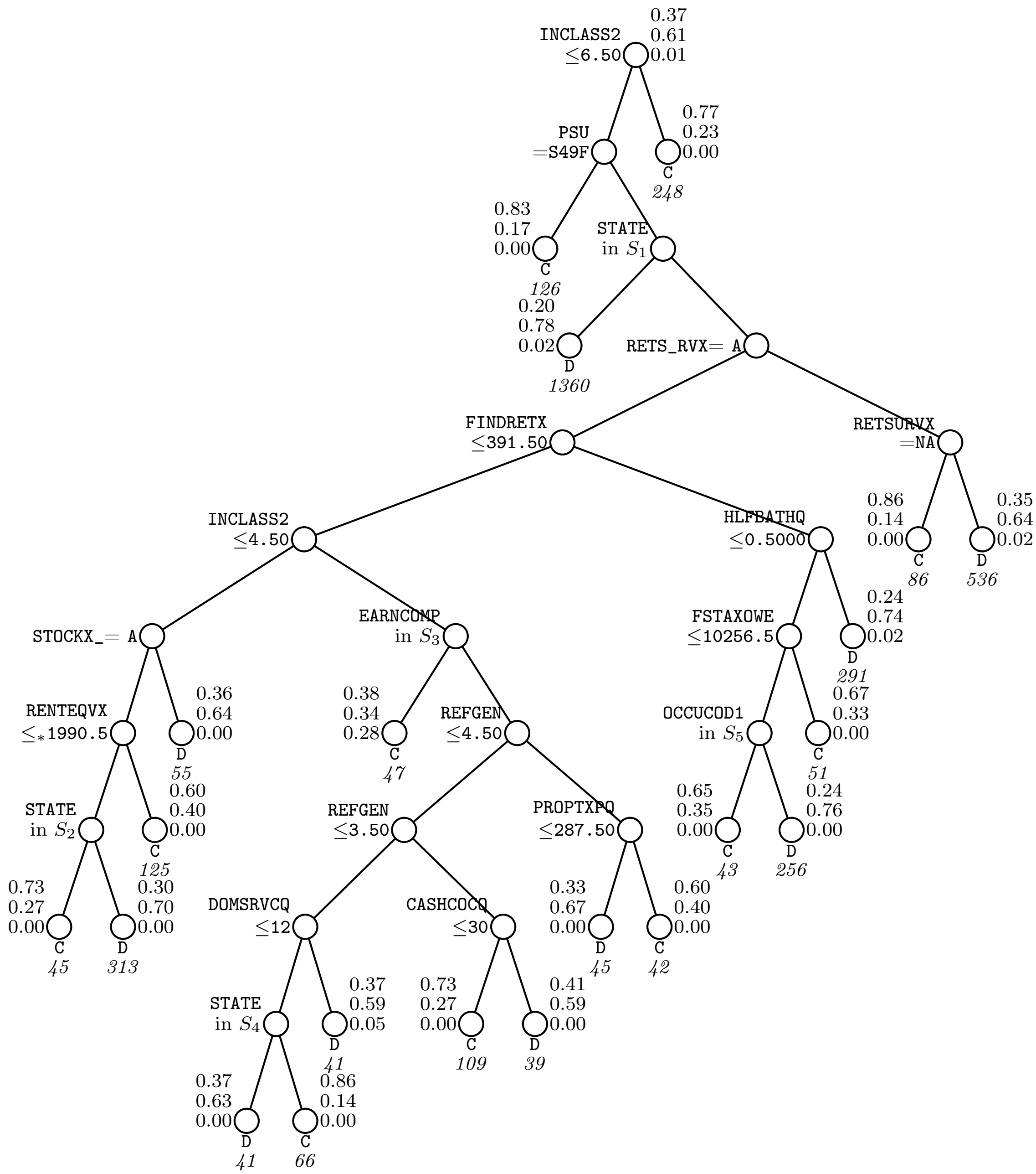

Figure 8: GUIDE classification tree for predicting INTRDVX_ using estimated priors and unit misclassification costs. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '≤ or missing'. S_1 = {2, 6, 10, 11, 21, 24, 25, 27, 31, 40, 41, 47, 48, 49}. S_2 = {13, 17, 19, 22, 28, 32, 45, 46}. S_3 = {7, 8}. S_4 = {26, 39, 42, 46, 51, NA}. S_5 = {6, 7, 11, 12, 15}. Predicted classes and sample sizes (in *italics*) printed below terminal nodes; class sample proportions for INTRDVX_ = C, D, and T, respectively, beside nodes. Second best split variable at root node is STATE.

6 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 (`d`) variable and excluding (`x`) its flag `INTRDVX_`. The description file is `ce2021reg.dsc`, which sets `FINLWT21` as a weight (`w`) variable.

6.1 Piecewise constant

6.1.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: cons.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: cons.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended for prediction)
Choose 1 for multiple regression
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise 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: ce2021reg.dsc
Reading data description file ...
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
384 N variables changed to S
D variable is INTRDVX
```

```
Reading data file ...
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 47 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: S variable DIRACC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable ROTHRFLC is constant
Warning: S variable WELFREBX is constant
Warning: S variable OTHLYRBX is constant
Warning: S variable OTHLNYRB is constant
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1        0        0      384
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Weight variable FINLWT21 in column: 31
Number of cases used for training: 2487
Number of split variables: 431
Number of cases excluded due to 0 W or missing D variable: 1478
Finished reading data file
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction tests; too many predictor variables
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): cons.tex
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: cons.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: cons.r
Input rank of top variable to split root node ([1:431], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < cons.in
```

6.1.2 Contents of `cons.out`

```
Least squares regression tree
Pruning by cross-validation
Data description file: ce2021reg.dsc
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
384 N variables changed to S
D variable is INTRDVX
Piecewise constant model
Number of records in data file: 3965
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 DIRACC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable ROTHRFLC is constant
Warning: S variable WELFREBX is constant
Warning: S variable OTHLYRBX is constant
Warning: S variable OTHLNYRB is constant
Smallest and largest positive weights are  1.0725E+03 and 9.3902E+04

Summary information for training sample of size 2487 (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    s    1.000       1.000                    125
      2  DIRACC_   m                                 2
      3  AGE_REF   s    19.00       87.00
      4  AGE_REF_  m                                 0
      5  AGE2      s    21.00       87.00                   1092
      6  AGE2_     m                                 1
      :
     30  FIND_ETX  m                                 0
     31  FINLWT21  w    1072.      0.9390E+05
     32  FJSSDEDX  s    0.000      0.4366E+05
     33  FJSS_EDX  m                                 0
      :
    406  INTRDVX   d    1.000      0.1413E+06
      :
    547  WHLFYR    c                                 1      2487
    548  WHLFYR_   m                                 1
    549  FFTAXOWE  s  -0.3368E+05  0.3380E+06
    550  FSTAXOWE  s   -3074.      0.5654E+05

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1        0        0      384
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Weight variable FINLWT21 in column: 31
Number of cases used for training: 2487
Number of split variables: 431
Number of cases excluded due to 0 W or missing D variable: 1478

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Weighted error estimates used for pruning
Warning: No interaction tests; too many predictor variables
No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 12
Minimum node sample size: 24
Top-ranked variables and 1-df chi-squared values at root node
     1  0.8297E+02   REFGEN
     2  0.8111E+02   AGE_REF
```

```
     3  0.7066E+02   INCNONW1
     4  0.6985E+02   STOCKX
     :
   381  0.3923E-02   HORREF1
   382  0.3569E-02   TGASMOTC
   383  0.1029E-03   MAINRPPQ

 Size and CV MSE and SE of subtrees:
  Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
    1       79   5.890E+12   7.198E+11   4.740E+11   5.838E+12   7.710E+11
    2       78   5.890E+12   7.198E+11   4.740E+11   5.838E+12   7.710E+11
    :
   39       21   5.860E+12   7.167E+11   4.808E+11   5.830E+12   8.050E+11
   40+      14   5.859E+12   7.168E+11   4.806E+11   5.822E+12   8.039E+11
   41        9   5.869E+12   7.194E+11   4.902E+11   5.849E+12   8.075E+11
   42**      8   5.789E+12   7.184E+11   5.133E+11   5.860E+12   8.402E+11
   43        5   6.554E+12   8.516E+11   7.323E+11   6.439E+12   1.068E+12
   44        4   7.614E+12   9.359E+11   7.113E+11   8.502E+12   1.107E+12
   45        1   8.287E+12   1.032E+12   6.955E+11   8.542E+12   7.418E+11

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

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

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

 D-mean is weighted mean of INTRDVX in the node
 Cases fit give the number of cases used to fit node
 MSE is residual sum of squares divided by number of cases in node
        Node    Total    Cases Matrix    Node      Node          Split
        label   cases      fit  rank    D-mean     MSE           variable
           1     2487     2487     1  5.131E+03  8.287E+12   REFGEN
           2     1914     1914     1  6.476E+03  1.027E+13   INC_RANK
           4T    1345     1345     1  2.823E+03  1.657E+12   REF_RACE
           5      569      569     1  1.498E+04  2.828E+13   EARNCOMP
          10       75       75     1  5.278E+04  7.423E+13   RETSURV
```

```
        20T      46       46     1  2.764E+04  4.076E+13   -
        21T      29       29     1  8.586E+04  7.873E+13   -
        11      494      494     1  9.170E+03  1.560E+13   FFTAXOWE
        22T     247      247     1  2.647E+03  3.560E+12   UNISTRQ
        23      247      247     1  1.570E+04  2.573E+13   AGE2
        46T     156      156     1  8.036E+03  1.274E+13   FINCBTAX
        47       91       91     1  3.030E+04  4.131E+13   BATHRMQ
        94T      47       47     1  1.087E+04  1.582E+13   -
        95T      44       44     1  4.670E+04  5.504E+13   -
         3T     573      573     1  9.804E+02  1.105E+12   PSU

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 AGE_REF

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

Node 1: REFGEN <= 4.5000000
  Node 2: INC_RANK <= 0.84018625
    Node 4: INTRDVX-mean = 2822.6445
  Node 2: INC_RANK > 0.84018625 or NA
    Node 5: EARNCOMP = "8"
      Node 10: RETSURV = "1"
        Node 20: INTRDVX-mean = 27641.282
      Node 10: RETSURV /= "1"
        Node 21: INTRDVX-mean = 85859.276
    Node 5: EARNCOMP /= "8"
      Node 11: FFTAXOWE <= 27769.500
        Node 22: INTRDVX-mean = 2646.5367
      Node 11: FFTAXOWE > 27769.500 or NA
        Node 23: AGE2 <= 56.500000 or NA
          Node 46: INTRDVX-mean = 8036.3341
        Node 23: AGE2 > 56.500000
          Node 47: BATHRMQ <= 2.5000000
            Node 94: INTRDVX-mean = 10866.520
          Node 47: BATHRMQ > 2.5000000 or NA
            Node 95: INTRDVX-mean = 46702.398
Node 1: REFGEN > 4.5000000 or NA
  Node 3: INTRDVX-mean = 980.35292

***************************************************************
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 REFGEN <= 4.5000000
 REFGEN mean = 3.4302980
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant      5131.        12.43       0.2220E-15
 INTRDVX mean = 5130.60
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if INC_RANK <= 0.84018625
 INC_RANK mean = 0.64479638
 ----------------------------
Node 4: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      2823.        11.12       0.3331E-15
 INTRDVX mean = 2822.64
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if EARNCOMP = "8"
 EARNCOMP mode = "2"
 ----------------------------
:
:
Node 94: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.1087E+05    2.340       0.2368E-01
 INTRDVX mean = 10866.5
 ----------------------------
Node 95: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.4670E+05    6.082       0.2775E-06
 INTRDVX mean = 46702.4
 ----------------------------
Node 3: Terminal node
```

```
Coefficients of least squares regression functions:
Regressor    Coefficient  t-stat      p-value
Constant      980.4        3.232       0.1300E-02
INTRDVX mean = 980.353
----------------------------
Proportion of variance (R-squared) explained by tree model: 0.3766

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

In the above results, the pruned tree is marked with two asterisks (tree #42). It has 8 terminal nodes and a cross-validation estimate of prediction mean squared error of 5.789E+12. Figure 9 shows the tree. The first split is on `REFGEN` ≤ 4, meaning that millenials (`REFGEN`=5) go to node 3, which is terminal. The first 7 entries of `cons.fit` below show that the 1st 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 4 and has a predicted value of $2822.64.

```
train          node       observed          predicted
    n             4       NA                 2822.64
    y             4       1087.00            2822.64
    y             4       1000.00            2822.64
    y             4       300.000            2822.64
    y             3       10.0000            980.353
    y            94       141304.            10866.5
    y             4       55.0000            2822.64
```

6.2 Piecewise simple polynomial

GUIDE can also fit a simple polynomial regression model in each node of the form

$$y = \beta_0 + \sum_{k=1}^{p} \beta_k x^k + \epsilon \tag{1}$$

where p is the degree of polynomial desired and x is selected from the set of `n` and `f` variables. The variable x is the one among all `n` and `f` variables that yields the smallest sum of squared residuals. Variable x can vary from node to node. If there are missing values in the x variable, GUIDE fits two separate models to the data in the node: model (1) to the observations with complete values in x and y and a

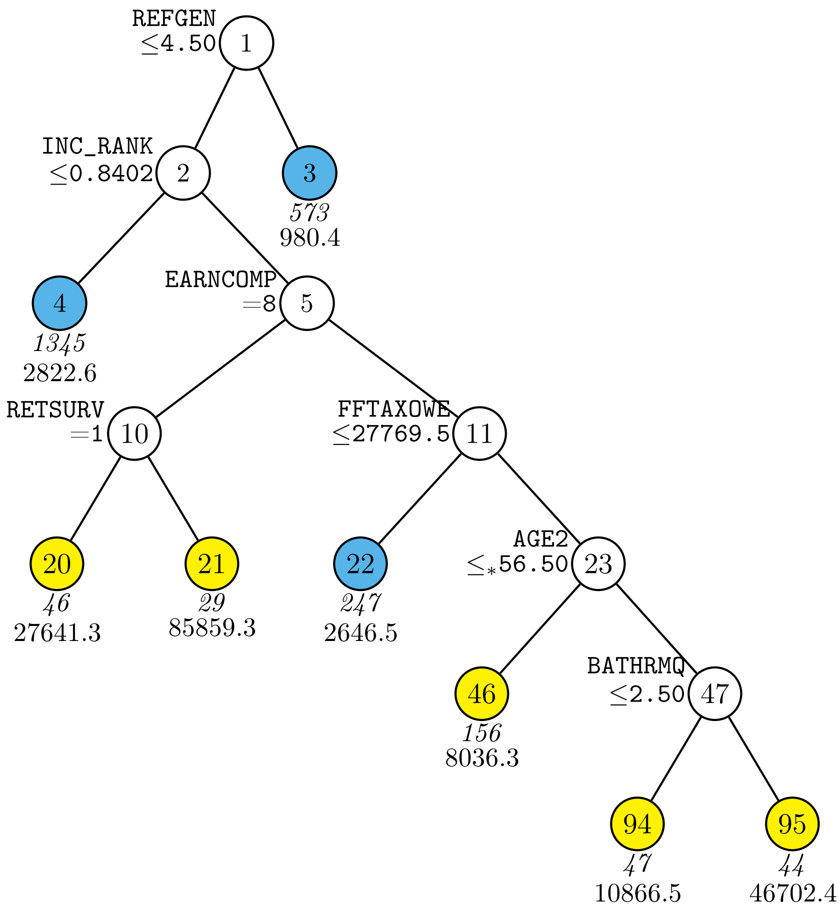

Figure 9: GUIDE v.41.0 0.250-SE piecewise-constant weighted least-squares regression tree for predicting `INTRDVX`. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. Sample size (in *italics*) and weighted mean of `INTRDVX` printed below nodes. Terminal nodes with means above and below value of 5130.6 at root node are colored yellow and skyblue respectively. Second best split variable at root node is `AGE_REF`.

constant ($y = \beta_0 + \epsilon$) to those with missing values in x. This is equivalent to imputing missing x values with a constant c and adding the missing-value indicator $I(x = \texttt{NA})$ as linear predictor:

$$y = \beta_0 + \sum_{k=1}^{p} \beta_k x_1^k + \beta_2 x_2 + \epsilon$$

where $x_1 = xI(x \neq \texttt{NA}) + cI(x = \texttt{NA})$ and $x_2 = I(x = \texttt{NA})$. The predicted values are independent of c but the least-squares estimates of the β coefficients are not.

Truncation note: Extrapolation can adversely affect the prediction accuracy of parametric models. To guard against extrapolation, GUIDE has several options to truncate the predicted values, with the default being to truncate the predicted values if they fall outside the range of the observed values. The option of no truncation is available as well. Default truncation is used in this user guide.

6.2.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: lin.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: lin.out
Input 1 for classification, 2 for regression, 3 for propensity score grouping
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended for prediction)
Choose 1 for multiple regression
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise 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: ce2021reg.dsc
Reading data description file ...
Training sample file: ce2021.txt
Missing value code: NA
```

```
Records in data file start on line 2
Number of M variables associated with C variables: 19
D variable is INTRDVX
Reading data file ...
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 47 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: N variable DIRACC is constant
Warning: N variable TOTHVHRP is constant
Warning: N variable TOTHVHRC is constant
Warning: N variable ROTHRFLC is constant
Warning: N variable WELFREBX is constant
Warning: N variable OTHLYRBX is constant
Warning: N variable OTHLNYRB is constant
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1      384        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Weight variable FINLWT21 in column: 31
Number of cases used for training: 2487
Number of split variables: 431
Number of cases excluded due to 0 W or missing D variable: 1478
Finished reading data file
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction tests; too many predictor variables
```

```
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): lin.tex
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>=2):
Input file name: lin.var
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: lin.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: lin.r
Input rank of top variable to split root node ([1:431], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < lin.in
```

6.2.2 Partial output

```
Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       53   8.181E+12   1.005E+12   1.178E+12   7.611E+12   1.590E+12
   2       52   8.181E+12   1.005E+12   1.178E+12   7.612E+12   1.590E+12
   3       51   8.180E+12   1.005E+12   1.178E+12   7.612E+12   1.589E+12
   :
  25       13   6.783E+12   8.117E+11   9.406E+11   5.596E+12   1.393E+12
  26+       8   6.607E+12   7.917E+11   8.858E+11   5.223E+12   1.476E+12
  27**      7   6.470E+12   7.870E+11   8.430E+11   5.256E+12   1.496E+12
  28        5   6.923E+12   8.726E+11   7.409E+11   6.654E+12   9.529E+11
  29        4   7.212E+12   9.026E+11   7.598E+11   7.222E+12   1.033E+12
  30        1   8.303E+12   1.011E+12   7.002E+11   8.514E+12   8.534E+11

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.

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     2487      209     2  5.131E+03  7.702E+12  0.0710 CUTENURE  +STOCKYRX
          2      855      855     2  8.856E+03  1.219E+13  0.1292 FJSSDEDX  +FINCBTAX
          4      578      578     2  9.919E+03  1.137E+13  0.2914 FINCBTAX  +FINCBTAX
          8T     500      500     2  3.863E+03  1.323E+12  0.3043 INC_RANK  +ALCBEVCQ
          9       78       47     2  5.014E+04  4.793E+13  0.2919 RETSURV  -RETSURVX
         18T      47       47     2  2.613E+04  2.689E+13  0.2996 -  +FULOILCQ
         19T      31        7     2  8.229E+04  3.579E+13  0.5077 -  -ROYESTX
          5T     277      277     2  6.780E+03  4.930E+12  0.4882 PERINSCQ  +ETOTALC
          3     1632     1087     2  3.221E+03  4.540E+12  0.1032 RENTEQVX  +RENTEQVX
          6T    1558     1558     2  2.084E+03  2.137E+12  0.0673 STATE  +VELECTRC
          7       74       74     2  2.706E+04  4.068E+13  0.1976 OWNDWEPQ  -FSALARYX
         14T      38       38     2  3.714E+04  4.466E+13  0.3394 -  +DMSXCCPQ
         15T      36       36     2  1.651E+04  3.766E+12  0.8758 -  +ECARTKUC

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 REFGEN

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

Node 1: CUTENURE = "2", "6"
  Node 2: FJSSDEDX <= 2720.0000
    Node 4: FINCBTAX <= 114750.50
      Node 8: INTRDVX-mean = 3863.4422
    Node 4: FINCBTAX > 114750.50 or NA
      Node 9: RETSURV = "1"
        Node 18: INTRDVX-mean = 26127.783
      Node 9: RETSURV /= "1"
        Node 19: INTRDVX-mean = 82288.430
  Node 2: FJSSDEDX > 2720.0000 or NA
    Node 5: INTRDVX-mean = 6780.2396
Node 1: CUTENURE /= "2", "6"
  Node 3: RENTEQVX <= 4374.0000 or NA
    Node 6: INTRDVX-mean = 2083.6953
  Node 3: RENTEQVX > 4374.0000
    Node 7: OWNDWEPQ <= 5530.5000
      Node 14: INTRDVX-mean = 37135.317
    Node 7: OWNDWEPQ > 5530.5000 or NA
```

```
      Node 15: INTRDVX-mean = 16508.102

 ***************************************************************
 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 CUTENURE = "2", "6"
 CUTENURE mode = "1"
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      3167.        2.042       0.4246E-01
 STOCKYRX     0.1749E-01    11.42        0.000        0.000       0.3617E+06   0.5450E+07
 If regressors have missing values, predicted function value = 4720.7960
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if FJSSDEDX <= 2720.0000
 FJSSDEDX mean = 3273.8110
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if FINCBTAX <= 114750.50
 FINCBTAX mean = 60208.956
 ----------------------------
Node 8: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      2892.        8.287       0.9992E-15
 ALCBEVCQ      19.81        14.76        0.000        0.000         49.04        4670.
 If regressors have missing values, predicted function value = 3863.4422
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
Node 9: Intermediate node
 A case goes into Node 18 if RETSURV = "1"
 RETSURV mode = "1"
```

```
 ----------------------------
Node 18: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.2096E+05    3.886       0.3316E-03
 FULOILCQ      181.5        4.388       0.6841E-04    0.000        28.46        944.0
 If regressors have missing values, predicted function value = 26127.783
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
Node 19: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.1215E+06    8.524       0.3657E-03
 ROYESTX     -0.5119       -6.012       0.1830E-02    24.00       0.1298E+06   0.2300E+06
 If regressors have missing values, predicted function value = 97671.707
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
Node 5: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -953.9      -0.9683       0.3337
 ETOTALC      0.7695        16.20        0.000        0.000       0.1005E+05   0.1615E+06
 If regressors have missing values, predicted function value = 6780.2396
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if RENTEQVX <= 4374.0000 or NA
 RENTEQVX mean = 2434.6988
 ----------------------------
Node 6: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      1875.        7.721       0.2043E-13
 VELECTRC      166.1        10.59       0.4441E-15    0.000        1.258        484.0
 If regressors have missing values, predicted function value = 2083.6953
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if OWNDWEPQ <= 5530.5000
 OWNDWEPQ mean = 6589.8800
 ----------------------------
Node 14: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.1346E+05    1.500       0.1422
 DMSXCCPQ      112.6        4.300       0.1244E-03    0.000        210.3        1300.
```

```
 If regressors have missing values, predicted function value = 37135.317
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
Node 15: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      7371.        3.372       0.1874E-02
 ECARTKUC      302.2        15.49        0.000        0.000        30.23        415.0
 If regressors have missing values, predicted function value = 16508.102
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
 Proportion of variance (R-squared) explained by tree model: 0.5504

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

The pruned tree (marked with two asterisks) has 7 terminal nodes and a cross-validation estimate of prediction mean squared error of 6.470E+12.

The tree is shown in Figure 10. 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 mean of the `d` variable above and below the mean at the root node are colored yellow and purple, respectively.

6.2.3 Plots of data

Figure 11 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 12, which reads the files `lin.fit` and `lin.var`. The contents of the latter 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   ALCBEVCQ  2892.      19.81    1.000  0.1413E+6
18   FULOILCQ  0.2096E+5  181.5    1.000  0.1413E+6
19   ROYESTX   0.1215E+6  -0.5119  1.000  0.1413E+6
5    ETOTALC   -953.9     0.7695   1.000  0.1413E+6
6    VELECTRC  1875.      166.1    1.000  0.1413E+6
```

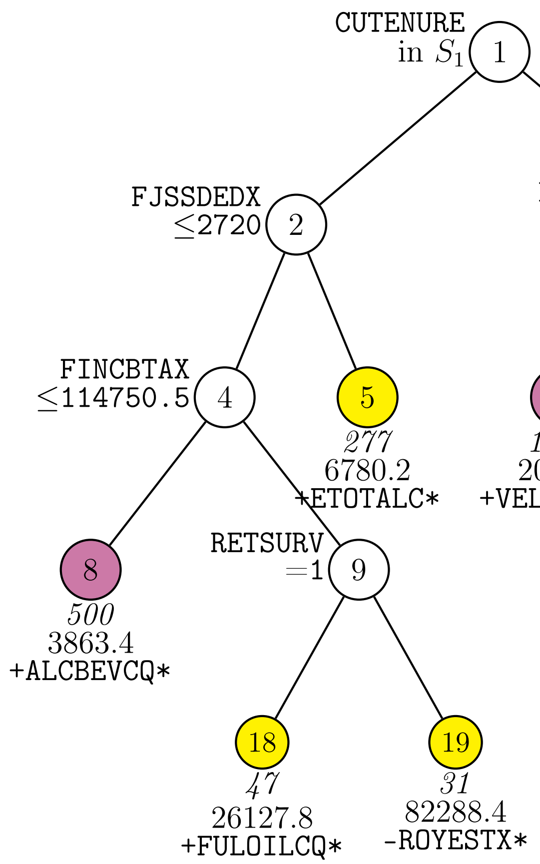

Figure 10: GUIDE v.41.0 0.250-SE piecewise simple linear weighted least-squares regression tree (constant fitted to incomplete cases) for predicting INTRDVX. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. $S_1 = \{2, 6\}$. Sample size (in *italics*), weighted mean of INTRDVX, and signed name of regressor variable printed below nodes. Terminal nodes with means above and below value of 5130.6 at root node are colored yellow and purple respectively. Asterisk appended to regressor name indicates its slope is significant at the 0.05 level (unadjusted for multiplicity and model fitting). Second best split variable at root node is REFGEN.

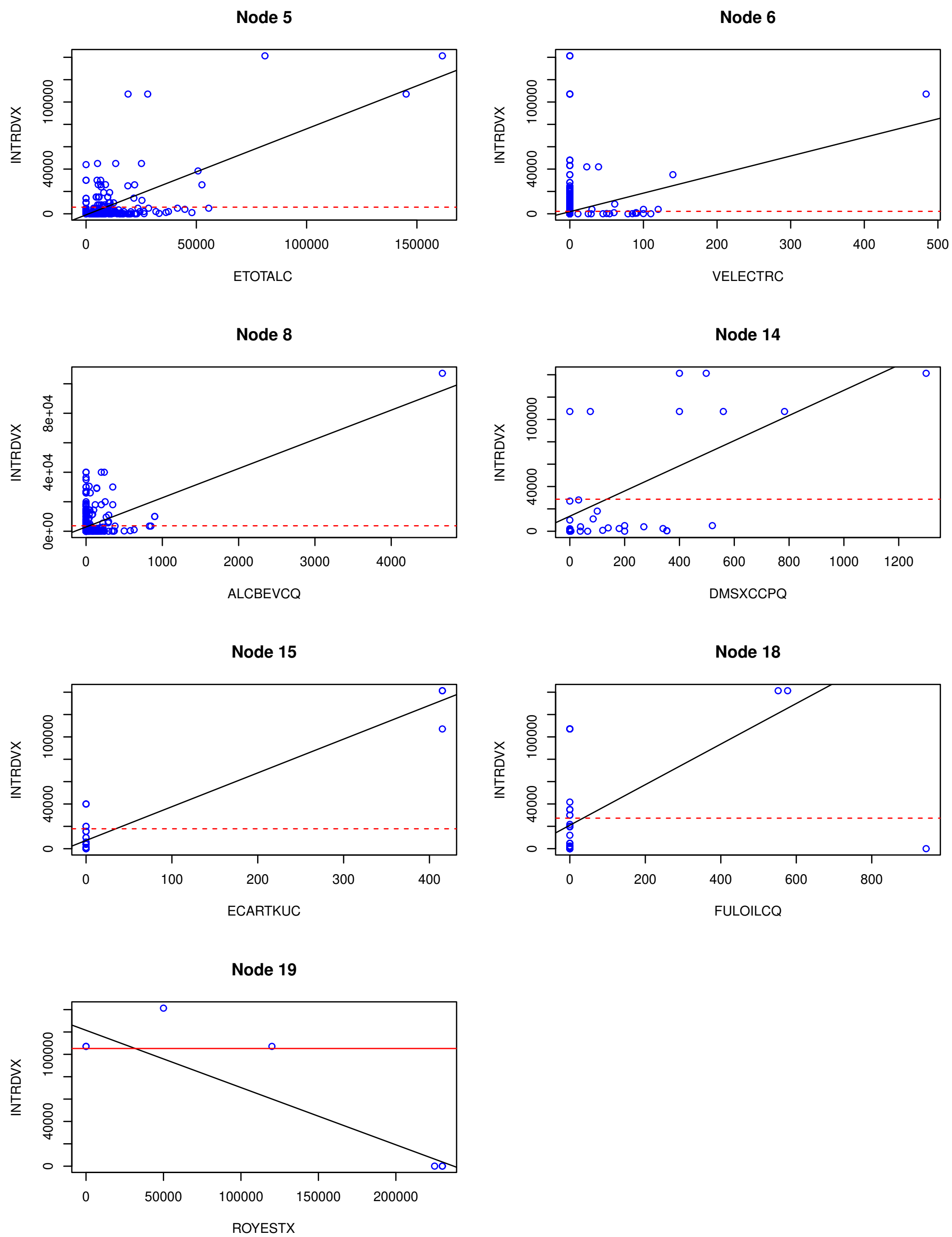

Figure 11: Data and regression lines in terminal nodes of tree in Figure 10. If there are missing values in the regressor, a solid red line marks their d mean. If there are no missing values, a dashed red line marks the d mean of all points in the node.

```
z <- read.table("ce2021.txt",header=TRUE)
par(mfrow=c(4,2))
z1 <- read.table("lin.fit",header=TRUE)
z2 <- read.table("lin.var",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)){
        miss <- z1$node == n & z1$train == "y" & is.na(x)
        abline(h=mean(y[miss]),col="red",lty=1)
    } else {
        abline(h=mean(y[gp]),col="red",lty=2)
    }
    title(paste("Node",n))
}
```

Figure 12: R code for Figure 11

```
14   DMSXCCPQ  0.1346E+5  112.6    1.000  0.1413E+6
15   ECARTKUC  7371.      302.2    1.000  0.1413E+6
```

6.3 Stepwise linear

Besides piecewise constant and best simple polynomial, GUIDE can fit a multiple linear (where all `n` and `f` variables are used as regressors) or a stepwise linear (where forward and backward selection is used to select a subset of regressors) regression model at each node. Quite often, these models have higher prediction accuracy, as hinted by the cross-validation estimates of MSE in the output.

For stepwise regression, missing values in each x variable are imputed with the mean of x in the node and a stepwise linear regression model is fitted to the y variable, using the imputed x variables and their missing-value indicators. The name of the indicator of x denoted by x.`NA`. In other words, $x.\mathtt{NA} = I(x = \mathtt{NA})$.

6.3.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: step.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: step.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended for prediction)
Choose 1 for multiple regression
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise 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: ce2021reg.dsc
Reading data description file ...
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
D variable is INTRDVX
Reading data file ...
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 47 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: N variable DIRACC is constant
Warning: N variable TOTHVHRP is constant
Warning: N variable TOTHVHRC is constant
Warning: N variable ROTHRFLC is constant
Warning: N variable WELFREBX is constant
Warning: N variable OTHLYRBX is constant
Warning: N variable OTHLNYRB is constant
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1      384       84        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Weight variable FINLWT21 in column: 31
Number of cases used for training: 2487
Number of split variables: 431
```

```
Number of cases excluded due to 0 W or missing D variable: 1478
Finished reading data file
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction tests; too many predictor variables
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): step.tex
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>=2):
Input file name: step.reg
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: step.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: step.r
Input rank of top variable to split root node ([1:515], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < step.in
```

6.3.2 Results

```
Least squares regression tree
Predictions truncated at global min. and max. of D sample values
Pruning by cross-validation
Data description file: ce2021reg.dsc
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
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: 3965
Length of longest entry in data file: 11
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Warning: N variable DIRACC is constant
Warning: N variable TOTHVHRP is constant
Warning: N variable TOTHVHRC is constant
Warning: N variable ROTHRFLC is constant
Warning: N variable WELFREBX is constant
```

```
Warning: N variable OTHLYRBX is constant
Warning: N variable OTHLNYRB is constant
Smallest and largest positive weights are  1.0725E+03 and 9.3902E+04

Summary information for training sample of size 2487 (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       n   1.0000E+00    1.000                    125
      2  DIRACC_      m                                  2
      3  AGE_REF      n    19.00        87.00
      4  AGE_REF_     m                                  0
      5  AGE2         n   2.1000E+01    87.00                   1092
      6  AGE2_        m                                  1
      :
    632  WHOLIFX.NA   f    0.000        1.000
    633  VEHICTAX.NA  f    0.000        1.000
    634  CREDYR.NA    f    0.000        1.000

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1      384       84        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Weight variable FINLWT21 in column: 31
Number of cases used for training: 2487
Number of split variables: 431
Number of cases excluded due to 0 W or missing D variable: 1478

Missing regressors imputed with means and missing-value indicators added
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Weighted error estimates used for pruning
Warning: No interaction tests; too many predictor variables
No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 12
Minimum node sample size: 20
```

```
Top-ranked variables and 1-df chi-squared values at root node
     1  0.6230E+02   RETSURV
     2  0.5638E+02   AGE_REF
     3  0.5312E+02   CUTENURE
     4  0.5223E+02   REFGEN
     :
   368  0.1382E-03   FURNTRPQ
   369  0.9471E-04   MAJAPPCQ
   370  0.8111E-04   ECARTKNC

Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       40   4.549E+12   6.212E+11   9.947E+11   3.505E+12   1.637E+12
   2       38   4.549E+12   6.212E+11   9.947E+11   3.505E+12   1.637E+12
   3       29   4.549E+12   6.212E+11   9.947E+11   3.505E+12   1.637E+12
   4**     28   4.549E+12   6.212E+11   9.947E+11   3.505E+12   1.637E+12
   5       27   5.221E+12   7.323E+11   1.103E+12   3.700E+12   2.079E+12
   6       26   5.275E+12   7.329E+11   1.095E+12   3.915E+12   2.068E+12
   7       24   5.218E+12   7.326E+11   1.098E+12   3.869E+12   2.028E+12
   8       23   5.183E+12   7.321E+11   1.099E+12   3.695E+12   2.025E+12
   9       22   5.191E+12   7.321E+11   1.099E+12   3.703E+12   2.024E+12
  10       21   5.185E+12   7.320E+11   1.099E+12   3.677E+12   2.025E+12
  11       17   5.073E+12   7.213E+11   1.123E+12   3.413E+12   2.120E+12
  12       15   5.074E+12   7.213E+11   1.123E+12   3.413E+12   2.119E+12
  13       13   5.078E+12   7.212E+11   1.121E+12   3.413E+12   2.119E+12
  14+      11   4.974E+12   7.145E+11   1.132E+12   2.891E+12   2.156E+12
  15       10   5.051E+12   7.084E+11   1.178E+12   2.891E+12   2.217E+12
  16++      9   4.892E+12   7.016E+11   1.129E+12   3.068E+12   2.162E+12
  17        8   5.043E+12   6.730E+11   1.029E+12   4.422E+12   1.489E+12
  18        6   5.286E+12   7.237E+11   1.134E+12   4.797E+12   2.107E+12
  19        1   7.978E+12   9.638E+11   6.871E+11   8.046E+12   8.631E+11

0-SE tree based on mean is marked with * and has 28 terminal nodes
0-SE tree based on median is marked with + and has 11 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as -- tree
* 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     2487     2487    26  5.131E+03  7.866E+12  0.0603 RETSURV
         2      651      651    24  5.786E+03  6.598E+12  0.0804 OTHREGX
        4T       28       28     4  2.938E+03  5.458E+10  0.9715 -
         5      623      623    24  5.926E+03  6.754E+12  0.0902 SEX2
        10      244      244    20  1.033E+04  4.946E+12  0.6703 FINDRETX
        20      217      217    15  9.797E+03  5.371E+12  0.5974 EOTHLODC
        40      193      193    15  9.425E+03  5.011E+12  0.6497 ALLFULPQ
        80      173      173    11  9.107E+03  5.570E+12  0.5629 ETRANPTP
       160       46       46     8  3.679E+03  8.093E+11  0.5962 ERANKHM
      320T       21       21     2  6.075E+03  1.276E+12  0.6159 -
      321T       25       25     1  1.713E+03  0.000E+00  1.0000 -
       161      127      127     9  1.122E+04  5.109E+12  0.6898 CUTENURE
       322       71       71     5  1.607E+04  9.315E+12  0.5794 RETSURVX
      644T       44       44     9  1.334E+04  2.211E+12  0.8700 HEALTHCQ
      645T       27       27     5  2.023E+04  1.200E+13  0.6888 -
       323       56       56     9  4.610E+03  8.122E+11  0.9030 MRTINTPQ
      646T       36       36     4  3.565E+03  4.629E+11  0.6672 -
      647T       20       20     5  6.682E+03  1.754E+09  0.9999 -
       81T       20       20     2  1.286E+04  2.110E+13  0.2045 -
       41T       24       24     4  1.285E+04  2.842E+12  0.6027 -
       21T       27       27    13  1.449E+04  2.975E+11  0.9936 -
       11T      379      379    41  3.114E+03  4.841E+11  0.8217 UNISTRQ
         3     1836     1836    22  4.915E+03  7.982E+12  0.1003 OTHRINCX
        6T       36       36     3  3.484E+02  2.533E+10  0.1177 -
         7     1800     1800    22  4.971E+03  8.307E+12  0.0817 STATE
        14      160      160    16  2.234E+03  1.719E+11  0.7542 EDUCAPQ
        28      140      140    10  2.335E+03  2.460E+11  0.6703 EMRTPNOP
        56      104      104    13  1.762E+03  8.363E+10  0.8491 OCCUCOD1
      112T       36       36    10  3.930E+03  2.080E+11  0.8734 -
       113       68       68     6  5.372E+02  5.714E+09  0.5147 FOODPQ
      226T       39       39    13  6.323E+02  4.990E+08  0.9750 -
      227T       29       29     4  4.123E+02  4.663E+09  0.3912 -
       57T       36       36    10  4.059E+03  1.084E+11  0.9346 -
       29T       20       20     2  1.580E+03  1.243E+11  0.3503 -
        15     1640     1640    22  5.255E+03  9.015E+12  0.0854 INCNONW1
        30      291      291    31  1.590E+04  8.301E+12  0.7299 FRRETIRM
        60      191      191    23  1.370E+04  5.513E+12  0.7716 BUILT
       120      131      131    13  1.784E+04  4.897E+12  0.8436 PERSCACQ
       240      109      109    12  1.671E+04  2.554E+12  0.9080 ERANKH
       480       81       81    12  1.153E+04  1.421E+12  0.9345 FOODPQ
      960T       43       43    10  8.826E+03  4.548E+10  0.9979 STATE
      961T       38       38     7  1.541E+04  1.748E+12  0.9288 -
```

```
       481T      28       28     7  3.204E+04  5.745E+12  0.8795 -
       241T      22       22     4  2.354E+04  1.333E+13  0.7562 -
       121       60       60    11  5.283E+03  3.681E+11  0.9132 NUM_AUTO
       242T      36       36     5  2.372E+03  2.014E+11  0.7755 -
       243T      24       24     8  9.103E+03  1.956E+10  0.9981 -
        61      100      100     8  2.028E+04  1.873E+12  0.9555 BUILT
       122T      37       37     7  1.479E+04  8.768E+10  0.9977 -
       123       63       63     7  2.343E+04  2.312E+12  0.9520 HEALTHPQ
       246T      27       27     5  7.647E+03  9.147E+10  0.9933 -
       247T      36       36     8  3.517E+04  2.369E+12  0.9690 -
        31     1349     1349    20  3.085E+03  4.887E+12  0.0800 FSALARYX
        62T     434      434    25  3.001E+03  4.335E+09  0.9993 -
        63T     915      915    18  3.124E+03  1.804E+08  1.0000 -
Warning: tree very large, omitting node numbers in LaTeX file

Number of terminal nodes of final tree: 28
Total number of nodes of final tree: 55
Second best split variable (based on curvature test) at root node is AGE_REF

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

Node 1: RETSURV = "1"
  Node 2: OTHREGX <= 11337.000
    Node 4: INTRDVX-mean = 2938.2174
  Node 2: OTHREGX > 11337.000 or NA
    Node 5: SEX2 = "2"
      Node 10: FINDRETX <= 2350.0000
        Node 20: EOTHLODC <= 575.00000
          Node 40: ALLFULPQ <= 41.000000
            Node 80: ETRANPTP <= 444.16650
              Node 160: ERANKHM <= 0.51617950
                Node 320: INTRDVX-mean = 6075.2030
              Node 160: ERANKHM > 0.51617950 or NA
                Node 321: INTRDVX-mean = 1712.6607
            Node 80: ETRANPTP > 444.16650 or NA
              Node 161: CUTENURE = "2"
                Node 322: RETSURVX <= 31600.000
                  Node 644: INTRDVX-mean = 13336.430
                Node 322: RETSURVX > 31600.000 or NA
                  Node 645: INTRDVX-mean = 20229.476
              Node 161: CUTENURE /= "2"
                Node 323: MRTINTPQ <= 970.00000
                  Node 646: INTRDVX-mean = 3564.6625
                Node 323: MRTINTPQ > 970.00000 or NA
                  Node 647: INTRDVX-mean = 6681.5390
```

```
           Node 40: ALLFULPQ > 41.000000 or NA
             Node 81: INTRDVX-mean = 12860.244
         Node 20: EOTHLODC > 575.00000 or NA
           Node 41: INTRDVX-mean = 12851.752
       Node 10: FINDRETX > 2350.0000 or NA
         Node 21: INTRDVX-mean = 14488.623
     Node 5: SEX2 /= "2"
       Node 11: INTRDVX-mean = 3114.4256
 Node 1: RETSURV /= "1"
   Node 3: OTHRINCX <= 4083.0000
     Node 6: INTRDVX-mean = 348.37734
   Node 3: OTHRINCX > 4083.0000 or NA
     Node 7: STATE = "10", "22", "40", "41", "53"
       Node 14: EDUCAPQ <= 162.00000
         Node 28: EMRTPNOP <= 991.50000
           Node 56: OCCUCOD1 = "1"
                   or (OCCUCOD1 = NA & OCCU_OD1 = "A")
             Node 112: INTRDVX-mean = 3929.7658
           Node 56: OCCUCOD1 /= "1"
                   & not (OCCUCOD1 = NA & OCCU_OD1 = "A")
             Node 113: FOODPQ <= 1282.6667
               Node 226: INTRDVX-mean = 632.29761
             Node 113: FOODPQ > 1282.6667 or NA
               Node 227: INTRDVX-mean = 412.30881
         Node 28: EMRTPNOP > 991.50000 or NA
           Node 57: INTRDVX-mean = 4059.2022
       Node 14: EDUCAPQ > 162.00000 or NA
         Node 29: INTRDVX-mean = 1580.4405
     Node 7: STATE /= "10", "22", "40", "41", "53"
       Node 15: INCNONW1 = "1"
         Node 30: FRRETIRM <= 29823.000
           Node 60: BUILT <= 1999.5000
             Node 120: PERSCACQ <= 50.000000
               Node 240: ERANKH <= 0.55780640
                 Node 480: FOODPQ <= 686.66665
                   Node 960: INTRDVX-mean = 8825.6885
                 Node 480: FOODPQ > 686.66665 or NA
                   Node 961: INTRDVX-mean = 15411.105
               Node 240: ERANKH > 0.55780640 or NA
                 Node 481: INTRDVX-mean = 32038.502
             Node 120: PERSCACQ > 50.000000 or NA
               Node 241: INTRDVX-mean = 23538.397
           Node 60: BUILT > 1999.5000 or NA
             Node 121: NUM_AUTO <= 0.50000000
               Node 242: INTRDVX-mean = 2372.2425
             Node 121: NUM_AUTO > 0.50000000 or NA
```

```
              Node 243: INTRDVX-mean = 9103.2088
        Node 30: FRRETIRM > 29823.000 or NA
          Node 61: BUILT <= 1962.5000
            Node 122: INTRDVX-mean = 14794.456
          Node 61: BUILT > 1962.5000 or NA
            Node 123: HEALTHPQ <= 1336.0000
              Node 246: INTRDVX-mean = 7646.6109
            Node 123: HEALTHPQ > 1336.0000 or NA
              Node 247: INTRDVX-mean = 35168.090
      Node 15: INCNONW1 /= "1"
        Node 31: FSALARYX <= 60200.000
          Node 62: INTRDVX-mean = 3000.6016
        Node 31: FSALARYX > 60200.000 or NA
          Node 63: INTRDVX-mean = 3123.6922

 ***************************************************************
 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    -0.3652E+05  -0.8428       0.3994
 FRRETIRX     0.2151        7.734        0.000        0.000      0.1071E+05   0.1116E+06
 FSALARYX     0.7259E-02    1.817       0.6927E-01    0.000      0.8172E+05   0.7645E+06
 FSSIX        0.4632E-01   0.1486       0.8818        0.000       83.02       0.3600E+05
 OTHRINCX     0.3675E-01   0.5329       0.5941        150.0      0.2064E+05   0.1041E+06
 FSMPFRMX     0.4211E-01    4.769       0.1956E-05  -0.1160E+06   6641.       0.7703E+06
 JFS_AMT      0.7798E-01   0.1264       0.8994        0.000       64.43        9600.
 NETRENTX    -0.1567E-01  -0.3371       0.7361      -0.1402E+05  0.1462E+05   0.1589E+06
 NETRNTBX    -0.1043      -0.3341       0.7383       -2400.       9325.       0.7130E+05
 OTHREGBX     0.1112       0.2365       0.8131        488.0       6380.       0.4200E+05
 OTHREGX      0.1041        1.332       0.1829        100.0      0.1281E+05   0.8288E+05
```

```
 RETSURVX     0.7802E-01    3.849       0.1216E-03    134.0        0.2762E+05   0.1739E+06
 RETSRVBX     0.2851        1.384       0.1665        3500.        0.2850E+05   0.6200E+05
 ROYESTBX     0.2867       0.7365       0.4615        200.0         7464.       0.6000E+05
 ROYESTX     -0.3525E-01   -1.330       0.1835        5.000        0.4599E+05   0.2300E+06
 OTHRINCX.NA   1144.       0.5108       0.6095        0.000        0.9715        1.000
 WELFAREX.NA   6850.       0.8444       0.3985        0.000        0.9979        1.000
 WELFREBX.NA   842.0       0.2131E-01   0.9830        0.000        0.9999        1.000
 NETRENTB.NA   7035.       0.9969       0.3189        0.000        0.9972        1.000
 NETRENTX.NA  -2786.       -2.009       0.4461E-01    0.000        0.9211        1.000
 OTHREGBX.NA   3134.       0.6210       0.5347        0.000        0.9932        1.000
 OTHREGX.NA    690.4       0.5405       0.5889        0.000        0.9028        1.000
 RETSURVX.NA   2537.        2.468       0.1364E-01    0.000        0.7632        1.000
 RETSRVBX.NA   3491.       0.6662       0.5053        0.000        0.9949        1.000
 ROYESTBX.NA   8325.        1.103       0.2701        0.000        0.9976        1.000
 ROYESTX.NA   -4820.       -2.536       0.1128E-01    0.000        0.9594        1.000
 INTRDVX mean = 5130.60
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if OTHREGX <= 11337.000
 OTHREGX mean = 12453.689
 ----------------------------
:
:
Node 62: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.6644E+05    19.05        0.000
 FINCBTAX     0.9992        623.2        0.000      -0.1050E+06   0.5582E+05   0.5265E+06
 FRRETIRX    -0.9981       -373.3        0.000        0.000        5260.       0.9000E+05
 FSALARYX    -0.9983       -526.6        0.000        0.000       0.2803E+05   0.6000E+05
 FSSIX       -0.8487       -59.82        0.000        0.000        171.0       0.3600E+05
 OTHRINCX    -0.9928       -44.32        0.000        4166.       0.1168E+05   0.2000E+05
 ENOMOTRC     -64.69       -24.33        0.000        0.000        1.561        372.0
 FSMPFRMX    -0.9991       -535.9        0.000      -0.1160E+06   0.1448E+05   0.3852E+06
 IRABX       -0.1430       -39.57        0.000        5000.       0.7002E+05   0.3250E+06
 IRAYRB      -0.2897E+05   -24.59        0.000        2.000        3.377        5.000
 IRAYRBX      0.3323        32.66        0.000        5000.       0.8364E+05   0.3250E+06
 JFS_AMT      -1.012       -62.19        0.000        0.000        285.2        9600.
 LIQUIDBX    -0.1103       -11.91        0.000        100.0       0.1477E+05   0.7500E+05
 LIQUDYRB      517.6        4.919       0.1261E-05    1.000        3.556        6.000
 NETRENTX    -0.9983       -292.9        0.000      -0.1402E+05   0.1433E+05   0.1047E+06
 OTHREGBX    -0.9985       -6.880        0.000        488.0        1802.        6420.
 OTHREGX     -0.9996       -248.5        0.000        144.0       0.1341E+05   0.8288E+05
 ROYESTX     -0.9992       -367.6        0.000        5.000       0.2532E+05   0.2300E+06
 OTHRINCX.NA  0.1166E+05    84.11        0.000        0.000        0.9745        1.000
```

```
 WELFREBX.NA  0.1633E+05    11.87         0.000         0.000       0.9997        1.000
 IRAB.NA       3207.        8.123         0.000         0.000       0.9873        1.000
 IRAYRB.NA    -4268.       -12.27         0.000         0.000       0.9779        1.000
 NETRENTX.NA  0.1430E+05    175.7         0.000         0.000       0.9194        1.000
 OTHREGX.NA   0.1324E+05    205.2         0.000         0.000       0.8354        1.000
 ROYESTX.NA   0.2530E+05    191.3         0.000         0.000       0.9645        1.000
 INTRDVX mean = 3000.60
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
Node 63: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean        Maximum
 Constant     -6543.       -22.65        0.000
 FINCBTAX     0.9981        4925.        0.000       0.2884E+05   0.1760E+06   0.1039E+07
 FRRETIRX    -0.9982       -1909.        0.000        0.000        1016.       0.6000E+05
 FSALARYX    -0.9981       -4764.        0.000       0.6040E+05   0.1624E+06   0.7645E+06
 FSSIX        -1.009       -318.8        0.000        0.000        43.16       0.2400E+05
 OTHRINCX    -0.9968       -605.5        0.000        5000.       0.1821E+05   0.5640E+05
 FSMPFRMX    -0.9981       -4690.        0.000      -0.1160E+06    6606.       0.7703E+06
 NETRENTX    -0.9978       -2543.        0.000      -0.1402E+05   0.1458E+05   0.1589E+06
 OTHREGB      -1965.       -174.2        0.000        1.000        3.121        7.000
 OTHREGX     -0.9989       -832.9        0.000        400.0        9303.       0.4000E+05
 ROYESTX     -0.9984       -1761.        0.000        10.00       0.2781E+05   0.8700E+05
 OTHRINCX.NA  0.1823E+05    598.6        0.000        0.000       0.9908        1.000
 WELFREBX.NA  0.1069E+05    39.55        0.000        0.000       0.9999        1.000
 NETRENTB.NA  -2389.       -34.05        0.000        0.000       0.9983        1.000
 NETRENTX.NA  0.1458E+05    1271.        0.000        0.000       0.9202        1.000
 OTHREGBX.NA   4255.        144.9        0.000        0.000       0.9904        1.000
 OTHREGX.NA    9329.        839.8        0.000        0.000       0.9203        1.000
 ROYESTX.NA   0.2775E+05    1394.        0.000        0.000       0.9744        1.000
 INTRDVX mean = 3123.69
 Predicted values truncated at 1.00000 & 141304.
 ----------------------------
 Proportion of variance (R-squared) explained by tree model: 0.9375

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

The tree is shown in Figure 13. The contents of `step.reg` below show for each terminal node, the node number, lower and upper truncation values, and the variables selected by stepwise regression in each node.

```
node lower upper variables
```

```
4 1.0000 0.14130E+6 AS_COMP4 FPRIPENX INC_HRS1
320 1.0000 0.14130E+6 FINCBTAX
321 1.0000 0.14130E+6
644 1.0000 0.14130E+6 AGE2 FINCBTAX FJSSDEDX INC_RANK PERSOT64 POPSIZE RENTEQVX VEHQ
:
246 1.0000 0.14130E+6 FINCBTAX FJSSDEDX FRRETIRX FSALARYX
247 1.0000 0.14130E+6 FINCBTAX FRRETIRX INC_HRS2 INC_RANK POPSIZE RENTEQVX TOTEXPPQ
62 1.0000 0.14130E+6 FINCBTAX FRRETIRX FSALARYX FSSIX OTHRINCX ENOMOTRC FSMPFRMX IRABX IRAYRB IRAYRE
63 1.0000 0.14130E+6 FINCBTAX FRRETIRX FSALARYX FSSIX OTHRINCX FSMPFRMX NETRENTX OTHREGB OTHREGX ROY
```

7 Quantile regression: CE data

GUIDE can build piecewise-constant and piecewise-linear quantile regression models. First we show how to build a piecewise-constant 0.90-quantile regression model.

7.1 Piecewise constant: one quantile

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: quantcon.in
 Input 1 for model fitting, 2 for importance or DIF scoring,
       3 for data conversion ([1:3], <cr>=1):
 Name of batch output file: quantcon.out
 Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
 Input 1 for classification, 2 for regression, 3 for propensity score grouping
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
 Input choice ([1:7], <cr>=1): 2
 Choose complexity of model to use at each node:
 Choose 1 for multiple regression (recommended for prediction)
 Choose 2 for best simple polynomial in one N or F variable
 Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
 1: multiple linear, 2: best 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);
```

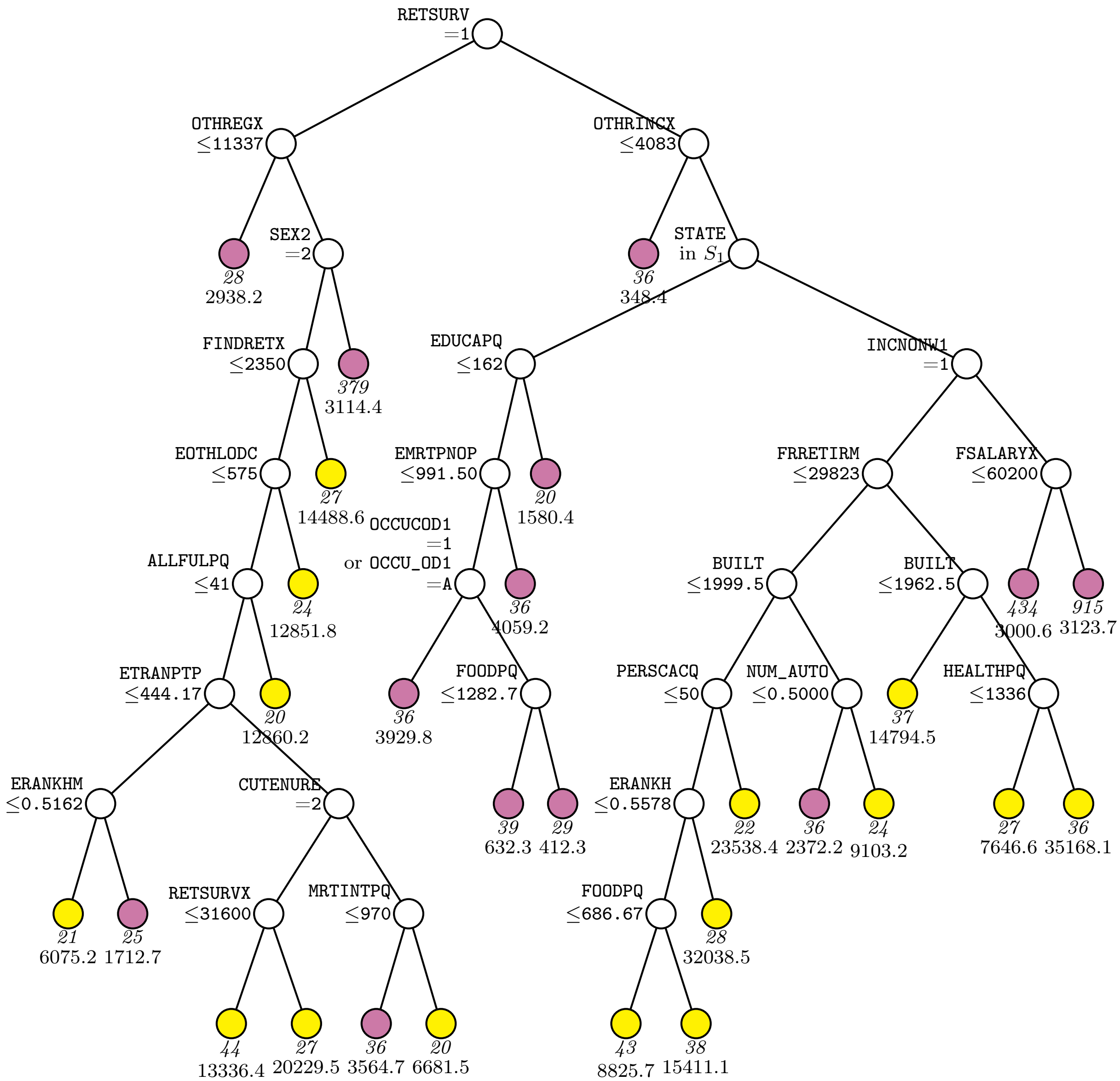

Figure 13: GUIDE v.41.1 0.250-SE piecewise-stepwise linear weighted least-squares regression tree (missing regressor values imputed and missing indicators added) for predicting `INTRDVX`. At each split, an observation goes to the left branch if and only if the condition is satisfied. $S_1 = \{10, 22, 40, 41, 53\}$. Sample size (in *italics*) and weighted mean of `INTRDVX` printed below nodes. Terminal nodes with means above and below value of 5130.6 at root node are colored yellow and purple respectively. Second best split variable at root node is `AGE_REF`.

```
enclose with matching quotes if it has spaces: ce2021reg.dsc
Reading data description file ...
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
384 N variables changed to S
D variable is INTRDVX
Reading data file ...
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 47 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: S variable DIRACC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable ROTHRFLC is constant
Warning: S variable WELFREBX is constant
Warning: S variable OTHLYRBX is constant
Warning: S variable OTHLNYRB is constant
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1        0        0      384
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Number of cases used for training: 2487
Number of split variables: 431
```

```
Number of cases excluded due to 0 W or missing D variable: 1478
Finished reading data file
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction tests; too many predictor variables
Warning: All positive weights treated as 1
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): quantcon.tex
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: quantcon.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: quantcon.r
Input rank of top variable to split root node ([1:431], <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: ce2021reg.dsc
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
384 N variables changed to S
D variable is INTRDVX
Piecewise constant model
Number of records in data file: 3965
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 DIRACC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable ROTHRFLC is constant
Warning: S variable WELFREBX is constant
Warning: S variable OTHLYRBX is constant
Warning: S variable OTHLNYRB is constant
Smallest and largest positive weights are  1.0725E+03 and 9.3902E+04
```

```
Summary information for training sample of size 2487 (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     s    1.000        1.000                   125
      2  DIRACC_    m                                 2
      3  AGE_REF    s    19.00        87.00
      4  AGE_REF_   m                                 0
      5  AGE2       s    21.00        87.00                  1092
      6  AGE2_      m                                 1
      :
    547  WHLFYR     c                                 1      2487
    548  WHLFYR_    m                                 1
    549  FFTAXOWE   s  -0.3368E+05   0.3380E+06
    550  FSTAXOWE   s   -3074.       0.5654E+05

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1        0        0      384
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Number of cases used for training: 2487
Number of split variables: 431
Number of cases excluded due to 0 W or missing D variable: 1478

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Weighted error estimates used for pruning
Warning: No interaction tests; too many predictor variables
Warning: All positive weights treated as 1
No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 12
Minimum node sample size: 24
Top-ranked variables and 1-df chi-squared values at root node
     1  0.6943E+02   CUTENURE
```

```
     2  0.6324E+02   REFGEN
     3  0.5982E+02   RENTEQVX
     4  0.5957E+02   AGE_REF
     5  0.5754E+02   AGE2
     :
   382  0.3478E-03   TOTHFARP
   383  0.4489E-04   MAJAPPPQ

 Size and CV Loss and SE of subtrees:
  Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
    1*       9   6.283E+07   4.914E+06   4.909E+06   6.446E+07   9.486E+06
    2**      8   6.303E+07   4.991E+06   4.648E+06   6.440E+07   9.233E+06
    3++      6   6.427E+07   5.205E+06   4.954E+06   6.477E+07   9.011E+06
    4        5   6.977E+07   5.683E+06   6.221E+06   6.830E+07   1.084E+07
    5        4   7.116E+07   5.848E+06   6.732E+06   7.164E+07   1.353E+07
    6        3   7.477E+07   5.750E+06   5.747E+06   6.854E+07   8.824E+06
    7        1   9.107E+07   7.613E+06   5.017E+06   9.297E+07   5.472E+06

 0-SE tree based on mean is marked with * and has 9 terminal nodes
 0-SE tree based on median is marked with + and has 8 terminal nodes
 Selected-SE tree based on mean using naive SE is marked with **
 Selected-SE tree based on mean using bootstrap SE is marked with --
 Selected-SE tree based on median and bootstrap SE is marked with ++
 ** tree same as + tree
 ** tree same as -- tree

 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     2487     2487     1  9.800E+03  CUTENURE
           2      872      872     1  2.500E+04  INC_RANK
           4T     669      669     1  1.200E+04  LUMP_UMX
           5      203      203     1  1.071E+05  RETPENPQ
          10T      63       63     1  1.413E+05  PROPTXCQ
          11      140      140     1  3.832E+04  EENTMSCC
          22T     115      115     1  2.500E+04  STATE
          23T      25       25     1  1.071E+05  -
           3     1615     1615     1  4.200E+03  FFTAXOWE
           6     1363     1363     1  2.573E+03  REF_RACE
          12T      28       28     1  1.500E+04  -
          13     1335     1335     1  2.400E+03  STOCKYRX
```

```
        26T    1311     1311     1  2.000E+03  RETSURVX
        27T      24       24     1  4.000E+04  -
         7T     252      252     1  2.000E+04  REF_RACE

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 REFGEN

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

Node 1: CUTENURE = "2", "5"
  Node 2: INC_RANK <= 0.81944155
    Node 4: INTRDVX sample quantile = 12000.000
  Node 2: INC_RANK > 0.81944155 or NA
    Node 5: RETPENPQ <= 90.250000
      Node 10: INTRDVX sample quantile = 141304.00
    Node 5: RETPENPQ > 90.250000 or NA
      Node 11: EENTMSCC <= 44.000000
        Node 22: INTRDVX sample quantile = 25000.000
      Node 11: EENTMSCC > 44.000000 or NA
        Node 23: INTRDVX sample quantile = 107121.00
Node 1: CUTENURE /= "2", "5"
  Node 3: FFTAXOWE <= 30387.000
    Node 6: REF_RACE = "3"
      Node 12: INTRDVX sample quantile = 15000.000
    Node 6: REF_RACE /= "3"
      Node 13: STOCKYRX <= 83000.000 or NA
        Node 26: INTRDVX sample quantile = 2000.0000
      Node 13: STOCKYRX > 83000.000
        Node 27: INTRDVX sample quantile = 40000.000
  Node 3: FFTAXOWE > 30387.000 or NA
    Node 7: INTRDVX sample quantile = 20000.000

***************************************************************
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 CUTENURE = "2", "5"
 CUTENURE mode = "1"
 Predicted quantile = 9800.00
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if INC_RANK <= 0.81944155
 INC_RANK mean = 0.59625137
 ----------------------------
Node 4: Terminal node
 Predicted quantile = 12000.0
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if RETPENPQ <= 90.250000
 RETPENPQ mean = 2635.1886
 ----------------------------
:
Node 26: Terminal node
 Predicted quantile = 2000.00
 ----------------------------
Node 27: Terminal node
 Predicted quantile = 40000.0
 ----------------------------
Node 7: Terminal node
 Predicted quantile = 20000.0
 ----------------------------
 Observed and fitted values are stored in quantcon.fit
 LaTeX code for tree is in quantcon.tex
 R code is stored in quantcon.r
```

Figure 14 shows the quantile regression tree. The sample size (in *italics*) and 0.90-quantile are given beneath each terminal node.

7.2 Best simple linear

We demonstrate this with a linear 0.90-quantile regression tree.

7.2.1 Input file creation

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

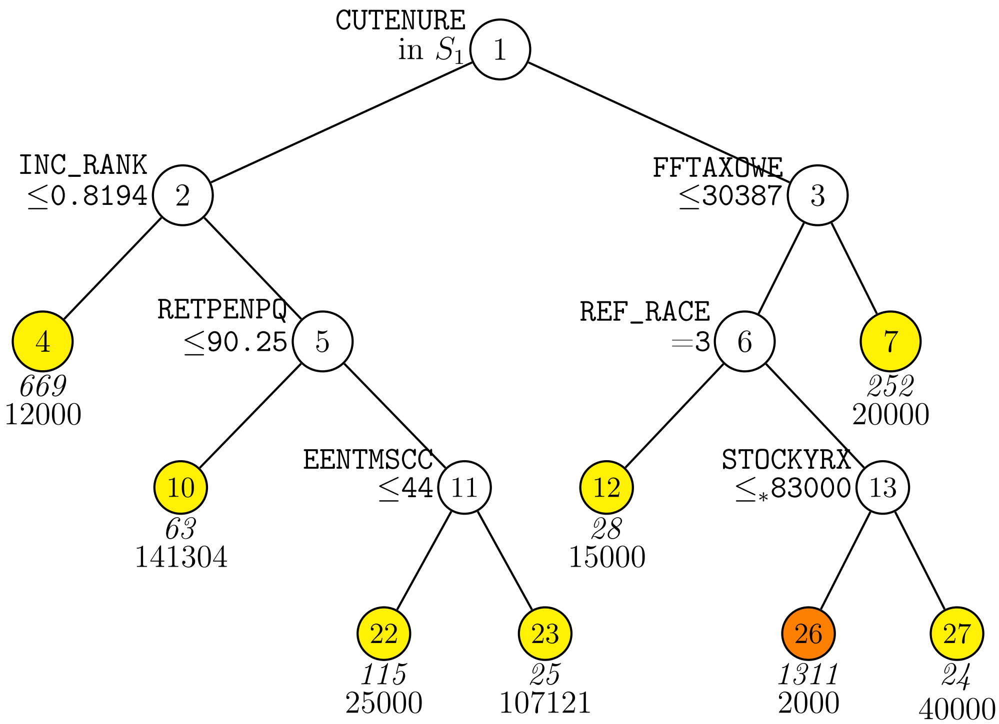

Figure 14: GUIDE v.41.1 0.250-SE piecewise-constant 0.900-quantile regression tree for predicting `INTRDVX`. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. $S_1 = \{2, 5\}$. Sample size (in *italics*) and 0.900-quantile of `INTRDVX` printed below nodes. Terminal nodes with quantiles above and below value of 9800 at root node are colored yellow and orange respectively. Second best split variable at root node is `REFGEN`.

```
Name of batch input file: quantlin.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: quantlin.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best 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: ce2021reg.dsc
Reading data description file ...
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
D variable is INTRDVX
Reading data file ...
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 47 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
```

```
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: N variable DIRACC is constant
Warning: N variable TOTHVHRP is constant
Warning: N variable TOTHVHRC is constant
Warning: N variable ROTHRFLC is constant
Warning: N variable WELFREBX is constant
Warning: N variable OTHLYRBX is constant
Warning: N variable OTHLNYRB is constant
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1      384        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Number of cases used for training: 2487
Number of split variables: 431
Number of cases excluded due to 0 W or missing D variable: 1478
Finished reading data file
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction tests; too many predictor variables
Warning: All positive weights treated as 1
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): quantlin.tex
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>=2):
Input file name: quantlin.reg
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: quantlin.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: quantlin.r
Input rank of top variable to split root node ([1:431], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < quantlin.in
```

Contents of quantlin1.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
Constant model fitted to incomplete cases in each node
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Warning: N variable OTHRINCB is constant
Warning: N variable NETRENTB is constant
Warning: N variable NETRNTBX is constant
Warning: N variable OTHLONBX is constant
Warning: N variable OTHLONB is constant
Smallest and largest positive weights are  1.3507E+03 and 7.0269E+04

Summary information for training sample of size 2922 (excluding observations with
non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
Levels of M variables are for missing values in associated variables
                                                #Codes/
                                                Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  DIRACC     c                                 2      116
      2  DIRACC_    m                                 1
      3  AGE_REF    n    18.00        87.00
      4  AGE_REF_   m                                 0
      5  AGE2       n   2.2000E+01    87.00                 1225
      6  AGE2_      m                                 1
      :
     50  FINLWT21   w    1351.       0.7027E+05
      :
    513  INTRDVX    d    1.000       0.9834E+05
      :
    651  FSTAXOWE   n   -2505.       0.5991E+05
    652  FSTA_OWE   m                                 0
    653  ETOTA      n    1199.       0.2782E+06
```

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

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: .2500

Weighted error estimates used for pruning
Warning: No interaction tests; too many predictor variables
Warning: All positive weights treated as 1
No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 12
Minimum node sample size: 30
Top-ranked variables and chi-squared values at root node
     1  0.1258E+03   STATE
     2  0.1148E+03   STOCKX
     3  0.1010E+03   STOCKYRX
     :
   386  0.7315E-03   MISCPQ
   387  0.7315E-03   MISC1PQ
   388  0.2416E-04   TOTHENTP
   389  0.1391E-07   TEXTILPQ

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       71   4.341E+07   3.751E+06   5.517E+06   3.688E+07   7.638E+06
   2       70   4.340E+07   3.751E+06   5.512E+06   3.688E+07   7.635E+06
   :
  44       14   4.435E+07   3.782E+06   5.659E+06   3.790E+07   7.155E+06
  45**     13   4.179E+07   3.452E+06   5.504E+06   3.604E+07   5.284E+06
  46--     11   4.296E+07   3.501E+06   5.728E+06   3.482E+07   7.137E+06
  47++     10   4.384E+07   3.522E+06   5.917E+06   3.482E+07   7.626E+06
  48        9   4.451E+07   3.544E+06   5.656E+06   3.956E+07   7.243E+06
  49        8   4.872E+07   3.801E+06   5.118E+06   4.717E+07   6.062E+06
  50        7   4.776E+07   3.582E+06   5.193E+06   4.391E+07   7.032E+06
  51        4   4.931E+07   3.502E+06   3.598E+06   4.980E+07   4.804E+06
  52        2   5.750E+07   4.270E+06   3.009E+06   5.606E+07   4.668E+06
```

```
  53       1   6.919E+07   5.240E+06   2.929E+06   6.737E+07   2.150E+06

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 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

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  STATE
          2T     250      250     2  9.834E+04  PERSOT64
          3     2672     2672     2  6.120E+03  STOCKX
          6     2540     2540     2  5.000E+03  FINCATAX
         12     1658     1658     2  3.400E+03  PERSOT64
         24T     936      936     2  1.200E+03  AGE_REF
         25      722      722     2  8.000E+03  TOTXEST
         50T     349      349     2  3.600E+03  SLOCTAXX
         51      373      373     2  1.200E+04  PSU
        102T      36       36     2  2.100E+04  -
        103T     337      337     2  9.500E+03  BUILDING
         13      882      882     2  1.000E+04  INCLASS2
         26       89       89     2  9.834E+04  FSALARYX
         52T      34       34     2  9.834E+04  -
         53T      55       55     2  1.000E+03  -
         27      793      793     2  9.000E+03  CUTENURE
         54      226      226     2  2.206E+04  FEDTAXX
        108      194      194     2  1.500E+04  FJSSDEDX
        216T      46       46     2  2.370E+04  -
        217T     148      148     2  9.000E+03  TOTEXPPQ
        109T      32       32     2  9.834E+04  -
         55T     567      567     2  2.000E+03  FEDTAXX
          7      132      132     2  9.834E+04  STOCKX
         14T     102      102     2  2.000E+04  FEDTAXX
         15T      30       30     2  9.834E+04  -

Number of terminal nodes of final tree: 13
Total number of nodes of final tree: 25
```

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

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

Node 1: STATE = "8", "11", "23", "32", "34", "45", "53", "54"
  Node 2: INTRDVX sample quantile = 98338.000
Node 1: STATE /= "8", "11", "23", "32", "34", "45", "53", "54"
  Node 3: STOCKX <= 86500.000  or STOCKX = NA & STOCKX_ = "A"
    Node 6: FINCATAX <= 98231.000
      Node 12: PERSOT64 <= .50000000
        Node 24: INTRDVX sample quantile = 1200.0000
      Node 12: PERSOT64 > .50000000 or NA
        Node 25: TOTXEST <= 82.500000
          Node 50: INTRDVX sample quantile = 3600.0000
        Node 25: TOTXEST > 82.500000 or NA
          Node 51: PSU = "1110", "1207", "1210", "1316", "1318", "1319", "1320"
            Node 102: INTRDVX sample quantile = 21000.000
          Node 51: PSU /= "1110", "1207", "1210", "1316", "1318", "1319", "1320"
            Node 103: INTRDVX sample quantile = 9500.0000
    Node 6: FINCATAX > 98231.000 or NA
      Node 13: INCLASS2 <= 5.5000000
        Node 26: FSALARYX <= 41500.000
          Node 52: INTRDVX sample quantile = 98338.000
        Node 26: FSALARYX > 41500.000 or NA
          Node 53: INTRDVX sample quantile = 1000.0000
      Node 13: INCLASS2 > 5.5000000 or NA
        Node 27: CUTENURE = "2"
          Node 54: FEDTAXX <= 3637.0000 or NA
            Node 108: FJSSDEDX <= 6440.0000
              Node 216: INTRDVX sample quantile = 23700.000
            Node 108: FJSSDEDX > 6440.0000 or NA
              Node 217: INTRDVX sample quantile = 9000.0000
          Node 54: FEDTAXX > 3637.0000
            Node 109: INTRDVX sample quantile = 98338.000
        Node 27: CUTENURE /= "2"
          Node 55: INTRDVX sample quantile = 2000.0000
  Node 3: not (STOCKX <= 86500.000 or STOCKX = NA & STOCKX_ = "A")
    Node 7: STOCKX <= 478846.50  or STOCKX = NA & STOCKX_ = "C"
      Node 14: INTRDVX sample quantile = 20000.000
    Node 7: not (STOCKX <= 478846.50 or STOCKX = NA & STOCKX_ = "C")
      Node 15: 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 STATE = "8", "11", "23", "32", "34", "45", "53", "54"
 STATE mode = "NA"
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     -6539.
 AGE_REF       297.7        18.00        55.40        87.00
 If regressors have missing values, predicted quantile = 9500.00
 ----------------------------
Node 2: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant      1114.
 FINCATAX     0.3297      -0.1374E+05   0.1119E+06   0.8418E+06
 If regressors have missing values, predicted quantile = 98338.0
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if STOCKX <= 86500.000 or STOCKX_ = "A"
 STOCKX mean = 404023.36
 ----------------------------
 :
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if STOCKX <= 478846.50 or STOCKX_ = "C"
 STOCKX mean = 1195543.9
 ----------------------------
Node 14: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     0.2000E+05
 TEXTILCQ      432.8        0.000        9.776        517.0
 If regressors have missing values, predicted quantile = 20000.0
 ----------------------------
```

```
Node 15: Terminal node
 Coefficients of quantile regression function:
 Regressor    Coefficient Minimum       Mean        Maximum
 Constant     0.9834E+05
 CHILDAGE    -0.1211E+05    0.000        1.034        7.000
 If regressors have missing values, predicted quantile = 98338.0
 ----------------------------
 Observed and fitted values are stored in quantlin1.fit
 Regressor names and coefficients are stored in quantlin1.reg
 LaTeX code for tree is in quantlin1.tex
 R code is stored in quantlin1.r
```

Figure 15 shows the 0.90-quantile regression tree.

7.3 Two quantiles: checking variance heterogeneity

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

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: twoquant.in
 Input 1 for model fitting, 2 for importance or DIF scoring,
       3 for data conversion ([1:3], <cr>=1):
 Name of batch output file: twoquant.out
 Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
 Input 1 for classification, 2 for regression, 3 for propensity score grouping
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
 Input choice ([1:7], <cr>=1): 2
 Choose complexity of model to use at each node:
 Choose 1 for multiple regression (recommended for prediction)
 Choose 2 for best simple polynomial in one N or F variable
 Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
 1: multiple linear, 2: best simple polynomial, 3: constant ([1:3], <cr>=3):
 Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
```

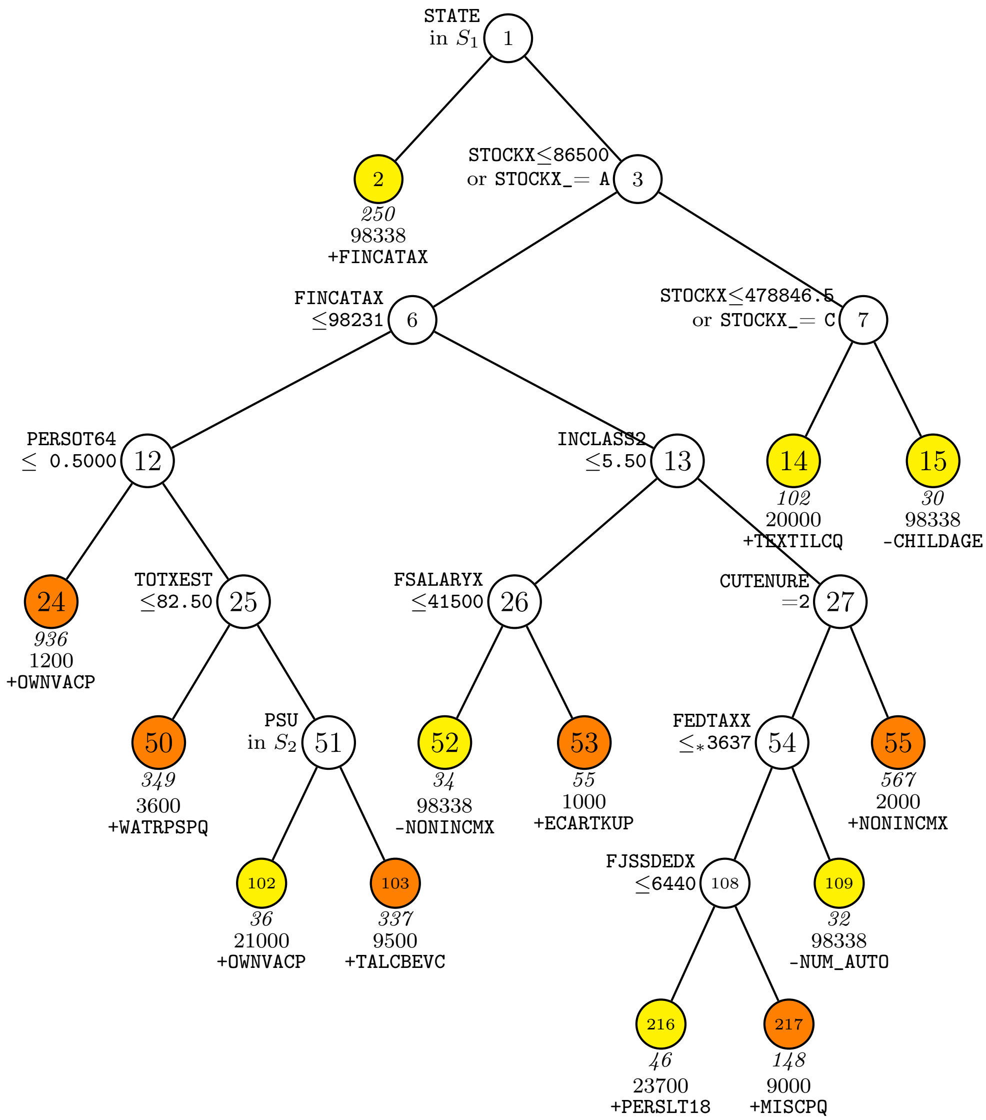

Figure 15: GUIDE v.40.2 0.250-SE piecewise simple linear 0.900-quantile regression tree (constant fitted to incomplete cases) for predicting `INTRDVX`. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '≤ or missing'. S_1 = {`8`, `11`, `23`, `32`, `34`, `45`, `53`, `54`}. S_2 = {`1110`, `1207`, `1210`, `1316`, `1318`, `1319`, `1320`}. Sample size (in *italics*), 0.900-quantile of `INTRDVX`, and sign and name of best regressor printed below nodes. Terminal nodes with quantiles above and below value of 9500 at root node are colored yellow and orange respectively. Second best split variable at root node is `STOCKX`.

```
Input 1 for 1 quantile, 2 for 2 quantiles ([1:2], <cr>=1): 2
Input 1st quantile probability ([0.00:1.00], <cr>=0.25):
Input 2nd quantile probability ([0.00:1.00], <cr>=0.75):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: ce2021reg.dsc
Reading data description file ...
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
384 N variables changed to S
D variable is INTRDVX
Reading data file ...
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 47 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: S variable DIRACC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable ROTHRFLC is constant
Warning: S variable WELFREBX is constant
Warning: S variable OTHLYRBX is constant
Warning: S variable OTHLNYRB is constant
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1        0        0      384
```

```
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Number of cases used for training: 2487
Number of split variables: 431
Number of cases excluded due to 0 W or missing D variable: 1478
Finished reading data file
Input 1 for unweighted, 2 for weighted error estimates during pruning ([1:2], <cr>=2):
Warning: No interaction tests; too many predictor variables
Warning: All positive weights treated as 1
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): twoquant.tex
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: twoquant.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: twoquant.r
Input rank of top variable to split root node ([1:431], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < twoquant.in
```

7.3.2 Output file

```
Dual-quantile regression tree with 0.2500 and 0.7500 quantiles
Pruning by cross-validation
Data description file: ce2021reg.dsc
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
384 N variables changed to S
D variable is INTRDVX
Piecewise constant model
Number of records in data file: 3965
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 DIRACC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable ROTHRFLC is constant
```

```
Warning: S variable WELFREBX is constant
Warning: S variable OTHLYRBX is constant
Warning: S variable OTHLNYRB is constant
Smallest and largest positive weights are  1.0725E+03 and 9.3902E+04

Summary information for training sample of size 2487 (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     s    1.000        1.000                  125
      2  DIRACC_    m                                2
      3  AGE_REF    s    19.00        87.00
      4  AGE_REF_   m                                0
      5  AGE2       s    21.00        87.00                 1092
      6  AGE2_      m                                1
      :
    549  FFTAXOWE   s  -0.3368E+05   0.3380E+06
    550  FSTAXOWE   s   -3074.       0.5654E+05

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965       1478       3965        1        0        0      384
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Number of cases used for training: 2487
Number of split variables: 431
Number of cases excluded due to 0 W or missing D variable: 1478

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Weighted error estimates used for pruning
Warning: No interaction tests; too many predictor variables
Warning: All positive weights treated as 1
No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 12
Minimum node sample size: 24
```

```
Top-ranked variables and 1-df chi-squared values at root node
     1  0.1744E+03   STATE
     2  0.1192E+03   FINCBTAX
     3  0.1135E+03   INC_RANK
     4  0.9547E+02   OCCUCOD1
     :
   382  0.4974E-03   INC_HRS2
   383  0.4468E-03   OTHLNYRX

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       76   1.016E+08   7.031E+06   5.504E+06   1.017E+08   6.605E+06
   2       75   1.016E+08   7.031E+06   5.504E+06   1.017E+08   6.605E+06
   :
  42       15   1.005E+08   7.073E+06   5.612E+06   9.975E+07   6.599E+06
  43       13   1.005E+08   7.144E+06   5.667E+06   9.975E+07   6.463E+06
  44*       7   1.003E+08   7.152E+06   5.731E+06   9.967E+07   6.602E+06
  45++      6   1.004E+08   7.153E+06   5.709E+06   9.967E+07   6.393E+06
  46**      5   1.010E+08   7.361E+06   5.788E+06   1.020E+08   6.408E+06
  47        1   1.164E+08   9.292E+06   5.964E+06   1.190E+08   7.350E+06

0-SE tree based on mean is marked with * and has 7 terminal nodes
0-SE tree based on median is marked with + and has 6 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as -- tree
+ tree same as ++ tree

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

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

Cases fit give the number of cases used to fit node
Column labeled 'Split variable' gives median if node is terminal
       Node    Total    Cases Matrix    Node    Split          Other
       label   cases      fit  rank     median  variable       variables
          1     2487     2487     1  1.500E+01  STATE
         2T      420      420     1  5.000E+00  2.500E+02  FFTAXOWE
          3     2067     2067     1  2.000E+01  INC_RANK
         6T     1560     1560     1  1.500E+01  1.500E+03  CUTENURE
          7      507      507     1  1.200E+02  INCNONW2
         14      468      468     1  1.000E+02  INCNONW2
         28T      56       56     1  2.191E+03  1.071E+05  OWNDWECQ
         29T     412      412     1  1.000E+02  3.000E+03  STATE
         15T      39       39     1  2.000E+02  2.000E+04  -
```

```
 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 FINCBTAX

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

 Node 1: STATE = "19", "24", "31", "40", "45", "49", "51", "54"
   Node 2: INTRDVX sample quantiles = 5.0000000, 250.00000
 Node 1: STATE /= "19", "24", "31", "40", "45", "49", "51", "54"
   Node 3: INC_RANK <= 0.88093190
     Node 6: INTRDVX sample quantiles = 15.000000, 1500.0000
   Node 3: INC_RANK > 0.88093190 or NA
     Node 7: INCNONW2 = "1"
             or (INCNONW2 = NA & INCN_NW2 = "A")
       Node 14: INCNONW2 = "1"
         Node 28: INTRDVX sample quantiles = 2191.0000, 107121.00
       Node 14: INCNONW2 /= "1"
         Node 29: INTRDVX sample quantiles = 100.00000, 3000.0000
     Node 7: INCNONW2 /= "1"
             & not (INCNONW2 = NA & INCN_NW2 = "A")
       Node 15: INTRDVX sample quantiles = 200.00000, 20000.000

 ***************************************************************
 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 STATE = "19", "24", "31", "40", "45", "49", "51", "54"
 STATE mode = "6"
 Sample 0.250-quantile, 0.750-quantile, and median:
     1.5000E+01     1.9300E+03     1.7400E+02
 ----------------------------
Node 2: Terminal node
 Sample 0.250-quantile, 0.750-quantile, and median:
```

```
    5.0000E+00     2.5000E+02     2.0000E+01
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if INC_RANK <= 0.88093190
 INC_RANK mean = 0.65419245
 ----------------------------
Node 6: Terminal node
 Sample 0.250-quantile, 0.750-quantile, and median:
    1.5000E+01     1.5000E+03     1.5000E+02
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if INCNONW2 = "1"
   or INCNONW2 = NA & INCN_NW2 =  "A"
 INCN_NW2 mode = "A"
 ----------------------------
Node 14: Intermediate node
 A case goes into Node 28 if INCNONW2 = "1"
 INCN_NW2 mode = "A"
 ----------------------------
Node 28: Terminal node
 Sample 0.250-quantile, 0.750-quantile, and median:
    2.1910E+03     1.0712E+05     2.5879E+04
 ----------------------------
Node 29: Terminal node
 Sample 0.250-quantile, 0.750-quantile, and median:
    1.0000E+02     3.0000E+03     5.0000E+02
 ----------------------------
Node 15: Terminal node
 Sample 0.250-quantile, 0.750-quantile, and median:
    2.0000E+02     2.0000E+04     6.5000E+03
 ----------------------------
 Observed and fitted values are stored in twoquant.fit
 LaTeX code for tree is in twoquant.tex
 R code is stored in twoquant.r
```

Figure 16 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. The large between-node variations in the inter-quartile ranges in the nodes indicates substantial variance heterogeneity.

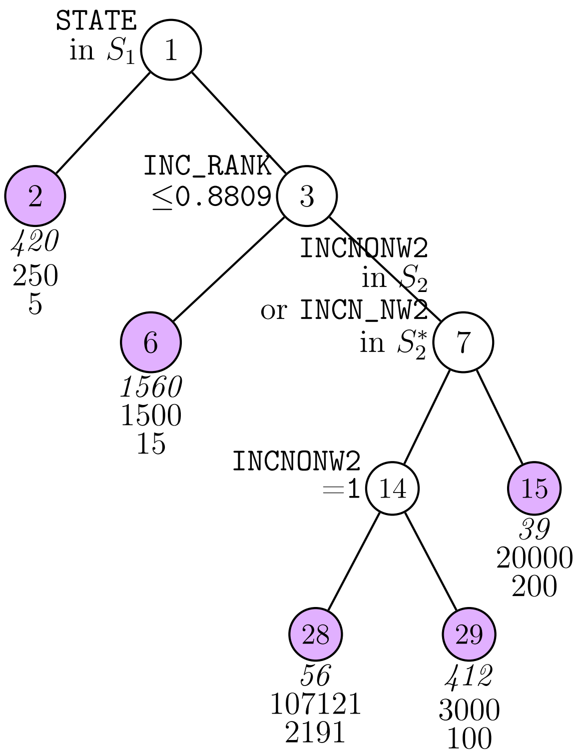

Figure 16: GUIDE v.41.1 0.250-SE piecewise-constant 0.250 and 0.750-quantile regression tree for predicting `INTRDVX`. At each split, an observation goes to the left branch if and only if the condition is satisfied. $S_1 = \{$19, 24, 31, 40, 45, 49, 51, 54$\}$. $S_2 = \{$1$\}$; $S_2^* = \{$A$\}$. Sample size (in *italics*) and sample 0.750 and 0.250-quantiles of `INTRDVX` printed below nodes. Second best split variable at root node is `FINCBTAX`.

8 Periodic variables: NHTSA data

Periodic variables that have a cyclic property, such as angular measurements, hour of day, day of week, and month of year, can be designated as `P` variables in the description file. There can be multiple `P` variables in the same data set. Unlike the other types of variables, each line in the description file containing a `P` variable must have the value of its period (e.g., 360 for angular measurements, 24 for hour of day, 7 for day of week, and 12 for month of year) immediately after `P` on the same line. GUIDE does not allow `P` variables to have missing-value flag (`M`) variables.

The National Highway Traffic Safety Administration (NHTSA) has been conducting vehicle crash tests since 1972. Data from 3310 crash tests are in the file `nhtsadatam.txt` (see www-nrd.nhtsa.dot.gov/database/veh/ for more information). Variable `HIC` (head injury criterion) is a measure of severity of head injury. Experts believe that HIC > 999 is life threatening. Table 9 gives the definitions of the variables appearing in the models below. Besides missing values, there are many variables with illogical values (such as negative values for diameter). To identify these values, we adopt the strategy in the CE data of creating a missing-value flag variable for each variable having illogical values, with the flags being `A`, `B`, and `D` for validly missing, illogical, and valid response, respectively. The data also contain some angular variables (with periods of 360 degrees and for which 0 degrees indicates straight-ahead or head-on) that are defined as `P` in the description file `nhtsadsc.txt` below.

```
nhtsadatam.txt
NA
2
1 BARRIG c
2 BARSHP b
3 BARANG p 360
4 BARDIA n
5 OCCWT n
6 OCCWT_ m
7 DUMSIZ c
8 HH n
9 HH_ m
10 HW n
11 HW_ m
12 HR n
13 HR_ m
14 HS n
15 HS_ m
16 CD n
```

```
17 CD_ m
18 CS n
19 CS_ m
20 AD n
21 AD_ m
22 HD n
23 HD_ m
24 KD n
25 KD_ m
26 HB n
27 HB_ m
28 NB n
29 NB_ m
30 CB n
31 CB_ m
32 KB n
33 SEPOSN c
34 HIC d
35 TKSURF c
36 TKCOND c
37 CLSSPD n
38 CLSSPD_ m
39 IMPANG p 360
40 OFFSET n
41 IMPPNT n
42 MAKED c
43 MODELD c
44 YEAR n
45 BODY c
46 ENGINE c
47 ENGDSP n
48 ENGDSP_ m
49 TRANSM c
50 VEHTWT n
51 VEHTWT_ m
52 CURBWT n
53 WHLBAS n
54 WHLBAS_ m
55 VEHLEN n
56 VEHLEN_ m
57 VEHWID n
58 VEHWID_ m
59 VEHCG n
60 VEHCG_ m
61 COLMEC c
62 BX1 n
```

```
63 BX1_ m
64 BX2 n
65 BX2_ m
66 BX3 n
67 BX3_ m
68 BX4 n
69 BX4_ m
70 BX5 n
71 BX5_ m
72 BX6 n
73 BX6_ m
74 BX7 n
75 BX7_ m
76 BX8 n
77 BX8_ m
78 BX9 n
79 BX9_ m
80 BX10 n
81 BX10_ m
82 BX11 n
83 BX11_ m
84 BX12 n
85 BX12_ m
86 BX13 n
87 BX13_ m
88 BX14 n
89 BX14_ m
90 BX15 n
91 BX15_ m
92 BX16 n
93 BX16_ m
94 BX17 n
95 BX17_ m
96 BX18 n
97 BX18_ m
98 BX19 n
99 BX19_ m
100 BX20 n
101 BX20_ m
102 BX21 n
103 BX21_ m
104 VEHSPD n
105 VEHSPD_ m
106 CRBANG p 360
107 PDOF p 360
108 CARANG p 360
```

Table 9: Some variable definitions for NHTSA data

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

```
109 VEHOR p 360
110 RSTFRT c
111 HIC2 x
112 estHIC2 x
```

We show the results of fitting a piecewise-linear regression tree here.

8.1 Input file creation

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

```
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended for prediction)
Choose 1 for multiple regression
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise 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: nhtsadsc.txt
Reading data description file ...
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
Warning: B variables changed to C
D variable is HIC
Reading data file ...
Number of records in data file: 3310
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 13 categorical variables
Finished assigning codes to 10 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310         34       3310        2       48        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         6       42        0       13        0
No weight variable in data file
Number of cases used for training: 3276
```

```
Number of split variables: 61
Number of cases excluded due to 0 W or missing D variable: 34
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.tex
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>=2):
Input file name: lin.reg
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: lin.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: lin.r
Input rank of top variable to split root node ([1:67], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < lin.in
```

8.2 Results

```
Least squares regression tree
Predictions truncated at global min. and max. of D sample values
Pruning by cross-validation
Data description file: nhtsadsc.txt
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
Warning: B variables changed to C
D variable is HIC
Piecewise simple linear or constant model
Powers are dropped if they are not significant at level 0.0500
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

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
Levels of M variables are for missing values in associated variables
                                                #Codes/
                                                Levels/
 Column  Name           Minimum      Maximum    Periods   #Missing
      1  BARRIG     c                                 3
      2  BARSHP     c                                21
      3  BARANG     p    0.000       330.0          360        14
      4  BARDIA     n   1.9100E+02   1000.                   2807
      5  OCCWT      n   7.2000E+01   83.00                   3265
      6  OCCWT_     m                                 2
      :
     33  SEPOSN     c                                 5        81
     34  HIC        d    0.000      0.1225E+05
     35  TKSURF     c                                 5        80
      :
    106  CRBANG     p    0.000       315.0          360        24
    107  PDOF       p    0.000       345.0          360        23
    108  CARANG     p    0.000       99.00          360       991
    109  VEHOR      p    0.000       90.00          360       995
    110  RSTFRT     c                                 3

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310        34      3310        2       48        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         6       42        0       13        0
No weight variable in data file
Number of cases used for training: 3276
Number of split variables: 61
Number of cases excluded due to 0 W or missing D variable: 34

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 13
Minimum node sample size: 33
Top-ranked variables and 1-df chi-squared values at root node
     1  0.2525E+03   BARSHP
     2  0.1423E+03   IMPANG
     3  0.1248E+03   BARDIA
     4  0.1245E+03   BODY
```

```
     5  0.1204E+03   CRBANG
     :
    66  0.1448E+01   DUMSIZ
    67  0.6191E-02   CARANG

 Size and CV MSE and SE of subtrees:
  Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
    1       60   3.748E+05   6.535E+04   7.046E+04   3.413E+05   5.078E+04
    2       58   3.748E+05   6.535E+04   7.046E+04   3.413E+05   5.078E+04
    3       56   3.748E+05   6.535E+04   7.046E+04   3.413E+05   5.078E+04
    :
   23+      28   3.640E+05   6.529E+04   6.996E+04   3.308E+05   4.855E+04
   24       21   3.676E+05   6.605E+04   7.105E+04   3.425E+05   5.089E+04
   25       20   3.673E+05   6.605E+04   7.105E+04   3.425E+05   5.067E+04
   26       19   3.711E+05   6.644E+04   7.118E+04   3.398E+05   5.463E+04
   27       17   3.686E+05   6.626E+04   7.117E+04   3.398E+05   5.222E+04
   28++     16   3.691E+05   6.632E+04   7.125E+04   3.428E+05   5.255E+04
   29       15   3.640E+05   6.635E+04   6.688E+04   3.572E+05   5.342E+04
   30*      13   3.559E+05   6.457E+04   6.599E+04   3.572E+05   5.196E+04
   31**     10   3.581E+05   6.470E+04   6.618E+04   3.572E+05   5.283E+04
   32        9   3.772E+05   6.825E+04   6.516E+04   3.608E+05   4.732E+04
   33        6   3.786E+05   6.882E+04   6.660E+04   3.608E+05   4.744E+04
   34        1   3.750E+05   6.843E+04   5.388E+04   4.032E+05   4.736E+04

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

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

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

 D-mean is mean of HIC 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     3276     3272     2  5.127E+02  3.743E+05  0.1007 BARSHP  -YEAR
          2T       40       32     2  1.606E+02  1.337E+04  0.3122 -  +HW
           3     3236     3232     2  5.170E+02  3.749E+05  0.1066 BARSHP  -YEAR
          6T       45       45     1  2.603E+02  5.101E+04  0.0000 -  *Constant*
           7     3191     3188     2  5.206E+02  3.757E+05  0.1137 BARSHP  -YEAR
          14      309      204     2  3.486E+02  4.873E+05  0.4317 BX12  -IMPPNT
```

```
        28T      44        7     2  7.805E+02  6.238E+05  0.7198 -  -BARDIA
        29T     265      173     2  2.769E+02  2.827E+05  0.5358 SEPOSN  -IMPPNT
        15     2882     2879     2  5.391E+02  3.315E+05  0.1147 CLSSPD  -YEAR
        30     1292        9     2  4.440E+02  5.638E+05  0.0267 HS  -CB
        60      593      593     1  5.405E+02  5.628E+05  0.0000 RSTFRT  *Constant*
       120T     334       82     2  4.267E+02  4.667E+05  0.1066 YEAR  -IMPPNT
       121T     259      259     1  6.873E+02  5.808E+05  0.0000 VEHCG  *Constant*
        61      699      199     2  3.621E+02  5.644E+05  0.0258 CURBWT  +CURBWT
       122      591      591     1  3.321E+02  3.861E+05  0.0000 IMPANG  *Constant*
       244T     342      341     2  2.158E+02  2.140E+04  0.3209 BX18  +VEHSPD
       245T     249      238     2  4.918E+02  5.705E+05  0.3164 IMPANG  +BARDIA
       123T     108        5     2  5.260E+02  1.345E+06  0.1740 VEHLEN  -BX3
        31T    1590     1590     2  6.164E+02  1.141E+05  0.4156 MAKED  -YEAR

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

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

Node 1: BARSHP = "488", "EOL", "GRL", "MBR", "OTH", "ROR"
  Node 2: HIC-mean = 160.57500
Node 1: BARSHP /= "488", "EOL", "GRL", "MBR", "OTH", "ROR"
  Node 3: BARSHP = "128", "IAT", "SGN", "UNK"
    Node 6: HIC-mean = 260.28889
  Node 3: BARSHP /= "128", "IAT", "SGN", "UNK"
    Node 7: BARSHP = "134", "EOB", "FAB", "LUM", "US1"
      Node 14: BX12 <= 2552.0000  or BX12 = NA & BX12_ = "A"
        Node 28: HIC-mean = 780.47727
      Node 14: not (BX12 <= 2552.0000 or BX12 = NA & BX12_ = "A")
        Node 29: HIC-mean = 276.88302
    Node 7: BARSHP /= "134", "EOB", "FAB", "LUM", "US1"
      Node 15: CLSSPD <= 55.450000
        Node 30: HS <= 325.50000 or NA
          Node 60: RSTFRT = "1"
            Node 120: HIC-mean = 426.69760
          Node 60: RSTFRT /= "1"
            Node 121: HIC-mean = 687.28185
        Node 30: HS > 325.50000
          Node 61: CURBWT <= 1575.0000 or NA
            Node 122: IMPANG in (-1, 1)
              Node 244: HIC-mean = 215.81579
            Node 122: IMPANG not in (-1, 1) or NA
              Node 245: HIC-mean = 491.84337
          Node 61: CURBWT > 1575.0000
```

```
            Node 123: HIC-mean = 525.97222
      Node 15: CLSSPD > 55.450000 or NA
        Node 31: HIC-mean = 616.36541

 ***************************************************************
 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 BARSHP = "488", "EOL", "GRL", "MBR", "OTH", "ROR"
 BARSHP mode = "LCB"
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.3893E+05    19.40       0.1554E-14
 YEAR         -19.21       -19.14       0.9992E-15    1972.        2000.        2017.
 If regressors have missing values, predicted value = 471.00000
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 2: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -84.23      -0.7127       0.4815
 HW           0.5093        2.369       0.2447E-01    414.0        540.9        787.0
 If regressors have missing values, predicted value = 37.875000
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if BARSHP = "128", "IAT", "SGN", "UNK"
 BARSHP mode = "LCB"
 ----------------------------
Node 6: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      260.3        7.731       0.9860E-09
 Predicted mean = 260.28889
```

```
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if BARSHP = "134", "EOB", "FAB", "LUM", "US1"
 BARSHP mode = "LCB"
 ----------------------------
Node 14: Intermediate node
 A case goes into Node 28 if BX12 <= 2552.0000 or BX12_ = "A"
 BX12 mean = 3274.8636
 ----------------------------
Node 28: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.5577E+06    3.041       0.2873E-01
 BARDIA       -2557.       -3.027       0.2917E-01    216.0        217.1        218.0
 If regressors have missing values, predicted value = 469.67568
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 29: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      146.9        3.108       0.2205E-02
 IMPPNT       -20.71       -15.12        0.000       -203.0       -5.867        0.000
 If regressors have missing values, predicted value = 292.79348
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 15: Intermediate node
 A case goes into Node 30 if CLSSPD <= 55.450000
 CLSSPD mean = 49.475944
 ----------------------------
Node 30: Intermediate node
 A case goes into Node 60 if HS <= 325.50000 or NA
 HS mean = 323.46129
 ----------------------------
Node 60: Intermediate node
 A case goes into Node 120 if RSTFRT = "1"
 RSTFRT mode = "1"
 ----------------------------
Node 120: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      439.4        2.708       0.8277E-02
 IMPPNT       -3.173       -2.797       0.6450E-02   -671.0       -62.13        34.00
 If regressors have missing values, predicted value = 358.40476
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
```

```
Node 121: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      687.3        14.51        0.000
 Predicted mean = 687.28185
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 61: Intermediate node
 A case goes into Node 122 if CURBWT <= 1575.0000 or NA
 CURBWT mean = 1689.4824
 ----------------------------
Node 122: Intermediate node
 A case goes into Node 244 if IMPANG in [-1, 1]
 IMPANG mean = 114.67005
 ----------------------------
Node 244: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -704.0       -9.590        0.000
 VEHSPD        21.13        12.61        0.000        23.00        43.55        55.33
 If regressors have missing values, predicted value = 57.000000
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 245: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant    -0.1118E+07   -11.04       0.3331E-15
 BARDIA        4401.        11.04       0.6661E-15    254.0        254.0        255.0
 If regressors have missing values, predicted value = 1317.1818
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 123: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.1481E+05    10.91       0.1647E-02
 BX3          -12.23       -9.761       0.2284E-02    743.0        1069.        1210.
 If regressors have missing values, predicted value = 467.44660
 Predicted values truncated at 0.00000 & 12246.0
 ----------------------------
Node 31: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.5474E+05    33.99       0.1110E-15
 YEAR         -27.07       -33.60       0.1110E-15    1974.        1999.        2017.
 If regressors have missing values, predicted value = 616.36541
 Predicted values truncated at 0.00000 & 12246.0
```

```
----------------------------
Proportion of variance (R-squared) explained by tree model: 0.3546

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

The piecewise-linear regression tree is shown in Figure 17. The angular split "IMPANG in [-1, 1]" suggests that head-on collision is less serious than otherwise.

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: solder data

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
```

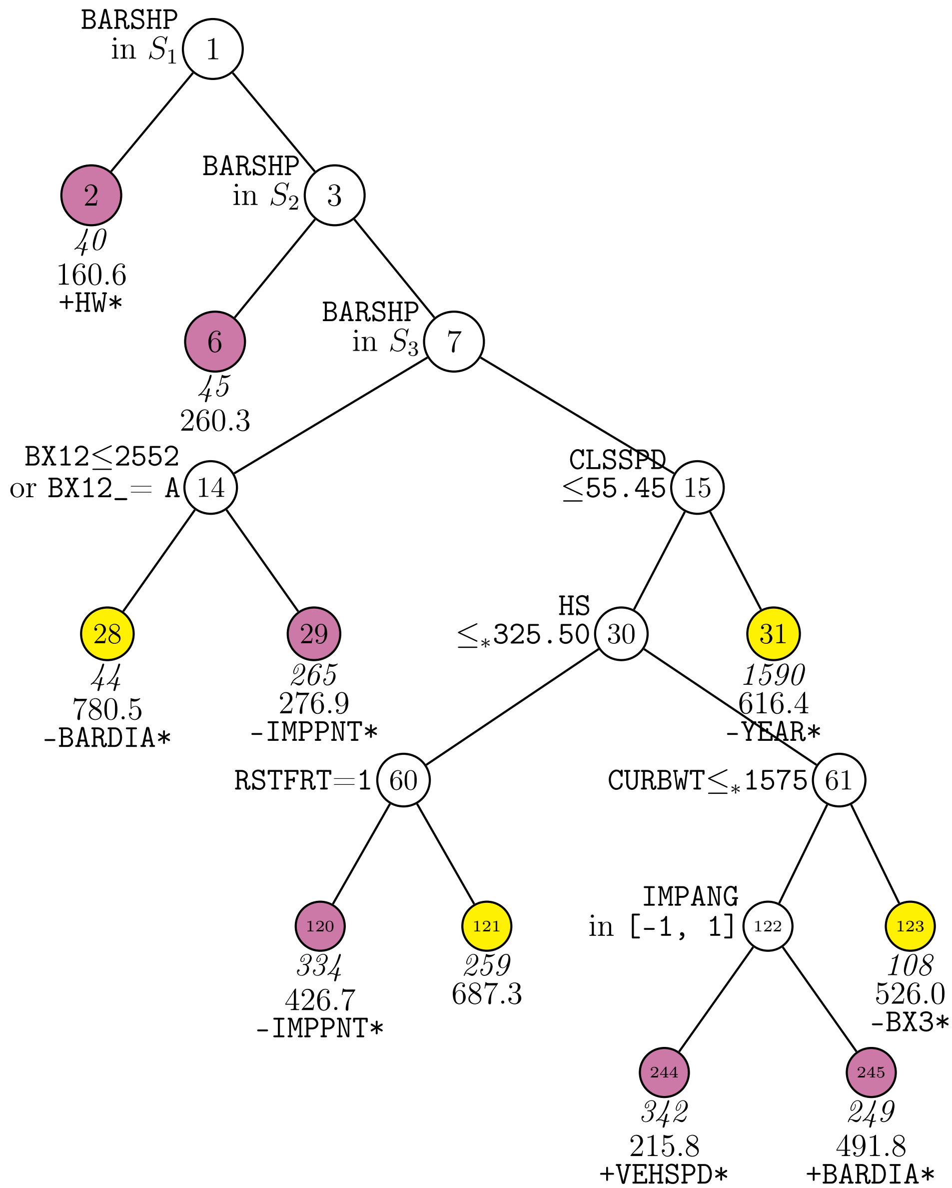

Figure 17: GUIDE v.41.1 0.250-SE piecewise simple linear least-squares regression tree (constant fitted to incomplete cases in terminal nodes) for predicting HIC. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. S_1 = {488, EOL, GRL, MBR, OTH, ROR}. S_2 = {128, IAT, SGN, UNK}. S_3 = {134, EOB, FAB, LUM, US1}. Sample size (in *italics*), mean of HIC, and signed name of regressor variable printed below nodes. Terminal nodes with means above and below value of 512.7 at root node are colored yellow and purple respectively. Asterisk appended to regressor name indicates its slope is significant at the 0.05 level (unadjusted for multiplicity and model search). Second best split variable at root node is IMPANG.

```
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 (recommended for prediction)
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple polynomial, 3: constant ([1:3], <cr>=3):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: solder.dsc
Reading data description file ...
Training sample file: solder.dat
Missing value code: NA
Records in data file start on line 1
Warning: B variables changed to C
D variable is skips
Reading data file ...
Number of records in data file: 720
Length of longest entry in data file: 6
Checking for missing values ...
Finished checking
Assigning integer codes to values of 5 categorical variables
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
Number of cases with positive D values: 478
Rereading data ...
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       720          0          0        0        0        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        5        0
No offset variable in data file.
Number of cases used for training: 720
Number of split variables: 5
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): cons.tex
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: cons.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=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 quite large. One way to reduce the size of the tree is to fit a more complex Poisson regression model in each node.

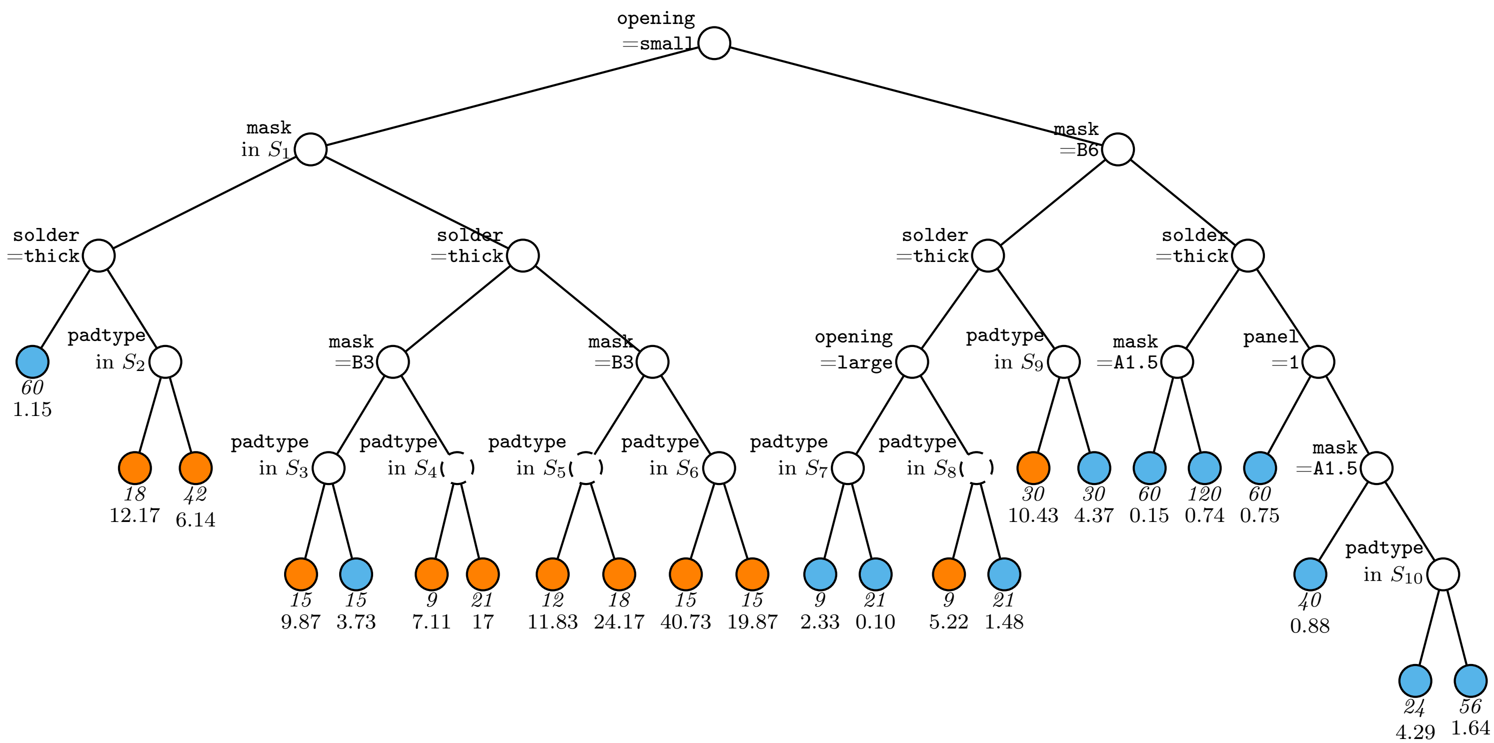

Figure 18: GUIDE v.41.1 0.250-SE piecewise constant Poisson regression tree for predicting `skips`. At each split, an observation goes to the left branch if and only if the condition is satisfied. S_1 = {`A1.5`, `A3`}. S_2 = {`D4`, `D7`, `L4`}. S_3 = {`D4`, `D7`, `L4`, `L7`, `L8`}. S_4 = {`L6`, `L9`, `W9`}. S_5 = {`L6`, `L7`, `L9`, `W9`}. S_6 = {`D4`, `D6`, `D7`, `L4`, `W4`}. S_7 = {`D4`, `W4`, `W9`}. S_8 = {`D7`, `L4`, `L8`}. S_9 = {`D4`, `D7`, `L4`, `L8`, `W4`}. S_{10} = {`D4`, `D7`, `L4`}. Circles with dashed lines are nodes with no significant split variables. 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: solder data

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 (recommended for prediction)
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple polynomial, 3: constant ([1:3], <cr>=3): 1
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: solder.dsc
Reading data description file ...
Training sample file: solder.dat
Missing value code: NA
Records in data file start on line 1
D variable is skips
Reading data file ...
Number of records in data file: 720
Length of longest entry in data file: 6
Checking for missing values ...
Finished checking
Assigning integer codes to values of 5 categorical variables
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
Number of cases with positive D values: 478
GUIDE will try to create the variables in the description file.
```

```
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 17
Creating dummy variables ...
Rereading data ...
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       720          0          0        0        0        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        5        0        0
No offset variable in data file.
Number of cases used for training: 720
Number of split variables: 5
Number of dummy variables created: 17
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): mul.tex
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: mul.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: mul.r
Input rank of top variable to split root node ([1:22], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < mul.in
```

9.2.2 Contents of mul.out

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

Summary information for training sample of size 720
d=dependent, b=split and fit cat variable using indicator variables,
```

```
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight,
z=offset variable
                                                      #Codes/
                                                      Levels/
 Column  Name               Minimum      Maximum    Periods   #Missing
      1  skips          d   0.000        48.00
      2  opening        b                                  3
      3  solder         b                                  2
      4  mask           b                                  4
      5  padtype        b                                 10
      6  panel          b                                  3
  =================== Constructed variables ===================
      7  opening.medium  f    0.000        1.000
      8  opening.small   f    0.000        1.000
      9  solder.thin     f    0.000        1.000
     10  mask.A3         f    0.000        1.000
     11  mask.B3         f    0.000        1.000
     12  mask.B6         f    0.000        1.000
     13  padtype.D6      f    0.000        1.000
     14  padtype.D7      f    0.000        1.000
     15  padtype.L4      f    0.000        1.000
     16  padtype.L6      f    0.000        1.000
     17  padtype.L7      f    0.000        1.000
     18  padtype.L8      f    0.000        1.000
     19  padtype.L9      f    0.000        1.000
     20  padtype.W4      f    0.000        1.000
     21  padtype.W9      f    0.000        1.000
     22  panel.2         f    0.000        1.000
     23  panel.3         f    0.000        1.000

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

Missing regressors imputed with means and missing-value indicators added
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500
```

```
Nodewise interaction tests on all variables
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 10
Minimum node sample size: 7
Top-ranked variables and chi-squared values at root node
     1  0.1782E+02   solder
     2  0.3481E+01   opening
     3  0.3357E+01   mask
     4  0.2453E+00   panel
     5  0.1361E+00   padtype

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

0-SE tree based on mean is marked with * and has 3 terminal nodes
0-SE tree based on median is marked with + and has 3 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

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

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

D-mean is mean of skips in the node
Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix    Node       Node    Split          Other
       label   cases      fit  rank    D-mean    deviance  variable       variables
          1      720      720    18  4.965E+00  1.610E+00  solder
          2T     360      360    17  2.481E+00  1.279E+00  mask
          3      360      360    17  7.450E+00  1.628E+00  opening :mask
          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 regression function for log mean:
 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 regression function for log mean:
 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 regression function for log mean:
 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 regression function for log mean:
 Regressor         Coefficient  t-stat       p-value      Minimum         Mean       Maximum
 Constant         -0.3711       -1.947        0.5284E-01
 mask.A3           0.8061        4.546        0.8965E-05    0.000        0.2500        1.000
 mask.B3            1.008        5.849        0.1735E-07    0.000        0.2500        1.000
 mask.B6            2.267        14.64        0.2220E-15    0.000        0.2500        1.000
 opening.medium    0.1030        1.379        0.1692        0.000        0.5000        1.000
 opening.small      0.000        0.000         1.000        0.000         0.000        0.000
 padtype.D6       -0.7995       -4.649        0.5709E-05    0.000        0.1000        1.000
 padtype.D7       -0.1915       -1.345        0.1800        0.000        0.1000        1.000
 padtype.L4        0.2065        1.601        0.1108        0.000        0.1000        1.000
 padtype.L6       -0.8201       -4.735        0.3894E-05    0.000        0.1000        1.000
 padtype.L7       -0.7595       -4.477        0.1206E-04    0.000        0.1000        1.000
 padtype.L8       -0.3606       -2.413        0.1662E-01    0.000        0.1000        1.000
 padtype.L9       -0.6660       -4.051        0.7039E-04    0.000        0.1000        1.000
 padtype.W4       -0.2254       -1.568        0.1183        0.000        0.1000        1.000
 padtype.W9        -1.747       -7.027        0.2514E-10    0.000        0.1000        1.000
 panel.2           0.5841        5.732        0.3190E-07    0.000        0.3333        1.000
 panel.3           0.6931        6.931        0.4388E-10    0.000        0.3333        1.000
 solder.thin        0.000        0.000         1.000        1.000         1.000        1.000
 ----------------------------
 Observed and fitted values are stored in mul.fit
 LaTeX code for tree is in mul.tex
 R code is stored in mul.r
```

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

9.3 Offset variable: lung cancer data

We use a data set from an epidemiological study of the effect of public drinking water on cancer mortality in Missouri (Choi et al., 2005). The data file `lungcancer.txt` gives the number of deaths (`deaths`) from lung cancer among 115 counties (`county`) during the period 1972–1981 for both sexes (`sex`) and four age groups (`agegp`): 45–54, 55–64, 65–74, and over 75. The description file `lungcancer.dsc` below lists

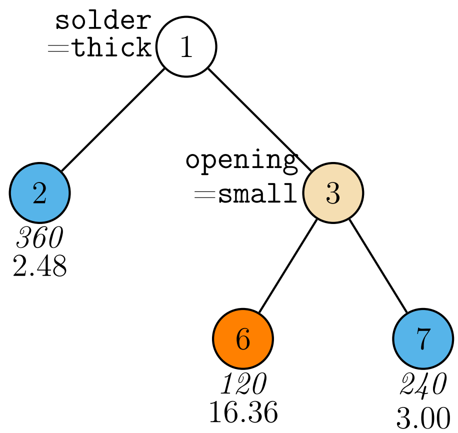

Figure 19: GUIDE v.41.1 0.250-SE multiple linear Poisson regression tree (missing regressor values imputed and missing indicators added) for predicting `skips`. At each split, an observation goes to the left branch if and only if the condition is satisfied. Intermediate nodes with splits due to interaction 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`.

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 `pop` is excluded (`x`) from the analysis. The contents of `lungcancer.dsc` are:

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

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

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

This is achieved by fitting a linear Poisson regression model with `sex` as `b` so that its dummy indicator variable serves as a linear predictor in the Poisson node models.

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 (recommended for prediction)
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple polynomial, 3: constant ([1:3], <cr>=3): 1
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: lungcancer.dsc
Reading data description file ...
Training sample file: lungcancer.txt
Missing value code: NA
Records in data file start on line 1
D variable is deaths
Reading data file ...
Number of records in data file: 920
Length of longest entry in data file: 8
Checking for missing values ...
Finished checking
Assigning integer codes to values of 3 categorical variables
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
Number of cases with positive D values: 869
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
```

```
Number of dummy variables created: 1
Creating dummy variables ...
Rereading data ...
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       920          0          0        1        0        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        1        2        0
Offset variable in column:          6
Number of cases used for training: 920
Number of split variables: 3
Number of dummy variables created: 1
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): poi.tex
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: poi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=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 regressors imputed with means and missing-value indicators added
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 10
Minimum node sample size: 7
Top-ranked variables and 1-df 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       48   2.863E+00   2.930E-01   2.063E-01   2.836E+00   2.864E-01
   2       47   2.863E+00   2.930E-01   2.063E-01   2.836E+00   2.864E-01
   :
  34        4   2.352E+00   3.233E-01   2.640E-01   2.262E+00   3.370E-01
  35**      3   2.249E+00   3.278E-01   2.705E-01   1.954E+00   2.648E-01
```

```
  36        2   4.702E+00   8.054E-01   4.866E-01   4.153E+00   6.629E-01
  37        1   9.431E+00   1.420E+00   9.674E-01   9.043E+00   9.329E-01

0-SE tree based on mean is marked with * and has 3 terminal nodes
0-SE tree based on median is marked with + and has 3 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

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

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

Rate is mean of Y/exp(offset)
Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix    Node       Node    Split          Other
       label   cases      fit  rank     rate     deviance  variable       variables
          1     920      920     2  1.382E-02  9.179E+00  agegp
          2T    230      230     2  5.493E-03  1.863E+00  county
          3     690      690     2  1.763E-02  4.357E+00  agegp
          6T    230      230     2  1.339E-02  3.003E+00  county
          7T    460      460     2  2.093E-02  1.802E+00  agegp

Number of terminal nodes of final tree: 3
Total number of nodes of final tree: 5
Second best split variable (based on curvature test) at root node is sex

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

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

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

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

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

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

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

Node 1: Intermediate node
 A case goes into Node 2 if agegp = "45-54"
 agegp mode = "45-54"
 Coefficients of regression function for log expected rate:
 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 regression function for log expected rate:
 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 regression function for log expected rate:
 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 regression function for log expected rate:
 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
```

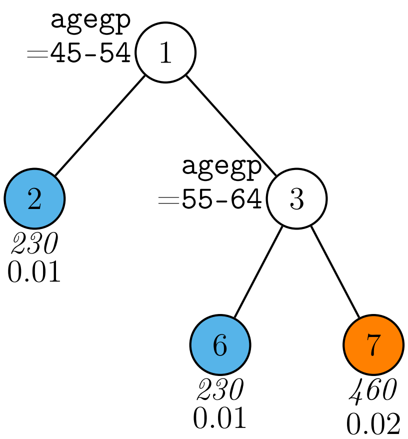

Figure 20: GUIDE v.41.1 0.250-SE multiple linear Poisson regression tree (constant fitted to incomplete cases in terminal nodes) for predicting rate of `deaths`. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*) and sample rate printed below nodes. Terminal nodes with rates above and below value of 0.014 at root node are colored orange and skyblue respectively. Second best split variable at root node is `sex`.

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.

10 Censored response: RHC data

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

```
rhcdata.txt
NA
2
1 X x
2 cat1 c
3 cat2 c
4 ca c
5 sadmdte x
```

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

```
52 gastr c
53 renal c
54 meta c
55 hema c
56 seps c
57 trauma c
58 ortho c
59 adld3p n
60 urin1 n
61 race c
62 income c
63 ptid x
64 survtime t
```

10.1 Proportional hazards

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

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

10.1.1 Input file generation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: censored.in
 Input 1 for model fitting, 2 for importance or DIF scoring,
       3 for data conversion ([1:3], <cr>=1):
 Name of batch output file: censored.out
 Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
 Input 1 for classification, 2 for regression, 3 for propensity score grouping
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
```

```
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple linear in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best 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
23 N variables changed to S
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
Smallest uncensored survtime: 2.0000
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime with positive weight: 1943.0000 1351.0000
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       31        0
Survival time variable in column: 64
```

```
Event indicator variable in column: 9
Proportion uncensored among nonmissing T and D variables: .649
Number of cases used for training: 5735
Number of split variables: 54
Number of cases excluded due to 0 W or missing D or T variables: 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
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: censored.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: censored.r
Input rank of top variable to split root node ([1:54], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < censored.in
```

10.1.2 Output file

```
Regression tree for censored response
Pruning by cross-validation
Data description file: rhcdsc2.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
23 N variables changed to S
D variable is death
Piecewise constant model
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
Smallest uncensored survtime: 2.0000
Largest uncensored and censored survtime with positive weight: 1943.0000 1351.0000

Summary information for training sample of size 5735
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
```

```
m=missing-value flag variable, p=periodic variable, w=weight,
t=survival time variable
                                                 #Codes/
                                                 Levels/
 Column  Name           Minimum      Maximum    Periods   #Missing
      2  cat1       c                                 9
      3  cat2       c                                 6     4535
      4  ca         c                                 3
      9  death      d    0.000       1.000
      :
     58  ortho      c                                 2
     59  adld3p     s    0.000       7.000                  4296
     60  urin1      s    0.000       9000.                  3028
     61  race       c                                 3
     62  income     c                                 4
     64  survtime   t    2.000       1943.
  =================== Constructed variables ===================
     65  lnbasehaz  z   -3.818       2.038

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

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 15
Minimum node sample size: 57
Number of iterations for fitting: 20
Top-ranked variables and 1-df chi-squared values at root node
     1  0.7573E+03   surv2md1
     2  0.3288E+03   adld3p
     3  0.2341E+03   cat1
     4  0.2263E+03   aps1
```

```
     :
    51  0.1094E-01   chrpulhx
    52  0.8247E-02   cardiohx

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

 0-SE tree based on mean is marked with * and has 10 terminal nodes
 0-SE tree based on median is marked with + and has 10 terminal nodes
 Selected-SE tree based on mean using naive SE is marked with **
 Selected-SE tree based on mean using bootstrap SE is marked with --
 Selected-SE tree based on median and bootstrap SE is marked with ++
 * tree same as + tree
 ** tree same as + tree
 ** tree same as -- tree
 * tree same as ** tree
 * tree same as -- tree

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

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

 Cases fit give the number of cases used to fit node
 Deviance is mean residual deviance for all cases in node
        Node    Total    Cases Matrix   Median      Node    Split          Other
        label   cases      fit  rank   survtime   deviance  variable       variables
           1     5735     5735     1  1.920E+02  1.459E+00  surv2md1
           2     2164     2164     1  2.300E+01  1.499E+00  adld3p
           4     1930     1930     1  1.800E+01  1.530E+00  surv2md1
           8T     709      709     1  1.100E+01  1.429E+00  cat1
           9     1221     1221     1  2.800E+01  1.498E+00  dnr1
          18T    1027     1027     1  3.700E+01  1.434E+00  surv2md1
          19T     194      194     1  8.000E+00  1.431E+00  aps1
           5T     234      234     1  1.950E+02  9.294E-01  ca
```

```
          3     3571     3571     1  3.290E+02  1.223E+00  surv2md1
          6     1805     1805     1  2.270E+02  1.347E+00  adld3p
         12     1364     1364     1  1.290E+02  1.457E+00  dnr1
         24T    1214     1214     1  1.710E+02  1.412E+00  das2d3pc
         25T     150      150     1  2.550E+01  1.600E+00  hema1
         13T     441      441     1  3.750E+02  8.602E-01  das2d3pc
          7     1766     1766     1  4.030E+02  1.019E+00  chfhx
         14     1276     1276     1  4.410E+02  1.036E+00  das2d3pc
         28T     815      815     1  3.640E+02  1.065E+00  wtkilo1
         29T     461      461     1  6.720E+02  9.083E-01  surv2md1
         15T     490      490     1  3.730E+02  9.322E-01  surv2md1

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

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

Node 1: surv2md1 <= 0.56447053
  Node 2: adld3p = NA
    Node 4: surv2md1 <= 0.35847378
      Node 8: Median survival time = 11.000000
    Node 4: surv2md1 > 0.35847378 or NA
      Node 9: dnr1 = "No"
        Node 18: Median survival time = 37.000000
      Node 9: dnr1 /= "No"
        Node 19: Median survival time = 8.0000000
  Node 2: adld3p /= NA
    Node 5: Median survival time = 195.00000
Node 1: surv2md1 > 0.56447053 or NA
  Node 3: surv2md1 <= 0.71744752
    Node 6: adld3p = NA
      Node 12: dnr1 = "No"
        Node 24: Median survival time = 171.00000
      Node 12: dnr1 /= "No"
        Node 25: Median survival time = 25.500000
    Node 6: adld3p /= NA
      Node 13: Median survival time = 375.00000
  Node 3: surv2md1 > 0.71744752 or NA
    Node 7: chfhx = "0"
      Node 14: das2d3pc <= 23.857420
        Node 28: Median survival time = 364.00000
      Node 14: das2d3pc > 23.857420 or NA
        Node 29: Median survival time = 672.00000
    Node 7: chfhx /= "0"
```

```
      Node 15: Median survival time = 373.00000

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

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

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

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

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

Node 1: Intermediate node
 A case goes into Node 2 if surv2md1 <= 0.56447053
 surv2md1 mean = 0.59245008
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value
 Constant      0.000
 Predicted log-relative hazard = 192.00000
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if adld3p = NA
 adld3p mean = 1.3589744
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if surv2md1 <= 0.35847378
 surv2md1 mean = 0.38175857
 ----------------------------
Node 8: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value
 Constant      1.015
 Predicted log-relative hazard = 11.000000
 ----------------------------
:
Node 15: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value
 Constant    -0.4135
 Predicted log-relative hazard = 373.00000
 ----------------------------
 Observed and fitted values are stored in censored.fit
```

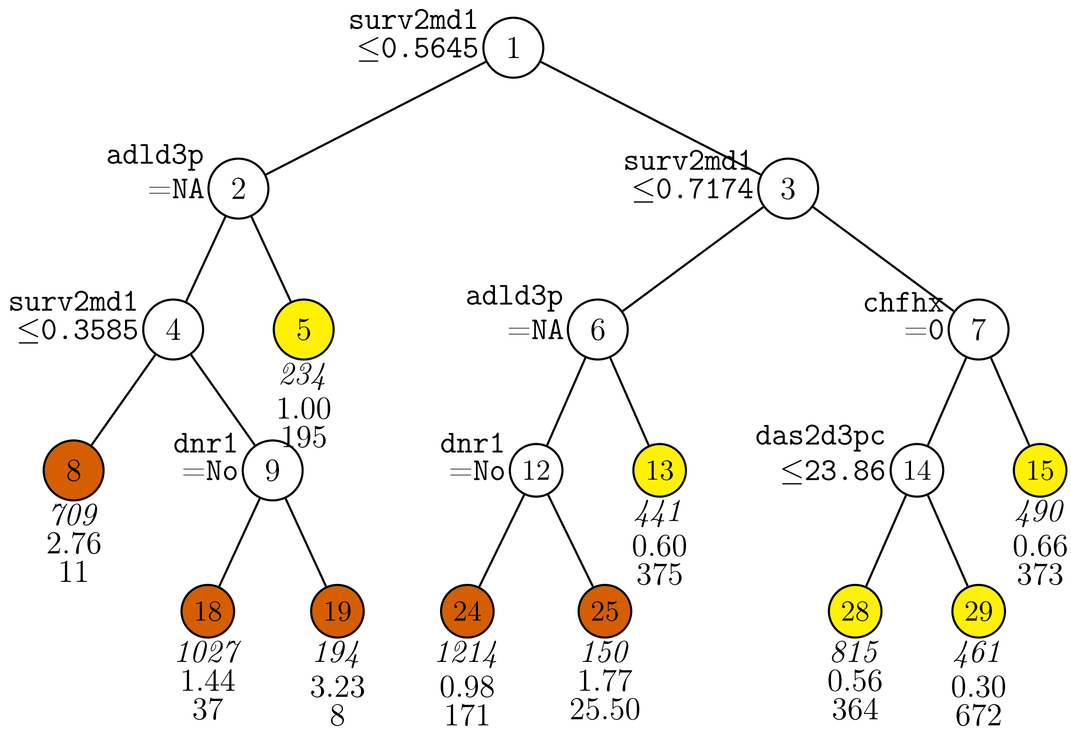

Figure 21: GUIDE v.41.1 0.250-SE piecewise-constant proportional hazards regression tree for `survtime`. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*), relative hazard, and median survival time printed below nodes. Terminal nodes with median survival times above and below 192 (median at root node) are colored yellow and vermillion respectively. Second best split variable at root node is `adld3p`.

```
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     observed       event logbasecumhaz      survivalprob  mediansurvtime
    y       13     240.000          n   -0.261185        0.631158          375.000
    y       15     45.0000          y   -0.804384        0.743903          373.000
    y        8     317.000          n   -0.500244E-001   0.725445E-001     11.0000
    y       18     37.0000          y   -0.889004        0.553180          37.0000
    y       19     2.00000          y     -4.01055       0.943144          8.00000
```

The columns are:

train: equals y if observation is used for model fitting; equals n if not used.

node: terminal node label of observation.

observed: observed survival time (t variable in description file).

event: equals y if observed is uncensored (d=1); equals n if censored (d=0).

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

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

$$\begin{aligned}\exp\{-\Lambda_0(t)\exp(\boldsymbol{\beta}'\mathbf{x})\} &= \exp\{-\exp(\beta_0 + \texttt{logbasecumhaz})\} \\ &= \exp(-\exp(-0.514911594896 - 0.261185)) \\ &= 0.6311581\end{aligned}$$

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

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

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

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

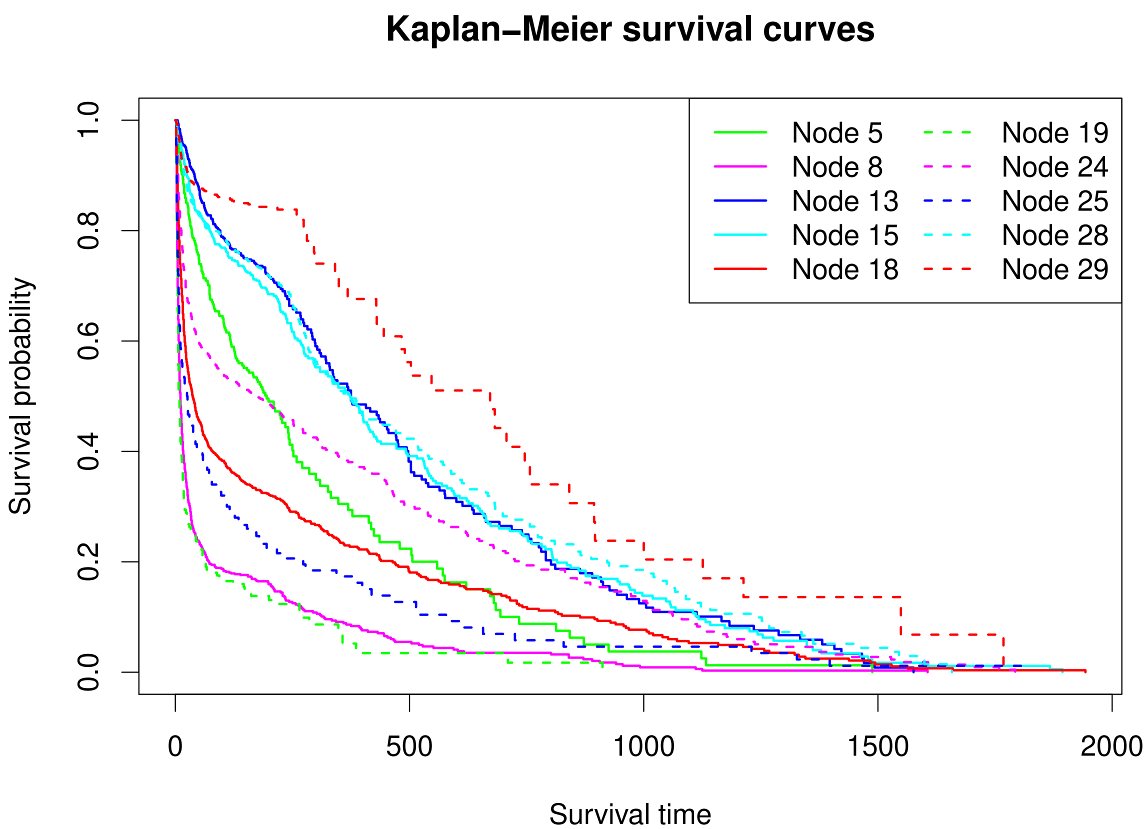

Figure 22: Kaplan-Meier survival curves for data in terminal nodes of Figure 21

10.2 Restricted mean event time

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

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 for prediction)
Choose 1 for multiple regression
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise 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
23 N variables changed to S
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
```

```
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime with positive weight: 1943.0000 1351.0000
Smallest observed uncensored time is 2.0000
Largest observed censored or uncensored time is 1943.0000
Input restriction on event time ([2.00:1943.00], <cr>=972.00):

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       31        0
No weight variable in data file
Number of cases used for training: 3732
Number of split variables: 54
Number of cases excluded due to 0 W or missing D variable: 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
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: rest.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: rest.r
Input rank of top variable to split root node ([1:54], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < rest.in
```

10.2.2 Contents of `rest.out`

```
Restricted mean event time regression tree
Pruning by cross-validation
Data description file: rhcdsc2.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
23 N variables changed to S
D variable is death
Piecewise constant model
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Smallest uncensored survtime: 2.0000
Largest uncensored and censored survtime with positive weight: 1943.0000 1351.0000
Interval for restricted mean event time is from 0 to 972.

Summary information for training sample of size 3732 (excluding observations with
non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                #Codes/
                                                Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      2  cat1       c                                 9
      3  cat2       c                                 6     2807
      4  ca         c                                 3
      9  death      d    0.000        1.000
      :
     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       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       31        0
No weight variable in data file
Number of cases used for training: 3732
Number of split variables: 54
```

```
Number of cases excluded due to 0 W or missing D variable: 2003

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

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

Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       75   1.121E+05   3.376E+03   2.477E+03   1.120E+05   2.101E+03
   2       74   1.121E+05   3.376E+03   2.477E+03   1.119E+05   2.107E+03
   :
  42       10   1.105E+05   3.343E+03   2.142E+03   1.106E+05   2.773E+03
  43+       8   1.086E+05   3.212E+03   2.008E+03   1.082E+05   3.190E+03
  44        7   1.086E+05   3.184E+03   2.177E+03   1.086E+05   3.279E+03
  45**      6   1.067E+05   3.063E+03   1.467E+03   1.084E+05   2.196E+03
  46        4   1.091E+05   3.044E+03   1.503E+03   1.090E+05   2.580E+03
  47        3   1.097E+05   3.045E+03   1.425E+03   1.090E+05   1.927E+03
  48        2   1.102E+05   3.062E+03   1.527E+03   1.102E+05   2.279E+03
  49        1   1.225E+05   3.100E+03   2.805E+02   1.225E+05   4.687E+02

0-SE tree based on mean is marked with * and has 6 terminal nodes
0-SE tree based on median is marked with + and has 8 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as ++ tree
** tree same as -- tree
++ tree same as -- tree
* tree same as ** tree
* tree same as ++ tree
* tree same as -- tree

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

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

D-mean is weighted mean of death in the node
Cases fit give the number of cases used to fit node
MSE is residual sum of squares divided by number of cases in node
       Node    Total    Cases Matrix    Node      Node      Split          Interacting
       label   cases      fit  rank    D-mean     MSE       variable       variable
          1     3732     3732     1  3.144E+02  1.800E+05   adld3p
          2      664      664     1  4.685E+02  2.273E+05   surv2md1
          4T     168      168     1  3.244E+02  1.404E+05   immunhx
          5      496      496     1  5.040E+02  2.427E+05   urin1
         10T     314      314     1  5.756E+02  2.829E+05   sod1
         11T     182      182     1  3.515E+02  1.074E+05   race
          3     3068     3068     1  2.647E+02  1.556E+05   surv2md1
          6T    1262     1262     1  1.607E+02  8.878E+04   dnr1
          7     1806     1806     1  3.225E+02  1.880E+05   urin1
         14T    1000     1000     1  4.001E+02  2.482E+05   surv2md1
         15T     806      806     1  2.057E+02  8.243E+04   swang1 :immunhx

Number of terminal nodes of final tree: 6
Total number of nodes of final tree: 11
Second best split variable (based on curvature test) at root node is surv2md1

Regression tree:

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

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

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

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

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

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

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

```
 A case goes into Node 6 if surv2md1 <= 0.49098337
 surv2md1 mean = 0.54259828
 ----------------------------
Node 6: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      160.7        19.16        0.000
 survtime mean = 160.701
 ----------------------------
Node 7: Intermediate node
 A case goes into Node 14 if urin1 = NA
 urin1 mean = 1998.7301
 ----------------------------
Node 14: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      400.1        25.39        0.000
 survtime mean = 400.063
 ----------------------------
Node 15: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      205.7        20.34        0.000
 survtime mean = 205.708
 ----------------------------
 Observed and fitted values are stored in rest.fit
 LaTeX code for tree is in rest.tex
 R code is stored in rest.r
```

Figure 23 shows the restricted mean event time tree.

11 Randomized treatments

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

There are two types of covariates for identification of subgroups with differential treatment effects. A *prognostic* variable is a clinical or biologic characteristic that provides information on the likely outcome of the disease in an untreated individual (e.g., patient age, family history, disease stage, and prior therapy). A *predictive*

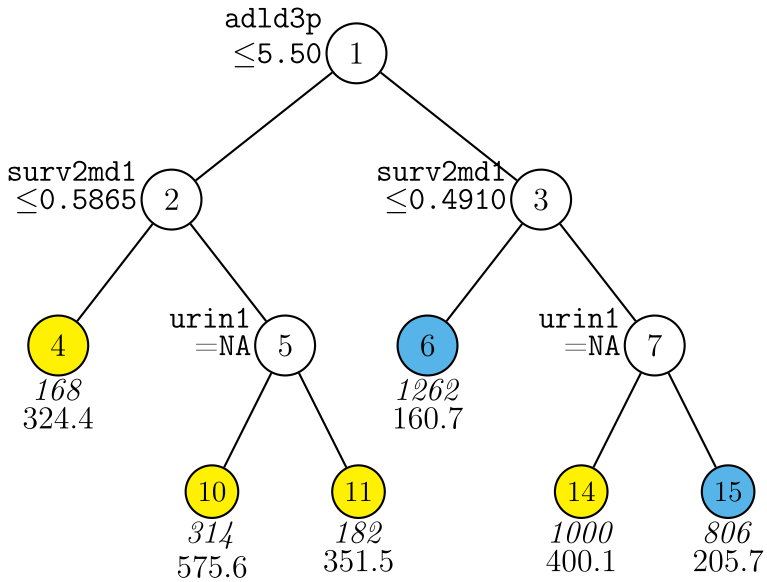

Figure 23: GUIDE v.41.1 0.250-SE piecewise-constant regression tree for mean `survtime` restricted to less than 972.000. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*) and restricted mean of `survtime` printed below nodes. Terminal nodes with means above and below value of 314.4 at root node are colored yellow and skyblue respectively. Second best split variable at root node is `surv2md1`.

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).

11.1 Multiple treatment arms: CAPE data

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

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

11.1.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: gi.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: gi.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1):
Input 1 for least squares, 2 least median of squares ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended for prediction)
Choose 1 for multiple regression
```

```
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise ANCOVA ([0:4], <cr>=3):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: cape.dsc
Reading data description file ...
Training sample file: cape.txt
Missing value code: NA
Records in data file start on line 1
R variable present
21 N variables changed to S
Warning: model changed to linear in treatment
D variable is resp6
Reading data file ...
Number of records in data file: 1681
Length of longest entry in data file: 25
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 18 categorical variables
Finished assigning codes to 10 categorical variables
Treatment (R) variable is group with values "Control", "DVD", and "Phone"
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 2
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
Proportion of training sample for each level of group
"Control"    0.3278
    "DVD"    0.3309
  "Phone"    0.3413
     Total  #cases w/   #missing
```

```
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      1681         43         84        1        0        0       21
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       17        0        1
No weight variable in data file
Number of cases used for training: 1638
Number of split variables: 38
Number of dummy variables created: 2
Number of cases excluded due to 0 W or missing D or R variables: 43
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): gi.tex
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: gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: gi.r
Input rank of top variable to split root node ([1:41], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < gi.in
```

11.1.2 Contents of gi.out

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

```
"Control"    0.3278
    "DVD"    0.3309
  "Phone"    0.3413

Summary information for training sample of size 1638 (excluding observations with
non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                      #Codes/
                                                      Levels/
 Column  Name              Minimum      Maximum    Periods   #Missing
      1  resp6         d    0.000        1.000
      3  group         r                                  3
      4  age           s    41.00        75.00                    1
      5  educyrs       s    2.000        20.00
      :
     39  fatal         s    11.00        42.00
     40  know          s    1.000        7.000
     41  stage         c                                  4
  ================== Constructed variables ===================
     42  group.DVD     f    0.000        1.000
     43  group.Phone   f    0.000        1.000

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

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

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 11
Minimum node sample size: 8
Minimum fraction of cases per treatment at each node: 0.066
```

```
Top-ranked variables and 1-df chi-squared values at root node
     1  0.6775E+01   sf12gh
     2  0.5072E+01   know
     3  0.3940E+01   incle75k
     :
    30  0.1110E-03   sf12pf
    31  0.1774E-07   sf12mh

Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1      125   3.439E-01   9.506E-03   1.064E-02   3.585E-01   1.561E-02
   2      124   3.439E-01   9.506E-03   1.064E-02   3.585E-01   1.561E-02
   :
  76       13   2.732E-01   6.053E-03   5.569E-03   2.691E-01   6.443E-03
  77       12   2.491E-01   4.721E-03   6.754E-03   2.462E-01   6.768E-03
  78**      5   2.390E-01   3.240E-03   2.264E-03   2.410E-01   3.959E-03
  79++      1   2.414E-01   2.372E-03   5.044E-04   2.410E-01   6.719E-04

0-SE tree based on mean is marked with * and has 5 terminal nodes
0-SE tree based on median is marked with + and has 1 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
** tree same as -- tree
+ tree same as ++ tree
* tree same as ** tree
* tree same as -- tree

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

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

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

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

 Regression tree:

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

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

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

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

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

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

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

```
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if know <= 6.5000000
 know mean = 5.6087154
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if educyrs <= 15.500000
 educyrs mean = 13.800853
 ----------------------------
Node 8: Intermediate node
 A case goes into Node 16 if yearmam <= 3.5000000
 yearmam mean = 2.0055249
 ----------------------------
Node 16: Terminal node
 Coefficients of least squares regression functions:
 Regressor      Coefficient  t-stat      p-value     Minimum         Mean       Maximum
 Constant       0.3333        8.279       0.2776E-14
 group.DVD     -0.9843E-01   -1.790       0.7419E-01    0.000       0.3489        1.000
 group.Phone    0.2237E-02   0.4068E-01   0.9676        0.000       0.3489        1.000
 resp6 mean = 0.299766
 No truncation of predicted values
 ----------------------------
Node 17: Terminal node
 Coefficients of least squares regression functions:
 Regressor      Coefficient  t-stat      p-value     Minimum         Mean       Maximum
 Constant       0.5000        6.149       0.1204E-07
 group.DVD      0.1154        1.037       0.3019        0.000       0.3362        1.000
 group.Phone    0.2674        2.458       0.1550E-01    0.000       0.3707        1.000
 resp6 mean = 0.637931
 No truncation of predicted values
 ----------------------------
Node 9: Terminal node
 Coefficients of least squares regression functions:
 Regressor      Coefficient  t-stat      p-value     Minimum         Mean       Maximum
 Constant       0.3788        6.298       0.2840E-08
 group.DVD      0.2366        2.611       0.9889E-02    0.000       0.3250        1.000
 group.Phone   -0.2165E-01  -0.2244       0.8227        0.000       0.2625        1.000
 resp6 mean = 0.450000
 No truncation of predicted values
 ----------------------------
Node 5: Terminal node
 Coefficients of least squares regression functions:
 Regressor      Coefficient  t-stat      p-value     Minimum         Mean       Maximum
 Constant       0.1831        3.417       0.7695E-03
 group.DVD      0.2883        3.791       0.1993E-03    0.000       0.3500        1.000
 group.Phone    0.1050        1.321       0.1882        0.000       0.2950        1.000
```

```
 resp6 mean = 0.315000
 No truncation of predicted values
 ----------------------------
Node 3: Terminal node
 Coefficients of least squares regression functions:
 Regressor      Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant        0.4895        15.21        0.000
 group.DVD     -0.1101       -2.407       0.1634E-01    0.000       0.3156        1.000
 group.Phone   -0.3832E-01  -0.8659       0.3868        0.000       0.3619        1.000
 resp6 mean = 0.440816
 No truncation of predicted values
 ----------------------------
 Number of times Li-Martin approximation used = 157
 Proportion of variance (R-squared) explained by tree model: 0.0579

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

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

11.2 Censored response: proportional hazards

We now consider a randomized controlled breast cancer trial where the response variable is a censored survival time (Schmoor et al., 1996). The data are in the file `cancerdata.txt`; they are included in the `TH.data` R package (Hothorn, 2017) as well. In the description file `cancerdsc.txt` below, the treatment variable is hormone therapy, `horTh`. The variable `time` is (censored) time to recurrence of cancer and the event indicator `event` = 1 if the cancer recurred and = 0 if it did not. Ordinal predictor variables may be designated as "`n`" or "`s`" (with this option of no linear prognostic control, `n` variables are automatically changed to `s` when the program executes). See Loh et al. (2019a, 2016, 2015, 2019c) and Loh and Zhou (2020) for further analysis of the data.

```
cancerdata.txt
NA
1
1 horTh r
2 age n
3 menostat c
4 tsize n
5 tgrade c
```

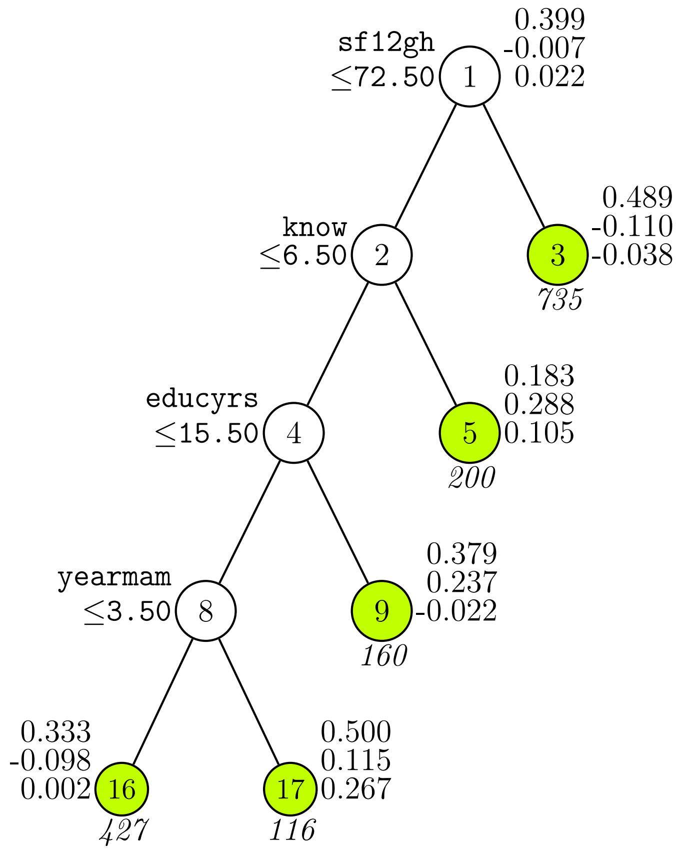

Figure 24: GUIDE v.41.2 0.250-SE least-squares regression tree using Gi option for dependent variable `resp6` without adjustment for linear prognostic effects. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *(italics*) printed below nodes. `resp6` mean for treatment reference level `Control` followed by treatment effects of levels `DVD`, `Phone` (relative to `Control`) beside nodes. Second best split variable at root node is `know`.

```
6 pnodes n
7 progrec n
8 estrec n
9 time t
10 event d
```

11.2.1 Without linear prognostic control

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

Input file generation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: ph-gi.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: ph-gi.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple linear in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple linear, 3: constant ([1:3], <cr>=3):
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: cancerdsc.txt
Reading data description file ...
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
6 N variables changed to S
Warning: model changed to linear in treatment
```

```
D variable is death
Reading data file ...
Number of records in data file: 686
Length of longest entry in data file: 4
Checking for missing values ...
Finished checking
Assigning integer codes to values of 2 categorical variables
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
Smallest uncensored time: 72.0000
Number of cases dropped due to missing D or T or censored T < smallest uncensored T: 14
Number of complete cases excluding censored T < smallest uncensored T: 672
Number of cases used to compute baseline hazard: 672
Number of cases with D=1 and T >= smallest uncensored: 299
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 1
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables ...
Rereading data ...
Largest uncensored and censored time by horTh
   "no"      2456.0000    2563.0000
  "yes"      2372.0000    2659.0000
Proportion of training sample for each level of horTh
 "no"    0.6399
"yes"    0.3601
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       686          0          0        0        0        0        6
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        1        0        1
Survival time variable in column: 9
Event indicator variable in column: 10
Proportion uncensored among nonmissing T and D variables: .445
Number of cases used for training: 672
Number of split variables: 7
Number of dummy variables created: 1
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): ph-gi.tex
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: ph-gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: ph-gi.r
Input rank of top variable to split root node ([1:9], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < ph-gi.in
```

Results The contents of ph-gi.out follow.

```
Regression tree for censored response
Pruning by cross-validation
Data description file: cancerdsc.txt
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
6 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Piecewise linear model
Number of records in data file: 686
Length of longest entry in data file: 4
Treatment (R) variable is horTh with values "no" and "yes"
Number of cases dropped due to missing D or T or censored T < smallest uncensored T: 14
Number of complete cases excluding censored T < smallest uncensored T: 672
Number of cases used to compute baseline hazard: 672
Number of cases with D=1 and T >= smallest uncensored: 299
Number of dummy variables created: 1
Smallest uncensored time: 72.0000
Largest uncensored and censored time by horTh
  horTh     Uncensored     Censored
   "no"      2456.0000    2563.0000
  "yes"      2372.0000    2659.0000
Proportion of training sample for each level of horTh
 "no"    0.6399
"yes"    0.3601

Summary information for training sample of size 672 (excluding observations with
non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
```

```
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight,
t=survival time variable
                                                 #Codes/
                                                 Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  horTh      r                                  2
      2  age        s    21.00        80.00
      3  menostat   c                                  2
      4  tsize      s    3.000        120.0
      5  tgrade     s    1.000        3.000
      6  pnodes     s    1.000        51.00
      7  progrec    s    0.000        2380.
      8  estrec     s    0.000        1144.
      9  time       t    72.00        2659.
     10  death      d    0.000        1.000
  =================== Constructed variables ===================
     11  lnbasehaz  z   -6.510       0.5887E-01
     12  horTh.yes  f    0.000        1.000

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       686         0          0        0        0        0        6
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        1        0        1
Survival time variable in column: 9
Event indicator variable in column: 10
Proportion uncensored among nonmissing T and D variables: 0.445
Number of cases used for training: 672
Number of split variables: 7
Number of dummy variables created: 1

Constant fitted to cases with missing values in regressor variables
Predictive priority (Gi)
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 10
Minimum node sample size: 6
Minimum fraction of cases per treatment at each node: 0.072
Number of iterations for fitting: 20
Top-ranked variables and 1-df 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.

Cases fit give the number of cases used to fit node
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix   Median      Node    Split
       label   cases      fit  rank   survtime   deviance  variable
          1      672      672     1  1.807E+03  1.431E+00  progrec
          2T     274      274     1  1.140E+03  1.601E+00  estrec
          3T     398      398     1  2.286E+03  1.188E+00  menostat

Number of terminal nodes of final tree: 2
Total number of nodes of final tree: 3
Second best split variable (based on curvature test) at root node is estrec

Regression tree:

Node 1: progrec <= 21.500000
  Node 2: Median survival time = 1140.0000
Node 1: progrec > 21.500000 or NA
  Node 3: Median survival time = 2286.0000
```

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

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

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

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

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

Node 1: Intermediate node
 A case goes into Node 2 if progrec <= 21.500000
 progrec mean = 110.91518
 Coefficients of log-relative hazard function (relative to baseline hazard):
 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
 Predicted log-relative hazard = -0.30206062E-2
 ----------------------------
Node 2: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 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
 Predicted log-relative hazard = 0.45682185
 ----------------------------
Node 3: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 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
 Predicted log-relative hazard = -0.34487497
 ----------------------------
 Observed and fitted values are stored in ph-gi.fit
 LaTeX code for tree is in ph-gi.tex
 R code is stored in ph-gi.r
```

Let $\lambda(u, \mathbf{x})$ denote the hazard function at time u and predictor values $\mathbf{x}$ and let $\lambda_0(u)$ denote the baseline hazard function. The results in `ph-gi.out` show that the

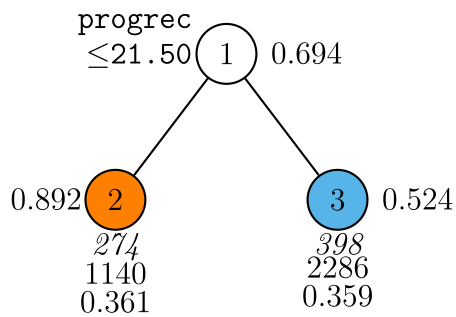

Figure 25: GUIDE v.41.1 0.250-SE proportional hazards regression tree using Gi option for `time` and event indicator `death` without adjustment for linear prognostic effects. (constant fitted to incomplete cases in terminal nodes). At each split, an observation goes to the left branch if and only if the condition is satisfied. Treatment `horTh` hazard ratio of level `yes` to level `no` beside nodes. Sample size (in *italics*), median survival time, and proportion of `horTh` = `yes` printed below nodes. Terminal nodes with treatment hazard ratio above and below 0.694 (ratio at root node) are colored orange and skyblue respectively. Second best split variable at root node is `estrec`.

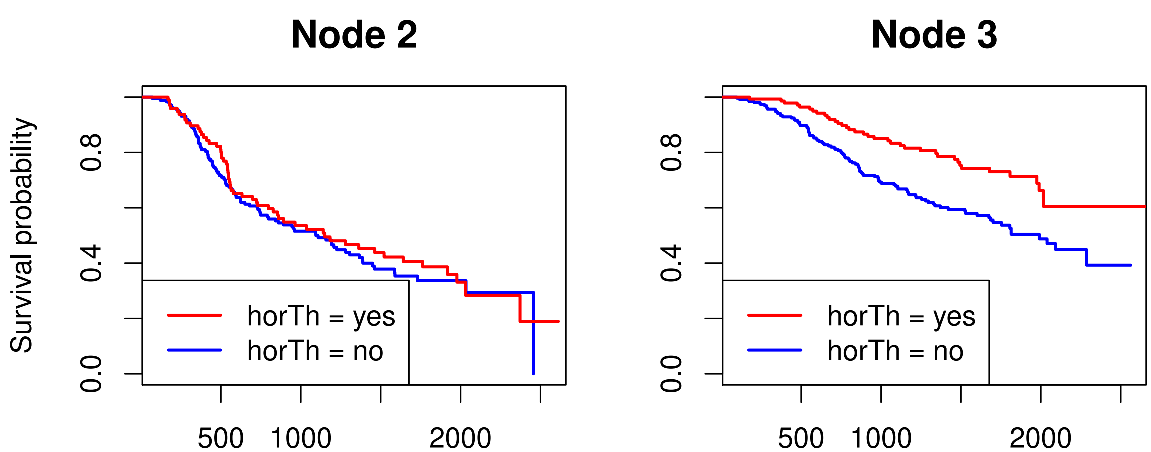

Figure 26: Estimated survival probability functions for breast cancer data

fitted proportional hazards model is

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

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

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

```
library(survival)
z <- read.table("cancerdata.txt",header=TRUE)
leg.txt <- c("horTh = yes","horTh = no")
leg.col <- c("red","blue")
leg.lty <- 1:2
xr <- range(z$time)
zg <- read.table("ph-gi.fit",header=TRUE)
nodes <- zg$node
uniq.gp <- unique(sort(nodes))
plotted <- FALSE
for(g in uniq.gp){
    gp <- nodes == g
    y <- z$time[gp]
    stat <- z$death[gp]
    treat <- z$horTh[gp]
    fit <- survfit(Surv(y,stat) ~ treat, conf.type="none")
    if(plotted){
        plot(fit,xlim=xr,mark.time=FALSE,xlab="",ylab="",col=c("blue","red"),lwd=2)
    } else {
        plot(fit,xlim=xr,mark.time=FALSE,xlab="",ylab="Survival probability",
             col=c("blue","red"),lwd=2)
        plotted <- TRUE
    }
    title(paste("Node",g))
    legend("bottomleft",legend=leg.txt,lty=1,col=leg.col,lwd=2)
}
```

Estimated relative risks and survival probabilities The file `ph-gi.fit` gives the terminal node number, observed survival time, event indicator (`y`=uncensored, `n`=censored), log baseline cumulative hazard, survival probability, median survival time, and treatment effect (regression coefficient of treatment indicator) of each observation in the training sample (`cancerdata.txt`). The results for the first few observations are shown below.

```
train  node  observed   event logbasecumhaz  survivalprob  mediansurvtime horTh.yes
    y     3  1814.00     y   -0.335623      0.576131          2286.00   -0.645311
    y     3  2018.00     y   -0.210308      0.720485          2286.00   -0.645311
    y     3  712.000     y    -1.28452      0.894065          2286.00   -0.645311
    y     3  1807.00     y   -0.358191      0.753697          2286.00   -0.645311
    y     3  772.000     y    -1.16232      0.785652          2286.00   -0.645311
    y     2  448.000     y    -2.08322      0.834592          1140.00   -0.114042
    y     3  2172.00     n   -0.121866      0.698971          2286.00   -0.645311
```

11.2.2 Simple linear prognostic control

To reduce or eliminate confounding between treatment and covariate variables, it may be desirable to adjust for the effects of the latter by fitting a regression model that allows for the linear effects of one or more prognostic variables in each node (Loh et al., 2019c). This is done by choosing the "simple linear" or the "multiple linear" option and specifying each potential linear predictor as "`n`" in the description file (no change is needed in `cancerdsc.txt`). First we show how to choose the simple linear model, where a single prognostic variable is used as regressor in each node. There are two options: the **Gi** (default) option is more sensitive to detecting *predictive* variables while the **Gs** option is equally sensitive to detecting *prognostic* variables—see Loh et al. (2015) for definitions.

Input file generation for Gi method

```
 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 (recommended for prediction)
Choose 2 for best simple linear in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple linear, 3: constant ([1:3], <cr>=3): 2
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: cancerdsc.txt
Reading data description file ...
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
D variable is death
Reading data file ...
Number of records in data file: 686
Length of longest entry in data file: 4
Checking for missing values ...
Finished checking
Assigning integer codes to values of 2 categorical variables
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
Smallest uncensored time: 72.0000
Number of cases dropped due to missing D or T or censored T < smallest uncensored T: 14
Number of complete cases excluding censored T < smallest uncensored T: 672
Number of cases used to compute baseline hazard: 672
Number of cases with D=1 and T >= smallest uncensored: 299
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 1
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables ...
Rereading data ...
Largest uncensored and censored time by horTh
   "no"      2456.0000    2563.0000
  "yes"      2372.0000    2659.0000
Proportion of training sample for each level of horTh
 "no"    0.6399
"yes"    0.3601
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals    #X-var   #N-var   #F-var   #S-var
       686          0          0         0        6        0        0
```

```
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        1        0        1
 Survival time variable in column: 9
 Event indicator variable in column: 10
 Proportion uncensored among nonmissing T and D variables: .445
 Number of cases used for training: 672
 Number of split variables: 7
 Number of dummy variables created: 1
 Finished reading data file
 Warning: missing regressor values imputed with node means
 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
 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>=2):
 Input file name: lin-gi.reg
 Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
 Input name of file to store node ID and fitted value of each case: lin-gi.fit
 Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
 Input file name: lin-gi.r
 Input rank of top variable to split root node ([1:9], <cr>=1):
 Input file is created!
 Run GUIDE with the command: guide < lin-gi.in
```

Results for Gi method The following output shows that the pruned tree is trivial with no splits and that the variable `pnodes` is the best simple linear predictor.

```
 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

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

No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 10
Minimum node sample size: 7
Minimum fraction of cases per treatment at each node: 0.072
Number of iterations for fitting: 20
Top-ranked variables and 1-df 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
   :
  19        8   2.741E+05   2.739E+05   2.591E+05   2.918E+00   5.727E-01
  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

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

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

 Cases fit give the number of cases used to fit node
 Deviance is mean residual deviance for all cases in node
        Node    Total    Cases Matrix   Median      Node    Split
        label   cases      fit  rank   survtime   deviance  variable
           1T     672      672     3  1.807E+03  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: Median survival time = 1807.0000

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

Node 1: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 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
 Regressor names and coefficients are stored in lin-gi.reg
 LaTeX code for tree is in lin-gi.tex
 R code is stored in lin-gi.r
```

The file `lin-gi.reg` reports the selected regressor in each terminal node of the tree (there is only one node here):

```
node bestvar
1 pnodes
```

Input file generation for Gs method

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: lin-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: lin-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 (recommended for prediction)
Choose 2 for best simple linear in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple linear, 3: constant ([1:3], <cr>=3): 2
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: cancerdsc.txt
Reading data description file ...
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
D variable is death
Reading data file ...
Number of records in data file: 686
Length of longest entry in data file: 4
Checking for missing values ...
Finished checking
Assigning integer codes to values of 2 categorical variables
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
Smallest uncensored time: 72.0000
Number of cases dropped due to missing D or T or censored T < smallest uncensored T: 14
Number of complete cases excluding censored T < smallest uncensored T: 672
Number of cases used to compute baseline hazard: 672
Number of cases with D=1 and T >= smallest uncensored: 299
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 1
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
```

```
Input your choice: ([1:2], <cr>=2): 1
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
Warning: missing regressor values imputed with node means
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-gs.tex
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>=2):
Input file name: lin-gs.reg
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: lin-gs.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: lin-gs.r
Input rank of top variable to split root node ([1:9], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < lin-gs.in
```

Results for Gs method The Gs method gives a tree with three terminal nodes.

```
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

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

No nodewise interaction tests
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 10
Minimum node sample size: 7
Minimum fraction of cases per treatment at each node: 0.072
Number of iterations for fitting: 20
Top-ranked variables and 1-df chi-squared values at root node
     1  0.2695E+02   pnodes
     2  0.1812E+02   progrec
     3  0.8046E+01   estrec
     4  0.3781E+01   tgrade
     5  0.8274E+00   menostat
     6  0.5154E+00   tsize
     7  0.3349E+00   age

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       45   9.913E+03   9.901E+03   9.080E+03   3.361E+00   7.785E-01
   2       44   9.913E+03   9.901E+03   9.080E+03   3.092E+00   8.253E-01
   :
  19        7   1.956E+00   3.378E-01   2.894E-01   1.630E+00   1.805E-01
  20        4   1.432E+00   6.770E-02   5.670E-02   1.424E+00   7.438E-02
  21**      3   1.336E+00   5.196E-02   3.403E-02   1.289E+00   3.960E-02
  22        2   1.362E+00   5.631E-02   3.638E-02   1.314E+00   5.650E-02
  23        1   1.383E+00   5.502E-02   2.787E-02   1.359E+00   2.776E-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
Deviance is mean residual deviance for all cases in node
       Node    Total    Cases Matrix   Median      Node    Split
       label   cases      fit  rank   survtime   deviance  variable
          1      672      672     3  1.807E+03  1.371E+00  pnodes
          2      370      370     3  2.659E+03+ 1.092E+00  age
          4T     142      142     3  2.563E+03+ 9.548E-01  tsize
          5T     228      228     3  2.030E+03  1.044E+00  tgrade
          3T     302      302     3  9.830E+02  1.552E+00  progrec

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 progrec

Regression tree:

Node 1: pnodes <= 3.5000000
  Node 2: age <= 49.500000
    Node 4: Median survival time = 2563.0000+
  Node 2: age > 49.500000 or NA
    Node 5: Median survival time = 2030.0000
Node 1: pnodes > 3.5000000 or NA
  Node 3: Median survival time = 983.00000

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

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 pnodes <= 3.5000000
 pnodes mean = 4.9866071
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.000
 pnodes       0.5725E-01    8.744        0.000        1.000        4.987        51.00
 horTh.yes   -0.3528       -2.828       0.4823E-02    0.000       0.3601        1.000
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if age <= 49.500000
 age mean = 53.235135
 ----------------------------
Node 4: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      5.162
 age         -0.1344       -5.463       0.2096E-06    21.00        43.00        49.00
 horTh.yes   -0.7981       -1.502       0.1353        0.000       0.1690        1.000
 ----------------------------
Node 5: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     0.3737
 progrec     -0.3152E-02   -2.547       0.1152E-01    0.000        112.1        1490.
 horTh.yes   -0.6723       -2.877       0.4400E-02    0.000       0.4474        1.000
 ----------------------------
Node 3: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      1.039
 progrec     -0.2870E-02   -4.036       0.6925E-04    0.000        105.2        2380.
 horTh.yes   -0.3303       -2.112       0.3549E-01    0.000       0.3841        1.000
 ----------------------------
 Observed and fitted values are stored in lin-gs.fit
 Regressor names and coefficients are stored in lin-gs.reg
 LaTeX code for tree is in lin-gs.tex
 R code is stored in lin-gs.r
```

The tree is shown in Figure 27. It does not display the linear predictor selected at each terminal node. This information is given in the file `lin-gs.out` or, more conveniently, in tabular form in `lin-gs.reg` as shown below.

```
node bestvar
```

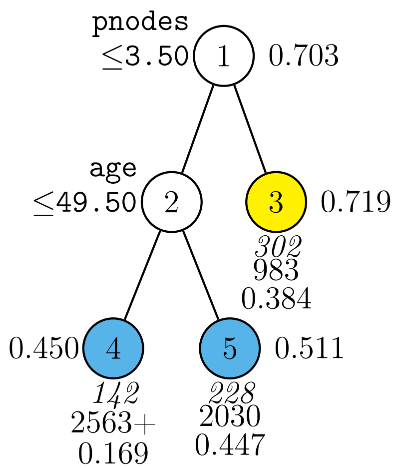

Figure 27: GUIDE v.41.2 0.250-SE proportional hazards regression tree using Gs option for `time` and event indicator `death` with adjustment for simple linear prognostic effects (missing regressor values imputed with node means). At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*), median survival time, and proportion of `horTh` = `yes` printed below nodes. Treatment `horTh` hazard ratio of level `yes` to `no` beside nodes. Terminal nodes with treatment hazard ratio above and below 0.703 (ratio at root node) are colored yellow and skyblue respectively. Second best split variable at root node is `progrec`.

```
4 age
5 progrec
3 progrec
```

11.3 Censored response: restricted mean

Besides a proportional hazards tree, GUIDE can also fit a tree to estimate the restricted mean survival time in each node (Chen and Tsiatis, 2001; Tian et al., 2014). This section shows how this is carried out. The time restriction may be changed by the user during when the input file is created.

11.3.1 Without linear prognostic control

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

Input file generation for Gi method

```
 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 for prediction)
 Choose 1 for multiple regression
 Choose 2 for best simple polynomial in one N or F variable
 Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
 0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
 4: best simple stepwise ANCOVA ([0:4], <cr>=3):
 Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
 Input name of data description file (max 100 characters);
```

```
enclose with matching quotes if it has spaces: cancerdsc.txt
Reading data description file ...
Training sample file: cancerdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
6 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Reading data file ...
Number of records in data file: 686
Length of longest entry in data file: 4
Checking for missing values ...
Finished checking
Assigning integer codes to values of 2 categorical variables
Treatment (R) variable is horTh with values "no" and "yes"
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 1
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables ...
Rereading data ...
Largest uncensored and censored time by horTh
   "no"      2456.0000    2563.0000
  "yes"      2372.0000    2659.0000
Smallest observed uncensored time is 72.0000
Largest observed censored or uncensored time is 2659.0000
Input restriction on event time ([72.00:2659.00], <cr>=1222.00):
Proportion of training sample for each level of horTh
 "no"    0.6360
"yes"    0.3640
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       686         0          0        0        0        0        6
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0        1        0        1
No weight variable in data file
Number of cases used for training: 533
Number of split variables: 7
Number of dummy variables created: 1
```

```
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): rest-gi.tex
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: rest-gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: rest-gi.r
Input rank of top variable to split root node ([1:9], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < rest-gi.in
```

Results for Gi method

```
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

Constant fitted to cases with missing values in regressor variables
Predictive priority (Gi) using restricted mean event time
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 10
Minimum node sample size: 6
Minimum fraction of cases per treatment at each node: 0.073
Top-ranked variables and 1-df 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       55   5.137E+05   2.795E+04   1.644E+04   5.090E+05   2.925E+04
    2       54   5.137E+05   2.795E+04   1.644E+04   5.090E+05   2.925E+04
    :
   34        5   5.035E+05   2.647E+04   1.048E+04   5.064E+05   1.206E+04
   35+       2   4.463E+05   2.216E+04   1.042E+04   4.353E+05   2.040E+04
   36**      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 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.

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

Results for Gs method The piecewise-constant Gs tree has one split, as shown below.

```
 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

Constant fitted to cases with missing values in regressor variables
Prognostic priority (Gs) using restricted mean event time
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 10
Minimum node sample size: 6
Minimum fraction of cases per treatment at each node: 0.073
Top-ranked variables and 1-df chi-squared values at root node
     1  0.4966E+02   pnodes
     2  0.3191E+02   progrec
     3  0.2229E+02   estrec
     4  0.1276E+02   tgrade
     5  0.6795E+01   tsize
     6  0.4436E+00   age
```

```
     7  0.1645E+00   menostat

Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1       58   4.781E+05   2.651E+04   2.824E+04   4.735E+05   3.191E+04
   2       57   4.781E+05   2.651E+04   2.824E+04   4.735E+05   3.191E+04
   :
  38        3   4.236E+05   2.187E+04   1.596E+04   4.273E+05   3.065E+04
  39**      2   3.798E+05   1.852E+04   1.523E+04   3.804E+05   1.576E+04
  40        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 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      533      533     2  9.873E+02  1.519E+05  0.0106 pnodes
          2T     332      332     2  1.073E+03  1.048E+05  0.0129 estrec
          3T     201      201     2  8.312E+02  1.842E+05  0.0174 progrec

Number of terminal nodes of final tree: 2
Total number of nodes of final tree: 3
Second best split variable (based on curvature test) at root node is progrec

Regression tree:

Node 1: pnodes <= 4.5000000
  Node 2: terminal
Node 1: pnodes > 4.5000000 or NA
  Node 3: terminal

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

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 pnodes <= 4.5000000
 pnodes mean = 4.8475943
 Coefficients of least squares regression function:
 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
 ----------------------------
Node 2: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      1050.        55.19       0.2220E-15
 horTh.yes     66.83        2.074       0.3884E-01    0.000       0.3483        1.000
 time mean = 1072.91
 No truncation of predicted values
 ----------------------------
Node 3: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      790.8        22.68        0.000
 horTh.yes     106.5        1.879       0.6164E-01    0.000       0.3786        1.000
 time mean = 831.171
 No truncation of predicted values
 ----------------------------
 Observed and fitted values are stored in rest-gs.fit
 LaTeX code for tree is in rest-gs.tex
 R code is stored in rest-gs.r
```

11.3.2 With linear prognostic control

A trivial tree is obtained for both the Gi and Gs methods if a linear regressor is included in each node.

12 Nonrandomized treatments: RHC data

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

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

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

12.1 Proportional hazards

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

12.1.1 Gi option

Gi input file creation

```
 0. Read the warranty disclaimer
```

```
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: surv-gi.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: surv-gi.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1):
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple linear in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best 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
23 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Re-checking data ...
Allocating missing value information ...
```

```
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
Smallest uncensored survtime: 2.0000
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 1
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime by swang1
 "NoRHC"      1867.0000    1243.0000
   "RHC"      1943.0000    1351.0000
Proportion of training sample for each level of swang1
"NoRHC"    0.6192
  "RHC"    0.3808
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735         0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
Survival time variable in column: 64
Event indicator variable in column: 9
Proportion uncensored among nonmissing T and D variables: .649
Number of cases used for training: 5735
Number of split variables: 53
Number of dummy variables created: 1
Number of cases excluded due to 0 W or missing D, T or R variables: 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
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: surv.gi.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: surv-gi.r
```

```
Input rank of top variable to split root node ([1:55], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < surv-gi.in
```

Contents of surv-gi.out

```
Regression tree for censored response
Pruning by cross-validation
Data description file: rhcdsc3.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
23 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Piecewise linear model
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
Number of dummy variables created: 1
Smallest uncensored survtime: 2.0000
Largest uncensored and censored survtime by swang1
  swang1     Uncensored     Censored
 "NoRHC"      1867.0000    1243.0000
   "RHC"      1943.0000    1351.0000
Proportion of training sample for each level of swang1
"NoRHC"    0.6192
  "RHC"    0.3808

Summary information for training sample of size 5735
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight,
t=survival time variable
                                                  #Codes/
                                                  Levels/
 Column  Name             Minimum      Maximum    Periods   #Missing
```

```
      2  cat1       c                            9
      3  cat2       c                            6     4535
      4  ca         c                            3
      9  death      d    0.000       1.000
     10  cardiohx   c                            2
      :
     59  adld3p     s    0.000       7.000                 4296
     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       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
        0        0        0       30        0        1
 Survival time variable in column: 64
 Event indicator variable in column: 9
 Proportion uncensored among nonmissing T and D variables: 0.649
 Number of cases used for training: 5735
 Number of split variables: 53
 Number of dummy variables created: 1
 Number of cases excluded due to 0 W or missing D, T or R variables: 0

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

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

```
    36  0.3221E-04   meanbp1

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

 0-SE tree based on mean is marked with * and has 5 terminal nodes
 0-SE tree based on median is marked with + and has 5 terminal nodes
 Selected-SE tree based on mean using naive SE is marked with **
 Selected-SE tree based on mean using bootstrap SE is marked with --
 Selected-SE tree based on median and bootstrap SE is marked with ++
 * tree, ** tree, + tree, and ++ tree all the same

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

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

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

 Number of terminal nodes of final tree: 5
 Total number of nodes of final tree: 9
 Second best split variable (based on curvature test) at root node is resp1

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

 Node 1: ph1 <= 7.3344730
   Node 2: cat2 = "MOSF w/Sepsis", "NA"
     Node 4: Median survival time = 157.00000
```

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

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

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

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

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

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

Node 1: Intermediate node
 A case goes into Node 2 if ph1 <= 7.3344730
 ph1 mean = 7.3884135
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean      Maximum
 Constant       0.000
 swang1.RHC    0.1504       4.494       0.7131E-05    0.000       0.3808        1.000
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if cat2 = "MOSF w/Sepsis", "NA"
 cat2 mode = "NA"
 ----------------------------
Node 4: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean      Maximum
 Constant     -0.6181E-01
 swang1.RHC    0.4067       6.034       0.2086E-08    0.000       0.4499        1.000
 ----------------------------
Node 5: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
```

```
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.8005
 swang1.RHC   -0.3295       -1.558       0.1223        0.000       0.3558         1.000
 ----------------------------
Node 3: Intermediate node
 A case goes into Node 6 if resp1 <= 38.500000 or NA
 resp1 mean = 28.418652
 ----------------------------
Node 6: Intermediate node
 A case goes into Node 12 if paco21 <= 29.498050
 paco21 mean = 36.054906
 ----------------------------
Node 12: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      0.3006
 swang1.RHC   -0.3237E-01  -0.3424       0.7322        0.000       0.3916         1.000
 ----------------------------
Node 13: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -0.7105E-01
 swang1.RHC    0.5937E-02   0.1159       0.9078        0.000       0.3632         1.000
 ----------------------------
Node 7: Terminal node
 Coefficients of log-relative hazard function (relative to baseline hazard):
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant     -0.1150E-01
 swang1.RHC    0.3555        4.329       0.1651E-04    0.000       0.3316         1.000
 ----------------------------
 Observed and fitted values are stored in surv.gi.fit
 LaTeX code for tree is in surv-gi.tex
 R code is stored in surv-gi.r
```

Figure 28 shows the tree and Figure 29 shows the estimated survival curves in its terminal nodes. The R code for making the plots is given below.

```
library(survival)
z0 <- read.table("rhcdata.txt",header=TRUE)
par(mar=c(3,4,3,1),mfrow=c(2,3),cex=1)
leg.txt <- c("NoRHC","RHC"); leg.col <- c("blue","red"); leg.lty <- 2:1
xr <- range(z0$survtime)
zg <- read.table("surv-gi.fit",header=TRUE)
nodes <- zg$node
uniq.gp <- unique(sort(nodes))
```

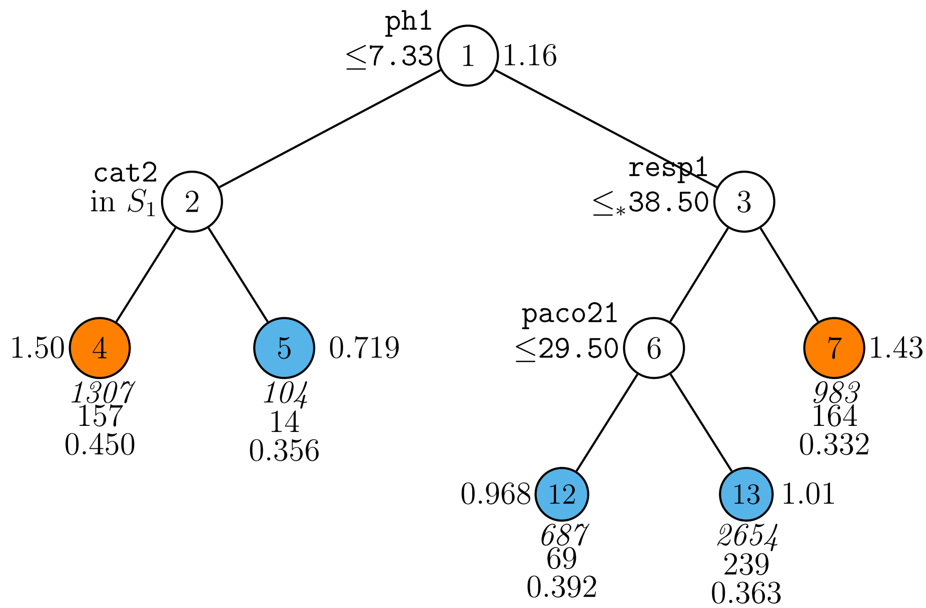

Figure 28: GUIDE v.41.1 0.250-SE proportional hazards regression tree using Gi option for `survtime` and event indicator `death` without adjustment for linear prognostic effects. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. S_1 = {`MOSF w/Sepsis`, `NA`}. Treatment `swang1` hazard ratio of level `RHC` to level `NoRHC` beside nodes. Sample size (in *italics*), median survival time, and proportion of `swang1` = `RHC` printed below nodes. Terminal nodes with treatment hazard ratio above and below 1.162 (ratio at root node) are colored orange and skyblue respectively. Second best split variable at root node is `resp1`.

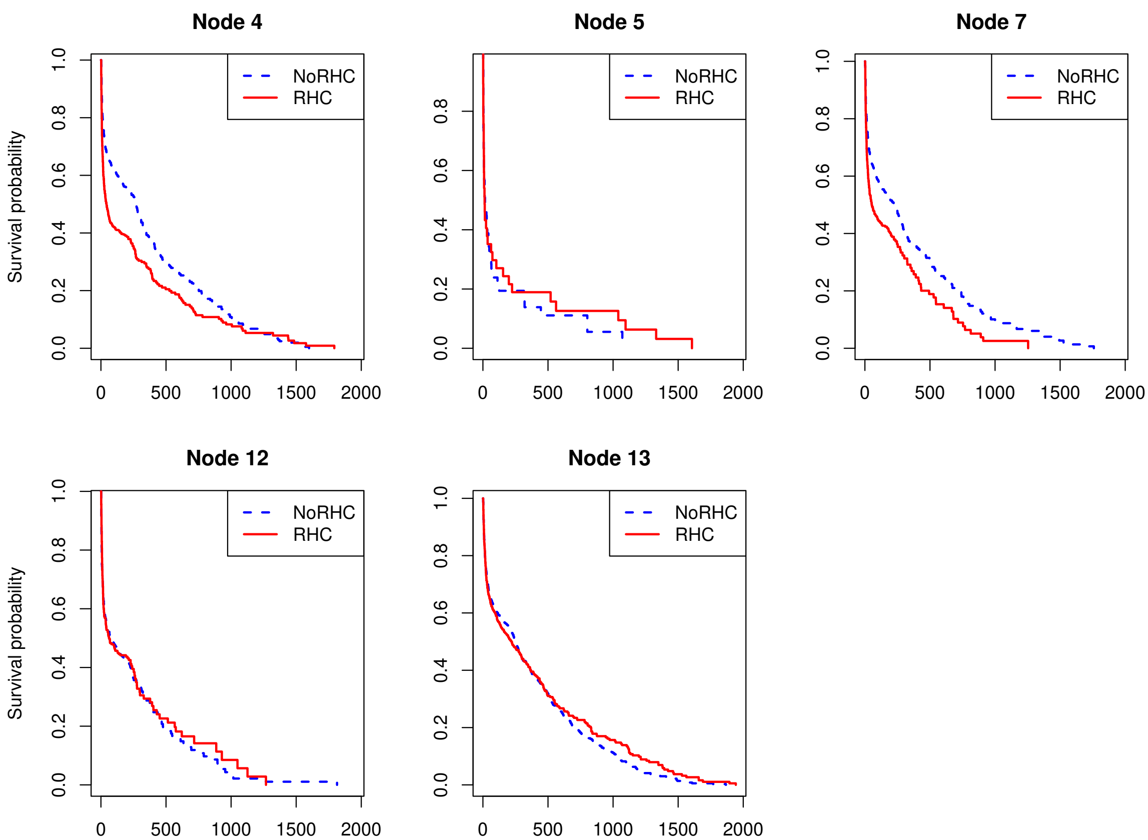

Figure 29: Survival curves for RHC data in nodes of Figure 28

```
ii <- 0
for(g in uniq.gp){
    ii <- ii+1
    gp <- nodes == g
    y <- z0$survtime[gp]
    stat <- z0$death[gp]
    treat <- z0$swang1[gp]
    fit <- survfit(Surv(y,stat) ~ treat, conf.type="none")
    if(g == 4 | g == 12){
        plot(fit,xlim=xr,mark.time=FALSE,xlab="",ylab="Survival probability",
             col=leg.col,lwd=2,lty=leg.lty)
    } else {
        plot(fit,xlim=xr,mark.time=FALSE,xlab="",ylab="",col=leg.col,lwd=2,lty=leg.lty)
    }
    title(paste("Node",g))
    legend("topright",legend=leg.txt,lty=leg.lty,col=leg.col,lwd=2)
}
```

Following are the top 3 lines of the file `surv-gi.fit`

```
train  node  observed  event logbasecumhaz   survivalprob  mediansurvtime  swang1.RHC
    y    13  240.000     n   -0.269165        0.490850          239.000    0.593672E-002
    y     4  45.0000     y   -0.757608        0.515901          157.000    0.406690
    y     7  317.000     n   -0.633003E-001   0.266047          164.000    0.355517
```

The column definitions are

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

`node`: terminal node label of observation.

`observed`: observed survival time t.

`event`: `y` if uncensored (death), `n` if censored.

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

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

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

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

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

swang1.RHC: estimated treatment effect β_1 for level RHC of swang1.

12.2 Restricted mean

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

12.2.1 Gi option

Gi input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: rest-gi.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: rest-gi.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 4
Input 1 for proportional hazards, 2 for restricted mean event time ([1:2], <cr>=1): 2
Choose complexity of model to use at each node:
Choose 0 for stepwise linear regression (recommended for prediction)
Choose 1 for multiple regression
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
0: stepwise linear, 1: multiple linear, 2: best simple polynomial, 3: constant,
4: best simple stepwise 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
23 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 1
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime by swang1
 "NoRHC"      1867.0000    1243.0000
   "RHC"      1943.0000    1351.0000
Smallest observed uncensored time is 2.0000
Largest observed censored or uncensored time is 1943.0000
Input restriction on event time ([2.00:1943.00], <cr>=622.00):
Proportion of training sample for each level of swang1
"NoRHC"    0.5993
  "RHC"    0.4007
     Total  #cases w/   #missing
```

```
     #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       5735          0       5157        8        0        0       23
     #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
          0        0        0       30        0        1
 No weight variable in data file
 Number of cases used for training: 3763
 Number of split variables: 53
 Number of dummy variables created: 1
 Number of cases excluded due to 0 W or missing D or R variables: 1972
 Finished reading data file
 Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
 Input file name to store LaTeX code (use .tex as suffix): rest-gi.tex
 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: rest-gi.fit
 Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
 Input file name: rest-gi.r
 Input rank of top variable to split root node ([1:55], <cr>=1):
 Input file is created!
 Run GUIDE with the command: guide < rest-gi.in
```

Contents of rest-gi.out

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

```
Largest uncensored and censored survtime by swang1
  swang1     Uncensored     Censored
 "NoRHC"      1867.0000    1243.0000
   "RHC"      1943.0000    1351.0000
Interval for restricted mean event time is from 0 to 622.
Proportion of training sample for each level of swang1
"NoRHC"    0.5993
  "RHC"    0.4007

Summary information for training sample of size 3763 (excluding observations with
non-positive weight or missing values in d, e, t, r or z variables)
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                  #Codes/
                                                  Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      2  cat1        c                                 9
      3  cat2        c                                 6     2836
      4  ca          c                                 3
      9  death       d    0.000        1.000
      :
     61  race        c                                 3
     62  income      c                                 4
     64  survtime    t    2.000        1943.
  =================== Constructed variables ===================
     65  swang1.RHC  f    0.000        1.000

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

Predictive priority (Gi) using restricted mean event time
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

No nodewise interaction tests
```

```
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 13
Minimum node sample size: 7
Minimum fraction of cases per treatment at each node: 0.080
Top-ranked variables and 1-df chi-squared values at root node
     1  0.9407E+01   scoma1
     2  0.7887E+01   ph1
     3  0.7551E+01   pafi1
     :
    37  0.1688E-01   meanbp1
    38  0.4169E-02   cat1

Size and CV MSE and SE of subtrees:
 Tree   #Tnodes  Mean MSE   SE(Mean)    BSE(Mean)  Median MSE  BSE(Median)
   1      325   1.644E+05   5.598E+03   3.914E+03   1.652E+05   6.405E+03
   2      324   1.644E+05   5.598E+03   3.914E+03   1.652E+05   6.405E+03
   3      323   1.644E+05   5.598E+03   3.914E+03   1.652E+05   6.403E+03
   :
 217        3   1.295E+05   4.444E+03   4.786E+03   1.294E+05   6.909E+03
 218**      2   1.157E+05   3.411E+03   2.378E+03   1.141E+05   3.229E+03
 219        1   1.198E+05   3.143E+03   9.972E+02   1.190E+05   1.421E+03

0-SE tree based on mean is marked with * and has 2 terminal nodes
0-SE tree based on median is marked with + and has 2 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

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

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

D-mean is weighted mean of death in the node
Cases fit give the number of cases used to fit node
MSE and R^2 are based on all cases in node
       Node    Total    Cases Matrix    Node      Node      Node  Split           Other
       label   cases      fit  rank    D-mean     MSE       R^2   variable        variables
          1     3763     3763     2  2.583E+02  9.489E+04  0.0043 scoma1
          2T    3124     3124     2  2.781E+02  9.938E+04  0.0075 pafi1
          3T     639      639     2  1.333E+02  4.975E+04  0.0016 sod1

Number of terminal nodes of final tree: 2
Total number of nodes of final tree: 3
Second best split variable (based on curvature test) at root node is ph1
```

```
 Regression tree:

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

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

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

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

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

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

Node 1: Intermediate node
 A case goes into Node 2 if scoma1 <= 49.500000
 scoma1 mean = 20.462797
 Coefficients of least squares regression function:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       271.2        52.27        0.000
 swang1.RHC    -33.80       -4.020       0.5926E-04    0.000       0.3808        1.000
 survtime mean = 258.284
 No truncation of predicted values
 ----------------------------
Node 2: Terminal node
 Coefficients of least squares regression functions:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       295.7        51.17        0.000
 swang1.RHC    -44.75       -4.866       0.1195E-05    0.000       0.3949        1.000
 survtime mean = 278.051
 No truncation of predicted values
 ----------------------------
Node 3: Terminal node
 Coefficients of least squares regression functions:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant       138.4        14.56        0.000
 swang1.RHC    -17.66       -1.003       0.3161        0.000       0.2916        1.000
 survtime mean = 133.272
```

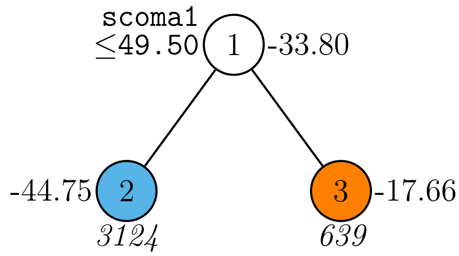

Figure 30: GUIDE v.41.1 0.250-SE regression tree using Gi option for mean `survtime` restricted to less than 622.00 without adjustment for linear prognostic effects. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *(italics)*) printed below nodes. Treatment `swang1` effects (relative to reference level `NoRHC`) of levels `RHC` (relative to `NoRHC`) beside nodes. Terminal nodes with treatment effect above and below -33.80 (effect at root node) are colored orange and skyblue respectively. Second best split variable at root node is `ph1`.

```
No truncation of predicted values
----------------------------
Number of times Li-Martin approximation used = 394
Observed and fitted values are stored in rest-gi.fit
LaTeX code for tree is in rest-gi.tex
R code is stored in rest-gi.r
```

Figure 30 shows the Gi restricted mean event time tree.

13 Multiresponse: NMES data

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

Table 10: Definitions of variables in NMES data

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

used in Section 14.

The data file `nmes.txt` contains observations on 4406 subjects from a National Medical Expenditure Survey (NMES) conducted in 1987 and 1988. Table 10 gives the names of the variables and their definitions. The data were previously analyzed in Deb and Trivedi (1997), Cameron and Trivedi (1998, chap. 6), and Zeileis (2006). Here we construct a regression tree to predict the outcomes for the first 6 variables (`ofp`, `ofnp`, `opp`, `opnp`, `emer`, and `hosp`). The contents of the description file `nmes.dsc` follow.

```
nmes.txt
NA
1
1 ofp d
2 ofnp d
3 opp d
4 opnp d
5 emer d
6 hosp d
```

```
7 health c
8 numchron n
9 adldiff c
10 region c
11 age n
12 black c
13 gender c
14 married c
15 school n
16 faminc n
17 employed c
18 privins c
19 medicaid c
```

13.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: mult.in
 Input 1 for model fitting, 2 for importance or DIF scoring,
       3 for data conversion ([1:3], <cr>=1):
 Name of batch output file: mult.out
 Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
 Input 1 for classification, 2 for regression, 3 for propensity score grouping
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
 Input choice ([1:7], <cr>=1): 5
 Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
 Input name of data description file (max 100 characters);
 enclose with matching quotes if it has spaces: nmes.dsc
 Reading data description file ...
 Training sample file: nmes.txt
 Missing value code: NA
 Records in data file start on line 1
 4 N variables changed to S
 Number of D variables: 6
 D variables are:
 ofp
 ofnp
 opp
```

```
opnp
emer
hosp
Multivariate or univariate split variable selection:
Choose multivariate if there is an order among the D variables;
choose univariate otherwise or if item response
Input 1 for multivariate, 2 for univariate ([1:2], <cr>=1): 2
D variables can be normalized to have unit variance,
e.g., if they have different scales or units
Input 1 to normalize D variables, 2 for no normalization ([1:2], <cr>=1):
Input 1 for equal, 2 for unequal weighting of D variables ([1:2], <cr>=1):
Reading data file ...
Number of records in data file: 4406
Length of longest entry in data file: 9
Checking for missing values ...
Finished checking
Assigning integer codes to values of 9 categorical variables
Re-checking data ...
Assigning codes to missing values, if any ...
Data checks complete
Normalizing data
Rereading data ...
PCA can be used for variable selection
Do not use PCA if differential item functioning (DIF) scores are wanted
Input 1 to use PCA, 2 otherwise ([1:2], <cr>=2):
#cases w/ miss. D = number of cases with all D values missing
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4406          0          0        0        0        0        4
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        9        0
Number of cases used for training: 4406
Number of split variables: 13
Finished reading data file
Input 0=skip LaTeX tree, 1=tree without node numbers, 2=with node numbers ([0:2], <cr>=2):
Input file name to store LaTeX code (use .tex as suffix): mult.tex
You can store the variables and/or values used to split and fit in a file
Choose 1 to skip this step, 2 to store split and fit variables,
3 to store split variables and their values
Input your choice ([1:3], <cr>=1):
Input 2 to save node IDs of 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>=2):
Input file name: mult.r
```

```
Input rank of top variable to split root node ([1:13], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < mult.in
```

13.2 Contents of mult.out

```
Multi-response or longitudinal data without T variables
Pruning by cross-validation
Data description file: nmes.dsc
Training sample file: nmes.txt
Missing value code: NA
Records in data file start on line 1
4 N variables changed to S
Number of D variables: 6
Univariate split variable selection method
Mean-squared errors (MSE) are calculated from normalized D variables
D variables equally weighted
Piecewise constant model
Number of records in data file: 4406
Length of longest entry in data file: 9
Model fitted to subset of observations with complete D values
Neither LDA nor PCA used

Summary information for training sample of size 4406
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                #Codes/
                                                Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      1  ofp        d    0.000        89.00
      2  ofnp       d    0.000        104.0
      3  opp        d    0.000        141.0
      4  opnp       d    0.000        155.0
      5  emer       d    0.000        12.00
      6  hosp       d    0.000        8.000
      7  health     c                                  3
      8  numchron   s    0.000        8.000
      9  adldiff    c                                  2
     10  region     c                                  4
     11  age        s    6.600        10.90
     12  black      c                                  2
     13  gender     c                                  2
```

```
     14  married    c                                  2
     15  school     s    0.000       18.00
     16  faminc     s   -1.012       54.84
     17  employed   c                                  2
     18  privins    c                                  2
     19  medicaid   c                                  2

 #cases w/ miss. D = number of cases with all D values missing
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      4406         0          0        0        0        0        4
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        9        0
 Number of cases used for training: 4406
 Number of split variables: 13

 Constant fitted to cases with missing values in regressor variables
 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: 0.2500

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

 Size and CV Loss and SE of subtrees:
  Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
    1       15   1.593E+03   1.075E+02   4.586E+02   1.393E+03   9.190E+02
    2       14   1.593E+03   1.075E+02   4.586E+02   1.393E+03   9.190E+02
    3       13   1.593E+03   1.075E+02   4.586E+02   1.393E+03   9.190E+02
    4       12   1.593E+03   1.075E+02   4.586E+02   1.393E+03   9.190E+02
```

```
   5       11   1.593E+03   1.075E+02   4.586E+02   1.393E+03   9.190E+02
   6       10   1.593E+03   1.075E+02   4.586E+02   1.393E+03   9.190E+02
   7        9   1.593E+03   1.075E+02   4.586E+02   1.393E+03   9.191E+02
   8        8   1.593E+03   1.075E+02   4.586E+02   1.393E+03   9.191E+02
   9        7   1.031E+03   9.120E+01   4.321E+02   2.019E+00   6.997E+02
  10        6   1.031E+03   9.120E+01   4.321E+02   2.019E+00   6.997E+02
  11        5   3.330E+02   7.592E+01   3.178E+02   1.211E+00   2.139E-01
  12        4   1.123E+00   1.298E-01   1.460E-01   9.148E-01   9.960E-02
  13**      3   1.123E+00   1.298E-01   1.460E-01   9.148E-01   9.960E-02
  14        2   1.259E+00   1.296E-01   1.461E-01   1.068E+00   9.920E-02
  15        1   1.635E+00   1.308E-01   1.448E-01   1.421E+00   1.078E-01

0-SE tree based on mean is marked with * and has 3 terminal nodes
0-SE tree based on median is marked with + and has 3 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

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

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

Cases fit give the number of cases used to fit node
MSE is residual sum of squares divided by number of cases in node
       Node    Total    Cases      Node    Split
      label    cases      fit       MSE    variable
          1     4406     4406  1.000E+00   numchron
          2T    2523     2523  5.688E-01   numchron
          3     1883     1883  1.528E+00   health
          6T     426      426  2.282E+00   -
          7T    1457     1457  1.277E+00   privins

Number of terminal nodes of final tree: 3
Total number of nodes of final tree: 5
Second best split variable (based on curvature test) at root node is health

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

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

```
     Node 7: Mean cost = 1.2765754

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

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

The tree is shown in Figure 31. The file `mult.fit` saves the mean values of the dependent variables in each terminal node:

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

The file `mult.nid` gives the terminal node number for each observation, including

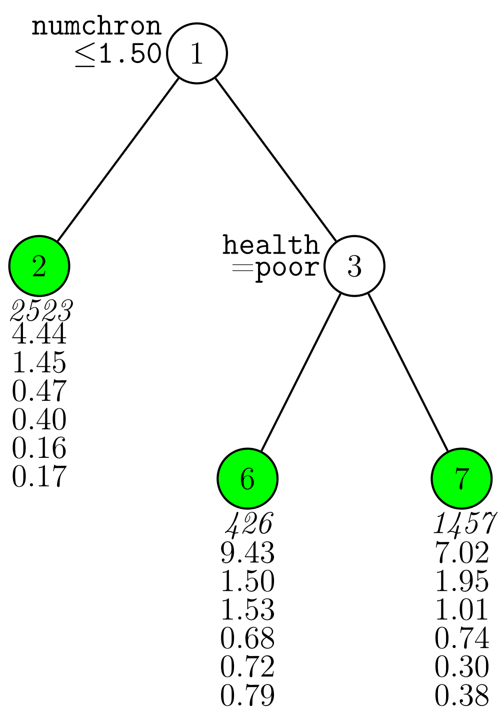

Figure 31: GUIDE v.41.1 0.250-SE regression tree for predicting response variables `ofp`, `ofnp`, `opp`, `opnp`, `emer`, and `hosp`, without using PCA at each node. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*) and predicted values of `ofp`, `ofnp`, `opp`, `opnp`, `emer`, and `hosp` printed below nodes. Second best split variable at root node is `health`.

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

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
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):
 Input name of data description file (max 100 characters);
 enclose with matching quotes if it has spaces: wagedsc.txt
 Reading data description file ...
 Training sample file: wagedat.txt
 Missing value code: NA
 Records in data file start on line 1
 One N variable changed to S
 Number of D variables: 13
 D variables are:
 wage1
 wage2
 wage3
 wage4
 wage5
 wage6
 wage7
 wage8
 wage9
```

```
wage10
wage11
wage12
wage13
T variables are:
exper1
exper2
exper3
exper4
exper5
exper6
exper7
exper8
exper9
exper10
exper11
exper12
exper13
D variables can be grouped into segments to look for patterns
Input 1 for equal-sized groups, 2 for custom groups ([1:2], <cr>=1):
Input number of roughly equal-sized groups ([2:9], <cr>=3):
Input number of interpolating points for prediction ([10:100], <cr>=31):
Reading data file ...
Number of records in data file: 888
Length of longest entry in data file: 16
Checking for missing values ...
Finished checking
Missing values found in D variables
Assigning integer codes to values of 1 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
#cases w/ miss. D = number of cases with all D values missing
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       888          0          0       40        0        0        1
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        1        0
Number of cases used for training: 888
Number of split variables: 2
Number of cases excluded due to 0 W or missing D variable: 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): wage.tex
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>=2):
Input file name: wage.r
Input rank of top variable to split root node ([1:2], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < wage.in
```

14.2 Contents of wage.out

```
Longitudinal data with T variables
Lowess smoothing
Pruning by cross-validation
Data description file: wagedsc.txt
Training sample file: wagedat.txt
Missing value code: NA
Records in data file start on line 1
One N variable changed to S
Number of D variables: 13
Number of D variables: 13
D variables are:
wage1
wage2
wage3
wage4
wage5
wage6
wage7
wage8
wage9
wage10
wage11
wage12
wage13
T variables are:
exper1
```

```
exper2
exper3
exper4
exper5
exper6
exper7
exper8
exper9
exper10
exper11
exper12
exper13
Number of records in data file: 888
Length of longest entry in data file: 16
Missing values found in D variables
Model fitted to subset of observations with complete D values

Summary information for training sample of size 888
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                #Codes/
                                                Levels/
 Column  Name            Minimum      Maximum    Periods   #Missing
      2  hgc        s    6.000        12.00
      3  exper1     t   0.1000E-02    5.637
      4  exper2     t    0.000        7.584                    38
      5  exper3     t    0.000        9.777                    77
      6  exper4     t    0.000        10.81                   124
      7  exper5     t    0.000        11.78                   159
      8  exper6     t    0.000        10.59                   233
      9  exper7     t    0.000        11.28                   325
     10  exper8     t    0.000        10.58                   428
     11  exper9     t    0.000        11.62                   551
     12  exper10    t    0.000        12.26                   678
     13  exper11    t    0.000        11.98                   791
     14  exper12    t    0.000        12.56                   856
     15  exper13    t    0.000        12.70                   882
     29  wage1      d    2.030        68.65
     30  wage2      d    2.069        50.40                    38
     31  wage3      d    2.046        34.50                    77
     32  wage4      d    2.117        33.15                   124
     33  wage5      d    2.104        49.30                   159
     34  wage6      d    2.208        74.00                   233
     35  wage7      d    2.104        47.28                   325
```

```
     36  wage8      d    2.316        37.71                  428
     37  wage9      d    2.529        46.11                  551
     38  wage10     d    2.998        56.54                  678
     39  wage11     d    4.084        22.20                  791
     40  wage12     d    3.432        46.20                  856
     41  wage13     d    4.563        7.776                  882
     68  race       c                                 3

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
       888         0          0       40        0        0        1
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        1        0
 Number of cases used for training: 888
 Number of split variables: 2
 Number of cases excluded due to 0 W or missing D variable: 0

 Constant fitted to cases with missing values in regressor variables
 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: 0.2500

 No nodewise interaction tests
 Split values for N and S variables based on exhaustive search
 Maximum number of split levels: 10
 Minimum node sample size: 44
 Top-ranked variables and 1-df 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

 WARNING: tree based on mean CV estimate of error has no splits
 Choosing smallest nontrivial tree with no larger CV error 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
          3T      311      311  1.513E+02   race

 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 race

 Regression tree for longitudinal data:

 Node 1: hgc <= 9.5000000
   Node 2: Mean cost = 103.80991
 Node 1: hgc > 9.5000000 or NA
   Node 3: Mean cost = 150.79730

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

Node 1: Intermediate node
 A case goes into Node 2 if hgc <= 9.5000000
 hgc mean = 8.9166667
 ----------------------------
Node 2: Terminal node
 ----------------------------
Node 3: 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 32 shows the tree and Figure 33 plots lowess-smoothed curves of mean wage in the two terminal nodes. The figure is produced by the following R code.

```
z <- read.table("wagedat.txt",header=FALSE)
```

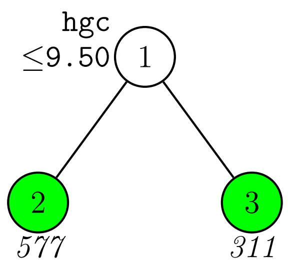

Figure 32: GUIDE v.41.1 0.053-SE (0.250-SE has no splits) regression tree for predicting longitudinal variables `wage1`, `wage2`, etc. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size *(in italics)* printed below nodes. Second best split variable at root node is `race`.

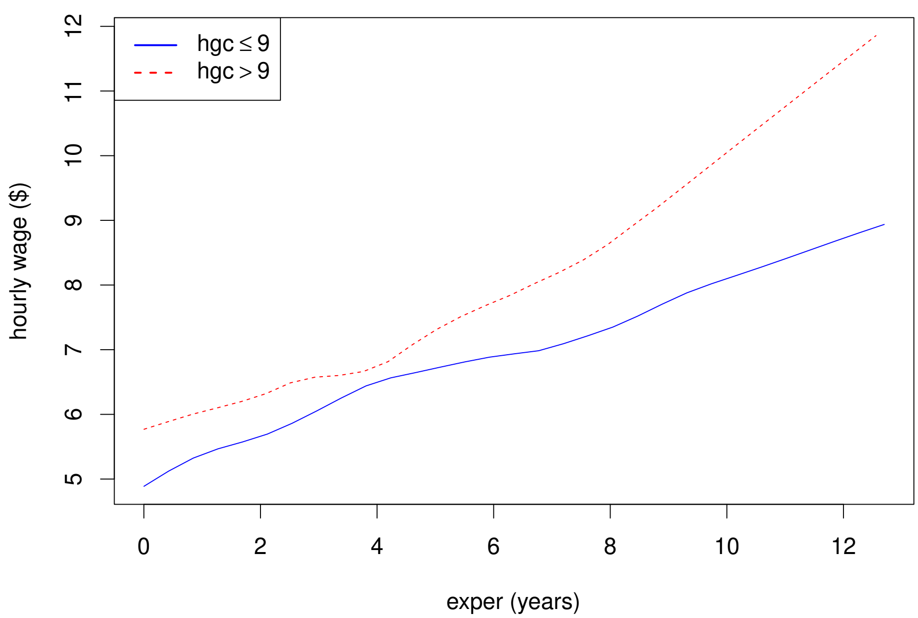

Figure 33: Lowess-smoothed mean wage curves in the terminal nodes of Figure 32.

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

guide.fit <- read.table("wage.fit",header=TRUE)
g.node <- guide.fit$node
g.start <- guide.fit$t.start
g.end <- guide.fit$t.end
n <- length(g.node)
m <- dim(guide.fit)[2]
npts <- m-3  # number of time points for plotting

xvals <- guide.fit[,2:3]
xvals <- as.numeric(unlist(xvals))
yvals <- guide.fit[,4:m]
yvals <- as.numeric(unlist(yvals))
plot(range(xvals),range(yvals),type="n",xlab="exper (years)",ylab="hourly wage ($)")
leg.col <- c("blue","red")
leg.lty <- c(1,2)
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("hgc" > 9)))
legend("topleft",legend=leg.txt,lty=leg.lty,col=leg.col,lwd=2)
```

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

```
 node t.start t.end fitted1 fitted2 fitted3 fitted4 fitted5 fitted6 fitted7 fitted8 fitted9 fitted10
    2   0.10000E-02  0.12700E+02  0.48875E+01  0.51221E+01  0.53241E+01  0.54668E+01  0.55738E+01  0
    3   0.20000E-02  0.12558E+02  0.57699E+01  0.58884E+01  0.60035E+01  0.60997E+01  0.61994E+01  0
```

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. If the node is terminal, the 5th column contains the letter "`t`"; otherwise if it is non-terminal, the 5th column is an integer indicating the number of split values to follow (a split on a `c` variable may have more than one value). In the example below, node 1 is split on `s` variable `hgc` at value 9.50. Nodes 2 and 3 are terminal nodes; each would be split on `race` if they were not terminal.

```
1 s hgc hgc        1   0.9500000000E+01
2 t race race   t
3 t race race   t
```

15 Logistic regression

If the dependent variable Y takes values 0 and 1, GUIDE can construct a tree model such that a simple or multiple linear logistic regression model is fitted in each node. The tree model may be more efficient (in terms of size and prediction accuracy) if a preliminary estimate of $p = P(Y = 1)$ is available. The preliminary estimate of p is not necessary, but it may be easily obtained by fitting a GUIDE forest or kernel discriminant model to the data. If a variable containing the estimated p values are included in the data, it should be specifed as an "`e`" variable in the description file (see Section 3.1). Missing values in the predictor variables used in the logistic regression node models are imputed with node means; see Loh (2021) for more details.

We use the NHTSA data to demonstrate this, with $Y =$ `HIC2`, which takes value 1 if `HIC` > 999 and 0 otherwise. The description file is `nhtsadsc2.txt`. The "`e`" variable is `estHIC2` which is a column of estimated values of $p = P(Y = 1)$ obtained from GUIDE forest.

15.1 Piecewise constant

15.1.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: logitc.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: logitc.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 7
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best 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: nhtsadsc2.txt
Reading data description file ...
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
48 N variables changed to S
Warning: B variables changed to C
D variable is HIC2
Reading data file ...
Number of records in data file: 3310
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 13 categorical variables
Finished assigning codes to 10 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
```

```
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310         34       3310        1        0        0       48
    #P-var   #M-var   #B-var   #C-var   #I-var
         6       42        0       13        0
Number of cases used for training: 3276
Number of split variables: 61
Number of cases excluded due to 0 W or missing D variable: 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): logitc.tex
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: logitc.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: logitc.r
Input rank of top variable to split root node ([1:67], <cr>=1):
Input file is created!
```

15.1.2 Contents of logitc.out

```
Binary logistic regression tree
Pruning by cross-validation
Data description file: nhtsadsc2.txt
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
48 N variables changed to S
Warning: B variables changed to C
D variable is HIC2
Piecewise constant 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

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
Levels of M variables are for missing values in associated variables
                                                 #Codes/
                                                 Levels/
 Column  Name           Minimum      Maximum    Periods   #Missing
      1  BARRIG     c                                 3
      2  BARSHP     c                                21
      3  BARANG     p    0.000        330.0         360        14
      4  BARDIA     s    191.0        1000.                  2807
      5  OCCWT      s    72.00        83.00                  3265
      6  OCCWT_     m                                 2
      :
    104  VEHSPD     s   0.3000        99.10                     6
    105  VEHSPD_    m                                 2
    106  CRBANG     p    0.000        315.0         360        24
    107  PDOF       p    0.000        345.0         360        23
    108  CARANG     p    0.000        99.00         360       991
    109  VEHOR      p    0.000        90.00         360       995
    110  RSTFRT     c                                 3
    111  HIC2       d    0.000        1.000
    112  estHIC2    e    0.000       0.8455

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals    #X-var   #N-var   #F-var   #S-var
      3310         34       3310         1        0        0       48
    #P-var   #M-var   #B-var   #C-var   #I-var
         6       42        0       13        0
Number of cases used for training: 3276
Number of split variables: 61
Number of cases excluded due to 0 W or missing D variable: 34
Proportion of ones in HIC2 variable: 0.084554

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500
```

```
Nodewise interaction tests on all variables
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 13
Minimum node sample size: 65
Minimum number of D=0 and D=1 in each node: 9
Top-ranked variables and 1-df chi-squared values at root node
     1  0.1218E+04   COLMEC
     2  0.9001E+03   YEAR
     3  0.8714E+03   MODELD
     4  0.7917E+03   RSTFRT
     5  0.6935E+03   HS
     6  0.5377E+03   HR
     7  0.3959E+03   CS
     :
    65  0.1349E+00   KB
    66  0.4871E-01   HB

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1--      8   4.534E-01   2.045E-02   7.571E-03   4.498E-01   7.411E-03
   2        7   4.599E-01   2.042E-02   6.132E-03   4.516E-01   7.028E-03
   3        5   4.580E-01   2.012E-02   6.699E-03   4.516E-01   7.863E-03
   4        4   4.580E-01   2.012E-02   6.699E-03   4.516E-01   7.863E-03
   5        3   4.580E-01   2.012E-02   6.699E-03   4.516E-01   7.863E-03
   6**      2   4.580E-01   2.012E-02   6.699E-03   4.516E-01   7.863E-03
   7        1   5.795E-01   2.316E-02   2.216E-03   5.834E-01   3.465E-03

0-SE tree based on mean is marked with * and has 8 terminal nodes
0-SE tree based on median is marked with + and has 8 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree same as + tree
** tree same as ++ tree
* tree same as -- tree

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     1  8.455E-02  5.797E-01  COLMEC
         2T    2610     2610     1  2.797E-02  2.553E-01  MODELD
         3T     666      666     1  3.063E-01  1.234E+00  MODELD

 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 YEAR

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

 Node 1: COLMEC = "BWU", "CYL", "NA", "NAP", "UNK"
   Node 2: HIC2 proportion of 1s = 0.27969349E-1
 Node 1: COLMEC /= "BWU", "CYL", "NA", "NAP", "UNK"
   Node 3: HIC2 proportion of 1s = 0.30630631

 ***************************************************************
 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 COLMEC = "BWU", "CYL", "NA", "NAP", "UNK"
 COLMEC mode = "UNK"
 Coefficients of logit function
 Regressor    Coefficient  t-stat      p-value
 Constant     -2.382        17.39        0.000
 Proportion of ones in variable HIC2 = 0.845543E-1
 ----------------------------
Node 2: Terminal node
 Coefficients of logit:
 Regressor    Coefficient  t-stat      p-value
 Constant     -3.548        8.666        0.000
 Proportion of ones in variable HIC2 = 0.279693E-1
 ----------------------------
Node 3: Terminal node
 Coefficients of logit:
```

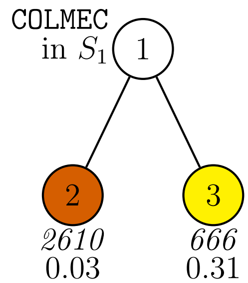

Figure 34: GUIDE v.41.1 0.250-SE piecewise-constant logistic regression tree for predicting P(`HIC2=1`). At each split, an observation goes to the left branch if and only if the condition is satisfied. S_1 = {`BWU`, `CYL`, `NA`, `NAP`, `UNK`}. Sample size (in *italics*) and proportion of 1s in `HIC2` 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. Second best split variable at root node is `YEAR`.

```
Regressor    Coefficient  t-stat      p-value
Constant    -0.8174        17.15        0.000
Proportion of ones in variable HIC2 = 0.306306
----------------------------
Observed and fitted values are stored in logitc.fit
LaTeX code for tree is in logitc.tex
R code is stored in logitc.r
```

The logistic regression tree is shown in Figure 34.

15.2 Simple linear

We can also construct a logistic regression tree with a simple linear logistic regression model fitted to each node.

15.2.1 Input file creation

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

```
Input 1 for classification, 2 for regression, 3 for propensity score grouping
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
 7=binary logistic regression.
Input choice ([1:7], <cr>=1): 7
Choose complexity of model to use at each node:
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best simple polynomial, 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: nhtsadsc2.txt
Reading data description file ...
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
Warning: B variables changed to C
D variable is HIC2
Reading data file ...
Number of records in data file: 3310
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 13 categorical variables
Finished assigning codes to 10 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310         34       3310        1       48        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         6       42        0       13        0
Number of cases used for training: 3276
Number of split variables: 61
```

```
Number of cases excluded due to 0 W or missing D variable: 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
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>=2):
Input file name: logits.reg
Input 2 to save fitted values and node IDs, 1 otherwise ([1:2], <cr>=2):
Input name of file to store node ID and fitted value of each case: logits.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: logits.r
Input rank of top variable to split root node ([1:67], <cr>=1):
Input file is created!
```

15.2.2 Contents of `logits.out`

```
Binary logistic regression tree
Pruning by cross-validation
Data description file: nhtsadsc2.txt
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
Warning: B variables changed to C
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

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
Levels of M variables are for missing values in associated variables
```

```
                                                      #Codes/
                                                      Levels/
 Column  Name           Minimum      Maximum    Periods   #Missing
      1  BARRIG     c                                 3
      2  BARSHP     c                                21
      3  BARANG     p    0.000       330.0          360        14
      4  BARDIA     n  1.9100E+02    1000.                    2807
      5  OCCWT      n  7.2000E+01    83.00                    3265
      6  OCCWT_     m                                 2
      7  DUMSIZ     c                                 7
      8  HH         n  5.8000E+01    4321.                     150
      9  HH_        m                                 2
      :
    104  VEHSPD     n  3.0000E-01    99.10                       6
    105  VEHSPD_    m                                 2
    106  CRBANG     p    0.000       315.0          360        24
    107  PDOF       p    0.000       345.0          360        23
    108  CARANG     p    0.000       99.00          360       991
    109  VEHOR      p    0.000       90.00          360       995
    110  RSTFRT     c                                 3
    111  HIC2       d    0.000        1.000
    112  estHIC2    e    0.000       0.8455

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310         34       3310        1       48        0        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         6       42        0       13        0
Number of cases used for training: 3276
Number of split variables: 61
Number of cases excluded due to 0 W or missing D variable: 34
Proportion of ones in HIC2 variable: 0.084554

Constant fitted to cases with missing values in regressor variables
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 13
Minimum node sample size: 65
Minimum number of D=0 and D=1 in each node: 9
Top-ranked variables and 1-df chi-squared values at root node
     1  0.4911E+03   RSTFRT
     2  0.4567E+03   MODELD
```

```
     3  0.3172E+03   IMPANG
     4  0.2900E+03   COLMEC
     5  0.2769E+03   BARDIA
     6  0.2617E+03   BARSHP
     :
    65  0.8221E+00   CARANG
    66  0.5257E+00   WHLBAS

 Size and CV Loss and SE of subtrees:
  Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
    1       11   6.281E-01   2.156E-02   6.402E-02   5.496E-01   7.028E-02
    2       10   6.509E-01   2.157E-02   6.659E-02   6.092E-01   9.245E-02
    3        8   6.836E-01   2.124E-02   6.729E-02   6.446E-01   1.075E-01
    4        7   6.786E-01   2.106E-02   6.970E-02   5.956E-01   1.181E-01
    5        6   6.774E-01   2.091E-02   6.991E-02   5.896E-01   1.185E-01
    6        5   6.765E-01   2.087E-02   7.003E-02   5.896E-01   1.194E-01
    7        3   7.436E-01   1.937E-02   8.730E-02   6.943E-01   1.526E-01
    8        2   4.547E-01   1.932E-02   9.157E-03   4.653E-01   1.100E-02
    9**      1   4.547E-01   1.932E-02   9.157E-03   4.653E-01   1.100E-02

 0-SE tree based on mean is marked with * and has 1 terminal node
 0-SE tree based on median is marked with + and has 1 terminal node
 Selected-SE tree based on mean using naive SE is marked with **
 Selected-SE tree based on mean using bootstrap SE is marked with --
 Selected-SE tree based on median and bootstrap SE is marked with ++
 * tree, ** tree, + tree, and ++ tree all the same

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

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

 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
           1T    3276     3276     2  8.455E-02  4.546E-01 RSTFRT   -YEAR
 Best split at root node is on RSTFRT

 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 RSTFRT

 Regression tree:

 Node 1: HIC2 proportion of 1s = 0.84554335E-1
```

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

Node 1: Terminal node
 Coefficients of logit:
 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.
 If regressors have missing values, predicted value = 0.84554335E-1
 ----------------------------
 Observed and fitted values are stored in logits.fit
 Regressor names and coefficients are stored in logits.reg
 LaTeX code for tree is in logits.tex
 R code is stored in logits.r
```

The results show that the tree has no splits. It fits a simple linear logistic regression model to the whole data set with `YEAR` as linear predictor. If the value of `YEAR` is missing, the predicted value of p is the mean of `HIC2`.

15.3 Multiple linear

This section shows how to fit a multiple logistic regression to each node of the tree.

15.3.1 Input file creation

```
 0. Read the warranty disclaimer
 1. Create a GUIDE input file
 Input your choice: 1
 Name of batch input file: logitm.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: logitm.out
 Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
 Input 1 for classification, 2 for regression, 3 for propensity score grouping
 Input your choice ([1:3], <cr>=1): 2
 Choose type of regression model:
  1=linear, 2=quantile, 3=Poisson, 4=censored response,
  5=multiresponse or itemresponse, 6=longitudinal data (with T variables),
  7=binary logistic regression.
 Input choice ([1:7], <cr>=1): 7
 Choose complexity of model to use at each node:
```

```
Choose 1 for multiple regression (recommended for prediction)
Choose 2 for best simple polynomial in one N or F variable
Choose 3 for constant fit (recommended for interpretability or if there is an R variable)
1: multiple linear, 2: best 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: nhtsadsc2.txt
Reading data description file ...
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
D variable is HIC2
Reading data file ...
Number of records in data file: 3310
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 13 categorical variables
Finished assigning codes to 10 categorical variables
Associating missing values of N, P and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
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: 20
Creating dummy variables ...
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       3310        1       48       48        0
    #P-var   #M-var   #B-var   #C-var   #I-var
         6       42        1       12        0
Number of cases used for training: 3276
Number of split variables: 61
Number of dummy variables created: 20
Number of cases excluded due to 0 W or missing D variable: 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): logitm.tex
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: logitm.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: logitm.r
Input rank of top variable to split root node ([1:135], <cr>=1):
Input file is created!
```

15.3.2 Results

```
Binary logistic regression tree
Pruning by cross-validation
Data description file: nhtsadsc2.txt
Training sample file: nhtsadatam.txt
Missing value code: NA
Records in data file start on line 2
D variable is HIC2
Piecewise multiple 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
Number of dummy variables created: 20

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
Levels of M variables are for missing values in associated variables
                                                   #Codes/
                                                   Levels/
 Column  Name               Minimum      Maximum    Periods   #Missing
      1  BARRIG         c                                 3
      2  BARSHP         b                                21
      3  BARANG         p    0.000        330.0         360       14
```

```
      4  BARDIA         n   1.9100E+02    1000.                     2807
      5  OCCWT          n   7.2000E+01    83.00                     3265
      6  OCCWT_         m                                     2
      :
    106  CRBANG         p    0.000        315.0             360       24
    107  PDOF           p    0.000        345.0             360       23
    108  CARANG         p    0.000        99.00             360      991
    109  VEHOR          p    0.000        90.00             360      995
    110  RSTFRT         c                                     3
    111  HIC2           d    0.000        1.000
    112  estHIC2        e    0.000       0.8455
  =================== Constructed variables ===================
    113  BARSHP.134     f    0.000        1.000
    114  BARSHP.488     f    0.000        1.000
    115  BARSHP.EOB     f    0.000        1.000
    116  BARSHP.EOL     f    0.000        1.000
    117  BARSHP.FAB     f    0.000        1.000
    118  BARSHP.FLB     f    0.000        1.000
    119  BARSHP.GRL     f    0.000        1.000
    120  BARSHP.IAT     f    0.000        1.000
    121  BARSHP.LCB     f    0.000        1.000
    122  BARSHP.LUM     f    0.000        1.000
    123  BARSHP.MBR     f    0.000        1.000
    124  BARSHP.OTH     f    0.000        1.000
    125  BARSHP.PDU     f    0.000        1.000
    126  BARSHP.POL     f    0.000        1.000
    127  BARSHP.ROR     f    0.000        1.000
    128  BARSHP.SGN     f    0.000        1.000
    129  BARSHP.UNK     f    0.000        1.000
    130  BARSHP.US1     f    0.000        1.000
    131  BARSHP.US2     f    0.000        1.000
    132  BARSHP.US3     f    0.000        1.000
    133  BARDIA.NA      f    0.000        1.000
    134  OCCWT.NA       f    0.000        1.000
      :
    178  BX20.NA        f    0.000        1.000
    179  BX21.NA        f    0.000        1.000
    180  VEHSPD.NA      f    0.000        1.000

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3310         34       3310        1       48       48        0
    #P-var   #M-var   #B-var   #C-var   #I-var
        6       42        1       12        0
 Number of cases used for training: 3276
 Number of split variables: 61
```

```
Number of dummy variables created: 20
Number of cases excluded due to 0 W or missing D variable: 34
Proportion of ones in HIC2 variable: 0.084554

Missing regressors imputed with means and missing-value indicators added
Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: 0.2500

Nodewise interaction tests on all variables
Fraction of cases used for splitting each node: 1.0000
Maximum number of split levels: 13
Minimum node sample size: 63
Minimum number of D=0 and D=1 in each node: 9
150 bootstrap calibration replicates
Scaling for N variables after bootstrap calibration: 1.000
Top-ranked variables and 1-df chi-squared values at root node
     1  0.7106E+03   MODELD
     2  0.6311E+03   COLMEC
     3  0.4562E+03   YEAR
     4  0.4439E+03   RSTFRT
     :
    63  0.3634E+01   OFFSET
    64  0.1315E+01   CRBANG
    65  0.3776E-01   IMPANG
    66  0.6323E-02   BARDIA

Size and CV Loss and SE of subtrees:
 Tree   #Tnodes  Mean Loss   SE(Mean)   BSE(Mean)  Median Loss  BSE(Median)
   1       14   1.798+308   1.798+308   1.798+308    Infinity   1.798+308
   2       13   1.798+308   1.798+308   1.798+308    Infinity   1.798+308
   3       12   1.798+308   1.798+308   1.798+308    Infinity   1.798+308
   4        8   1.798+308   1.798+308   1.798+308    Infinity   1.798+308
   5        7   1.798+308   1.798+308   1.798+308    Infinity   1.798+308
   6        6   1.798+308   1.798+308   1.798+308    Infinity   1.798+308
   7        4   1.798+308   1.798+308   1.798+308    Infinity   1.798+308
   8        2   1.798+308   1.798+308   1.798+308   5.013E-01   1.798+308
   9**      1   1.798+308   1.798+308   1.798+308   5.386E-01   1.798+308

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 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
           1T    3276     3276   108  8.455E-02  5.509E-01  COLMEC
 Best split at root node is on COLMEC

 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 COLMEC

 Regression tree:

 Node 1: HIC2 proportion of 1s = 0.84554335E-1

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

Node 1: Terminal node
 Coefficients of logit:
 Regressor     Coefficient  t-stat      p-value      Minimum         Mean       Maximum
 Constant      -8.116      -0.6531E-01   0.9479
 BARDIA         0.8790E-03  0.3432       0.7315        191.0        284.6        1000.
 OCCWT          0.2460      0.5460       0.5851        72.00        77.73        83.00
 HH             0.1349E-02  0.9004       0.3680        58.00        360.7        4321.
 HW            -0.3223E-03 -0.2249       0.8221        163.0        601.1        6355.
 HR             0.2768E-03  0.2094       0.8341        38.00        223.2        2801.
 HS            -0.2249E-02 -0.9436       0.3454        94.00        309.8        3051.
 CD             0.1583E-03  0.8754E-01   0.9303        241.0        529.7        5857.
 CS            -0.1346E-03 -0.7103E-01   0.9434        41.00        295.7        4077.
 AD             0.1287E-03  0.8821E-01   0.9297        10.00        124.4        7625.
 HD             0.3573E-03  0.1616       0.8717        20.00        162.5        1000.
 KD            -0.3990E-03 -0.1847       0.8535        2.000        139.8        315.0
 HB            -0.5234E-01 -0.4176       0.6763        83.00        800.7        1000.
 NB             0.1794      0.5362       0.5919        185.0        917.7        1000.
 CB             0.6487E-01  0.5378       0.5908        200.0        863.0        1000.
 KB            -0.1336     -0.5746       0.5656       -10.00        8.139        1000.
 CLSSPD         0.1903E-01  0.5326       0.5944        19.60        50.37        99.10
 OFFSET        -0.2703E-03 -0.4019       0.6878       -1054.       -24.73        900.0
 IMPPNT        -0.1379E-03 -0.1875       0.8513       -690.0        22.60        1739.
```

```
 YEAR         -0.3630E-01   -1.812       0.7001E-01    1972.         2000.         2017.
 ENGDSP        0.000         0.000        1.000       0.6000         2.940         99.90
 VEHTWT        0.000         0.000        1.000        728.0         1758.         0.2342E+05
 CURBWT        0.1303E-03    0.2514       0.8015       964.0         1704.         3096.
 WHLBAS        0.000         0.000        1.000        299.0         2755.         0.1000E+05
 VEHLEN       -0.3087E-04   -0.1183       0.9059       547.0         4708.         0.1125E+05
 VEHWID       -0.2450E-03   -0.4661       0.6412       185.0         1781.         5835.
 VEHCG         0.000         0.000        1.000        109.0         1246.         3435.
 BX1           0.000         0.000        1.000        481.0         4713.         0.2540E+05
 BX2          -0.1913E-03   -0.5081       0.6114       260.0         4062.         0.1073E+05
 BX3           0.2324E-03    0.3746       0.7080       496.0         3628.         0.1000E+06
 BX4           0.1566E-03    0.3952       0.6927       1715.         3271.         9500.
 BX5          -0.3331E-03   -0.3123       0.7549       80.00         3224.         7764.
 BX6           0.8214E-03    0.6416       0.5212       339.0         3217.         9487.
 BX7          -0.4450E-03   -0.3873       0.6986       153.0         3207.         7613.
 BX8           0.1878E-03    0.1459       0.8840       155.0         2151.         8583.
 BX9           0.6013E-04    0.3879E-01   0.9691       161.0         2149.         7677.
 BX10          0.2516E-03    0.1553       0.8766       167.0         2165.         8580.
 BX11         -0.3446E-04   -0.3161E-01   0.9748       25.00         2165.         7538.
 BX12         -0.9530E-04   -0.9348E-01   0.9255       214.0         3213.         9469.
 BX13         -0.2188E-03   -0.1696       0.8653       80.00         3214.         9469.
 BX14          0.6497E-05    0.5840E-01   0.9534       289.0         3561.         0.4000E+05
 BX15         -0.1788E-03   -0.2931       0.7695       341.0         3552.         9911.
 BX16         -0.4620E-04   -0.6165E-01   0.9508       10.00         2772.         9279.
 BX17         -0.3180E-03   -0.6487       0.5166       75.00         408.9         0.1085E+05
 BX18          0.1084E-03    0.2331       0.8157       10.00         468.4         0.1083E+05
 BX19         -0.2192E-04   -0.2132       0.8312       33.00         4520.         0.4230E+05
 BX20         -0.7397E-04   -0.2273       0.8202       356.0         4505.         0.1088E+05
 BX21          0.2424E-04    0.7224E-01   0.9424       54.00         499.6         0.1085E+05
 VEHSPD        0.1352E-02    0.4181E-01   0.9667      0.3000         50.30         99.10
 BARSHP.134    0.9360E-01    0.8664E-01   0.9310       0.000        0.2625E-01     1.000
 BARSHP.488   -0.2838       -0.8350E-01   0.9335       0.000        0.6105E-03     1.000
 BARSHP.EOB   -0.3344       -0.2943       0.7685       0.000        0.4915E-01     1.000
 BARSHP.EOL   -0.4358       -0.1692       0.8657       0.000        0.9158E-03     1.000
 BARSHP.FAB   -0.1791       -0.1480       0.8824       0.000        0.1160E-01     1.000
 BARSHP.FLB   -0.1395E-01   -0.1368E-01   0.9891       0.000        0.3126         1.000
 BARSHP.GRL    -2.054        -1.106       0.2688       0.000        0.2747E-02     1.000
 BARSHP.IAT    -2.103        -1.292       0.1966       0.000        0.5800E-02     1.000
 BARSHP.LCB    0.1818        0.1816       0.8559       0.000        0.4173         1.000
 BARSHP.LUM     1.979         1.070       0.2848       0.000        0.4884E-02     1.000
 BARSHP.MBR    -1.862       -0.5974       0.5503       0.000        0.6105E-03     1.000
 BARSHP.OTH   -0.2325       -0.1745       0.8615       0.000        0.6716E-02     1.000
 BARSHP.PDU    0.6018E-01    0.3244E-01   0.9741       0.000        0.2137E-02     1.000
 BARSHP.POL     1.468         1.008       0.3136       0.000        0.1371         1.000
 BARSHP.ROR   -0.8234       -0.2690       0.7879       0.000        0.6105E-03     1.000
 BARSHP.SGN    -2.242       -0.9906       0.3220       0.000        0.1526E-02     1.000
```

BARSHP.UNK	-0.3980	-0.1177	0.9063	0.000	0.1221E-02	1.000
BARSHP.US1	-0.2868	-0.1619	0.8714	0.000	0.2442E-02	1.000
BARSHP.US2	0.6299E-01	0.4168E-01	0.9668	0.000	0.4579E-02	1.000
BARSHP.US3	-0.5868E-01	-0.4326E-01	0.9655	0.000	0.6105E-02	1.000
BARDIA.NA	0.8465	0.8689	0.3849	0.000	0.8568	1.000
OCCWT.NA	0.3094	0.1923	0.8475	0.000	0.9966	1.000
HH.NA	0.9053	0.8528	0.3938	0.000	0.4579E-01	1.000
HW.NA	-0.5159	-0.3919	0.6951	0.000	0.3816E-01	1.000
HR.NA	-0.1275	-0.7099E-01	0.9434	0.000	0.5830E-01	1.000
HS.NA	0.8096E-01	0.6311E-01	0.9497	0.000	0.6105E-01	1.000
CD.NA	0.1951	0.4839	0.6285	0.000	0.1300	1.000
CS.NA	-0.8660	-0.8299	0.4067	0.000	0.4518E-01	1.000
AD.NA	-1.063	-0.6113	0.5410	0.000	0.5830E-01	1.000
HD.NA	1.360	0.7498	0.4535	0.000	0.6044E-01	1.000
KD.NA	0.1147	0.1530	0.8784	0.000	0.4976E-01	1.000
HB.NA	31.23	0.4179	0.6761	0.000	0.9936	1.000
NB.NA	-133.6	-0.5371	0.5912	0.000	0.9945	1.000
CB.NA	-17.77	-0.6137	0.5395	0.000	0.9939	1.000
KB.NA	0.9033	0.4730	0.6362	0.000	0.4014	1.000
CLSSPD.NA	18.63	0.4698	0.6386	0.000	0.6105E-03	1.000
OFFSET.NA	-0.1018	-0.4149	0.6783	0.000	0.1401	1.000
IMPPNT.NA	0.6058E-01	0.1270	0.8989	0.000	0.5168	1.000
YEAR.NA	1.150	0.5182	0.6044	0.000	0.1221E-02	1.000
ENGDSP.NA	0.8412E-01	0.1364	0.8915	0.000	0.1709E-01	1.000
VEHTWT.NA	0.1163	0.6647E-01	0.9470	0.000	0.1832E-02	1.000
CURBWT.NA	-0.9897E-01	-0.3568	0.7213	0.000	0.8712	1.000
WHLBAS.NA	0.3689	0.4035	0.6866	0.000	0.1282E-01	1.000
VEHLEN.NA	-0.2320	-0.1352	0.8925	0.000	0.4579E-02	1.000
VEHWID.NA	-0.2375	-0.5213	0.6022	0.000	0.3388E-01	1.000
VEHCG.NA	-0.1312	-0.2109	0.8330	0.000	0.5098E-01	1.000
BX1.NA	-0.7540	-0.4347	0.6638	0.000	0.1624	1.000
BX2.NA	0.7244	0.2546	0.7990	0.000	0.2900	1.000
BX10.NA	25.47	0.4125	0.6800	0.000	0.2888	1.000
BX11.NA	0.000	0.000	1.000	0.000	0.2891	1.000
BX12.NA	0.000	0.000	1.000	0.000	0.2888	1.000
BX13.NA	0.000	0.000	1.000	0.000	0.2888	1.000
BX14.NA	0.1146	0.2853E-01	0.9772	0.000	0.2891	1.000
BX15.NA	0.8082	0.3816	0.7028	0.000	0.2903	1.000
BX16.NA	-0.8212E-01	-0.3352E-01	0.9733	0.000	0.2845	1.000
BX17.NA	-3.485	-0.5043	0.6141	0.000	0.2894	1.000
BX18.NA	0.6419	0.4455	0.6560	0.000	0.2845	1.000
BX19.NA	0.2841	0.6351E-01	0.9494	0.000	0.1648	1.000
BX20.NA	0.2063	0.4209E-01	0.9664	0.000	0.1645	1.000
BX21.NA	0.4847	0.4262	0.6700	0.000	0.2943	1.000
BX3.NA	-0.2333E-01	-0.1048E-01	0.9916	0.000	0.2903	1.000
BX4.NA	0.6271	0.1868	0.8518	0.000	0.2897	1.000

```
BX5.NA       -0.7012      -0.1743       0.8616        0.000        0.2894        1.000
BX6.NA        -1.194      -0.3422       0.7322        0.000        0.2894        1.000
BX7.NA        -23.38      -0.3854       0.6999        0.000        0.2891        1.000
BX8.NA        0.3424       0.8482E-01   0.9324        0.000        0.2891        1.000
BX9.NA         0.000        0.000        1.000        0.000        0.2891        1.000
VEHSPD.NA    -0.3514      -0.1538       0.8777        0.000        0.1832E-02    1.000
Proportion of ones in variable HIC2 = 0.845543E-1
----------------------------
Observed and fitted values are stored in logitm.fit
LaTeX code for tree is in logitm.tex
R code is stored in logitm.r
```

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 thresholds for grouping the variables by their importance—see Loh and Zhou (2021).

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
23 N variables changed to S
D variable is swang1
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 2
Assigning integer codes to values of 30 categorical variables
Finished assigning codes to 10 categorical variables
```

```
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
Creating missing value indicators ...
Rereading data ...
Class  #Cases     Proportion
NoRHC    3551     0.61918047
RHC      2184     0.38081953
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157       10        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       30        0
Number of cases used for training: 5735
Number of split variables: 53
Number of cases excluded due to 0 W or missing D variable: 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 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):
You can create a description file with the selected variables included or excluded
Input 2 to create such a file, 1 otherwise ([1:2], <cr>=1):
You can also output the importance scores and variable names to a file
Input 1 to create such a file, 2 otherwise ([1:2], <cr>=1):
Input file name: imp.scr
Input file is created!
Run GUIDE with the command: guide < imp.in
```

16.1.2 Contents of `imp.out`

The most interesting part of the output file is at the end, as shown below.

```
Scaled importance scores of predictor variables
     Score     Rank  Variable
```

```
 2.375E+01     1.00  aps1
 2.319E+01     2.00  cat1
 2.154E+01     3.00  crea1
 2.091E+01     4.00  pafi1
 1.869E+01     5.00  meanbp1
 1.249E+01     6.00  neuro
 1.139E+01     7.00  alb1
 1.043E+01     8.00  card
 1.043E+01     9.00  cat2
 1.032E+01    10.00  hema1
 9.504E+00    11.00  wtkilo1
 8.182E+00    12.00  adld3p
 8.131E+00    13.00  seps
 6.673E+00    14.00  dnr1
 6.567E+00    15.00  bili1
 6.348E+00    16.00  resp
 5.646E+00    17.00  paco21
 5.555E+00    18.00  surv2md1
 4.160E+00    19.00  transhx
 3.957E+00    20.00  chrpulhx
 3.889E+00    21.00  hrt1
 3.821E+00    22.00  resp1
 3.571E+00    23.00  ph1
 3.423E+00    24.00  ninsclas
 3.183E+00    25.00  dementhx
 2.400E+00    26.00  das2d3pc
 2.316E+00    27.00  psychhx
 2.118E+00    28.00  gastr
 2.083E+00    29.00  renal
 1.799E+00    30.00  cardiohx
 1.744E+00    31.00  income
 1.454E+00    32.00  urin1
--------- variables above this line have scores above 99% threshold (A) -----
 1.336E+00    33.00  trauma
 1.224E+00    34.00  age
 1.168E+00    35.00  sex
 1.166E+00    36.00  edu
 1.165E+00    37.00  sod1
 1.038E+00    38.00  wblc1
--------- variables above this line have scores above 95% threshold (B) -----
 9.625E-01    39.00  immunhx
 8.970E-01    40.00  malighx
--------- variables above this line have scores above 90% threshold (C) -----
 8.870E-01    41.00  ca
--------- variables above this line have scores above 80% threshold (D) -----
 8.316E-01    42.00  scoma1
```

```
  8.198E-01    43.00  amihx
  6.724E-01    44.00  chfhx
  6.284E-01    45.00  gibledhx
  4.206E-01    46.00  pot1
  4.156E-01    47.00  ortho
  3.967E-01    48.00  renalhx
  3.695E-01    49.00  hema
  3.531E-01    50.00  liverhx
  3.119E-01    51.00  meta
  2.838E-01    52.00  temp1
  1.268E-01    53.00  race

 99% threshold is 1.3948
 95% threshold is 1.0000
 90% threshold is 0.8920
 80% threshold is 0.8593
 Number of variables above 99% threshold is 32
 Number of variables between 95% and 99% thresholds is 6
 Number of variables between 90% and 95% thresholds is 2
 Number of variables between 80% and 90% thresholds is 1
 LaTeX code for tree is in imp.tex
 Importance scores are stored in imp.scr
```

The variables, sorted according to their importance scores, are divided into 5 groups:

A. Scores above 99% threshold

B. Scores above 95% threshold and below 99% threshold

C. Scores above 90% threshold and below 95% threshold

D. Scores above 80% threshold and below 90% threshold

E. below 80% threshold

The groups and thesholds have the following interpretation. Let H_0 denote the null hypothesis H_0 that the dependent variable is independent of the predictor variables (it is not assumed that the predictor variables are independent of each other). If H_0 is true, there is a 0.01, 0.05, 0.10, and 0.20 probability that one or more predictor variables falls into groups A, $A \cup B$, $A \cup B \cup C$, $A \cup B \cup C \cup D$, respectively. The importance scores are normalized so that the 95% threshold is 1.0.

The file `imp.scr` lists the group memberships, scores and variables names:

```
Rank Type      Score       Variable
   1  A     2.375E+01     aps1
   2  A     2.319E+01     cat1
   3  A     2.154E+01     crea1
   4  A     2.091E+01     pafi1
   5  A     1.869E+01     meanbp1
   6  A     1.249E+01     neuro
   7  A     1.139E+01     alb1
   8  A     1.043E+01     card
   9  A     1.043E+01     cat2
  10  A     1.032E+01     hema1
  11  A     9.504E+00     wtkilo1
  12  A     8.182E+00     adld3p
  13  A     8.131E+00     seps
  14  A     6.673E+00     dnr1
  15  A     6.567E+00     bili1
  16  A     6.348E+00     resp
  17  A     5.646E+00     paco21
  18  A     5.555E+00     surv2md1
  19  A     4.160E+00     transhx
  20  A     3.957E+00     chrpulhx
  21  A     3.889E+00     hrt1
  22  A     3.821E+00     resp1
  23  A     3.571E+00     ph1
  24  A     3.423E+00     ninsclas
  25  A     3.183E+00     dementhx
  26  A     2.400E+00     das2d3pc
  27  A     2.316E+00     psychhx
  28  A     2.118E+00     gastr
  29  A     2.083E+00     renal
  30  A     1.799E+00     cardiohx
  31  A     1.744E+00     income
  32  A     1.454E+00     urin1
  33  B     1.336E+00     trauma
  34  B     1.224E+00     age
  35  B     1.168E+00     sex
  36  B     1.166E+00     edu
  37  B     1.165E+00     sod1
  38  B     1.038E+00     wblc1
  39  C     9.625E-01     immunhx
  40  C     8.970E-01     malighx
  41  D     8.870E-01     ca
  42  E     8.316E-01     scoma1
  43  E     8.198E-01     amihx
  44  E     6.724E-01     chfhx
  45  E     6.284E-01     gibledhx
```

```
46  E    4.206E-01    pot1
47  E    4.156E-01    ortho
48  E    3.967E-01    renalhx
49  E    3.695E-01    hema
50  E    3.531E-01    liverhx
51  E    3.119E-01    meta
52  E    2.838E-01    temp1
53  E    1.268E-01    race
```

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

```
par(las=1,mar=c(5,12,4,2),cex.axis=0.8)
leg.col <- c("red","orange","yellow","green","white")
leg.txt <- c("A (99%)","B (95%)","C (90%)","D (80%)","E (< 80%)")
x <- read.table("imp.scr",header=TRUE)
score <- x$Score
vars <- x$Variable
type <- x$Type
barcol <- rep("white",n)
letrs <- c("A","B","C","D","E")
for(i in 1:4){
    barcol[type == letrs[i]] <- leg.col[i]
}
n <- nrow(x)
barplot(rev(score[1:n]),names.arg=rev(vars[1:n]),col=rev(barcol[1:n]),horiz=TRUE,
        xlab="GUIDE importance scores")
legend("bottomright",legend=leg.txt,fill=leg.col)
```

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

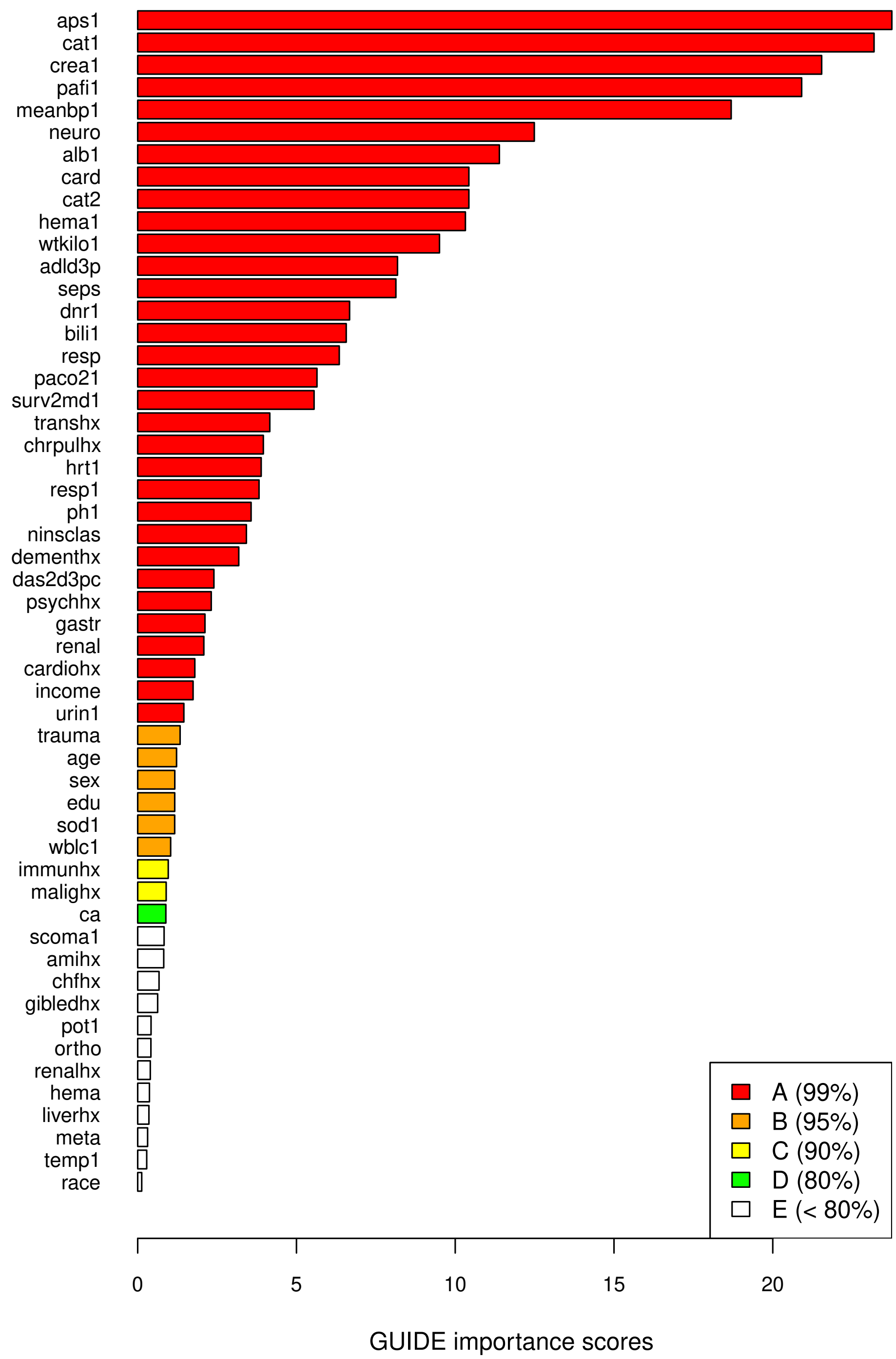

Figure 35: Scores of important variables for predicting `swang1`

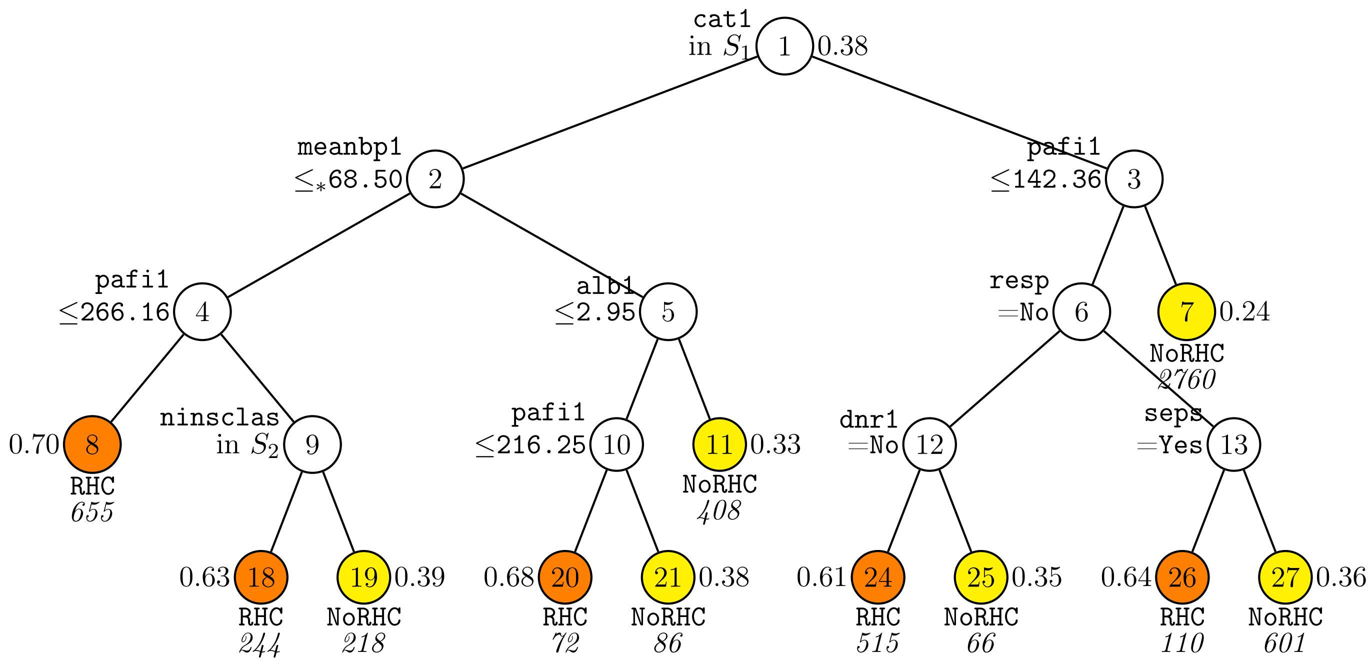

Figure 36: GUIDE v.41.1 importance scoring classification tree for predicting `swang1` using estimated priors and unit misclassification costs. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. S_1 = {`CHF`, `MOSF w/Sepsis`}. S_2 = {`No insurance`, `Private`, `Private & Medicare`}. Predicted classes and sample sizes (in *italics*) 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
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
23 N variables changed to S
Warning: model changed to linear in treatment
D variable is death
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Assigning integer codes to values of 31 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Treatment (R) variable is swang1 with values "NoRHC" and "RHC"
```

```
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
Smallest uncensored survtime: 2.0000
Number of complete cases excluding censored T < smallest uncensored T: 5735
Number of cases used to compute baseline hazard: 5735
Number of cases with D=1 and T >= smallest uncensored: 3722
GUIDE will try to create the variables in the description file.
If it is unsuccessful, please create the columns yourself...
Number of dummy variables created: 1
Input 1 if randomized trial, 2 if observational study: ([1:2], <cr>=1): 2
Choose a subgroup identification method:
1 = Prognostic priority (Gs)
2 = Predictive priority (Gi)
Input your choice: ([1:2], <cr>=2):
Creating dummy variables ...
Creating missing value indicators ...
Rereading data ...
Largest uncensored and censored survtime by swang1
 "NoRHC"      1867.0000    1243.0000
   "RHC"      1943.0000    1351.0000
Proportion of training sample for each level of swang1
"NoRHC"    0.6192
  "RHC"    0.3808
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        8        0        0       23
    #P-var   #M-var   #B-var   #C-var   #I-var   #R-var
         0        0        0       30        0        1
Survival time variable in column: 64
Event indicator variable in column: 9
Proportion uncensored among nonmissing T and D variables: .649
Number of cases used for training: 5735
Number of split variables: 53
Number of dummy variables created: 1
Number of cases excluded due to 0 W or missing D, T or R variables: 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 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):
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 file is created!
Run GUIDE with the command: guide < imp_surv.in
```

16.2.2 Partial contents of imp_surv.out

The output shows that there is only one important variable.

```
Scaled importance scores of predictor variables
 (F, I and R variables are excluded)
     Score     Rank  Variable
 1.055E+00     1.00  dnr1
--------- variables above this line have scores above 95% threshold (B) -----
--------- variables above this line have scores above 90% threshold (C) -----
 9.446E-01     2.00  ph1
 8.009E-01     3.00  chrpulhx
--------- variables above this line have scores above 80% threshold (D) -----
 7.860E-01     4.00  resp1
 7.851E-01     5.00  paco21
 4.947E-01     6.00  liverhx
 4.508E-01     7.00  pot1
 4.357E-01     8.00  gastr
 4.303E-01     9.00  cat2
 4.009E-01    10.00  gibledhx
 3.967E-01    11.00  age
 3.619E-01    12.00  pafi1
 3.456E-01    13.00  aps1
 3.229E-01    14.00  malighx
 3.161E-01    15.00  amihx
 2.988E-01    16.00  hrt1
 2.856E-01    17.00  surv2md1
 2.689E-01    18.00  ninsclas
 2.493E-01    19.00  das2d3pc
 2.441E-01    20.00  edu
 2.440E-01    21.00  meanbp1
 2.180E-01    22.00  income
 2.000E-01    23.00  scoma1
 1.802E-01    24.00  ortho
 1.753E-01    25.00  crea1
 1.736E-01    26.00  temp1
 1.708E-01    27.00  hema1
 1.603E-01    28.00  ca
```

```
 1.550E-01    29.00  hema
 1.466E-01    30.00  trauma
 1.461E-01    31.00  wtkilo1
 1.438E-01    32.00  renalhx
 1.435E-01    33.00  psychhx
 1.425E-01    34.00  sex
 1.393E-01    35.00  neuro
 1.325E-01    36.00  urin1
 1.312E-01    37.00  alb1
 1.263E-01    38.00  wblc1
 1.234E-01    39.00  chfhx
 9.847E-02    40.00  dementhx
 9.447E-02    41.00  adld3p
 9.210E-02    42.00  race
 8.458E-02    43.00  seps
 8.308E-02    44.00  sod1
 8.284E-02    45.00  cat1
 7.764E-02    46.00  resp
 7.314E-02    47.00  cardiohx
 5.035E-02    48.00  card
 4.893E-02    49.00  renal
 4.518E-02    50.00  transhx
 4.290E-02    51.00  meta
 4.066E-02    52.00  bili1
 3.810E-02    53.00  immunhx

99% threshold is 1.4556
95% threshold is 1.0000
90% threshold is 1.0000
80% threshold is 0.7934
Number of variables above 99% threshold is 0
Number of variables between 95% and 99% thresholds is 1
Number of variables between 90% and 95% thresholds is 0
Number of variables between 80% and 90% thresholds is 2
LaTeX code for tree is in imp_surv.tex
Importance scores are stored in imp_surv.scr
```

17 Causal inference

Propensity score matching is often used in causal inference to estimate average treatment effects. Given a treatment variable Z taking values 0 (no treatment) and 1 (treatment), the propensity score for a subject with covariate $X = x$ is $\pi(x) = P(Z = 1 \mid X = x)$. If n denotes the sample size and Y_i the response of the

ith subject, the average treatment effect may be estimated by the *Horvitz-Thompson estimate (HT)*

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

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 has difficulties if there are missing values in the covariates or if the number of covariates is large. Random forest has been used, but the version implemented in R is not applicable to data 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 $Z_i = 1$), resulting in $\hat{\pi}(x_i) = 0$ or 1. To avoid 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 *sample size weighted estimate* $n^{-1} \sum_t n_t \hat{\beta}_t$, where the sum is over the terminal nodes and n_t and $\hat{\beta}_t$ are the sample size and estimated treatment effect in node t.

We demonstrate the propensity score feature with the RHC data. Doctors believe that direct measurement of cardiac function by right heart catheterization for some critically ill patients yields better outcomes. The benefit of RHC has not been demonstrated in a randomized clinical trial due to ethical concerns. In observational studies, the relative risk of death was found to be higher in the elderly and in patients with acute myocardial infarction who received RHC. In such studies, the decision to use RHC is at the discretion of the physician. Therefore treatment assignment is confounded with patient factors that are also related to outcomes, e.g., patients with low blood pressure are more likely to get RHC, and such patients are also more likely to die. The data consist of observations on more than 60 variables for 5735 patients from 5 medical centers over 5 years (Connors et al., 1996). The treatment variable is `swang1` (`RHC` or `NoRHC`), and the response variables are `dth30` (1=death within 30 days, 0=survived more than 30 days) and `death` (1=eventual death, 0=censored).

The data and description files are `rhcdata.txt` and `rhcdsc4.txt`. In the latter, the variable `swang1` is designated as `r`, `dth30` as `d`, and `death` as `x`.

17.1 Input file creation

```
0. Read the warranty disclaimer
1. Create a GUIDE input file
Input your choice: 1
Name of batch input file: prop30.in
Input 1 for model fitting, 2 for importance or DIF scoring,
      3 for data conversion ([1:3], <cr>=1):
Name of batch output file: prop30.out
Input 1 for single tree, 2 for ensemble ([1:2], <cr>=1):
Input 1 for classification, 2 for regression, 3 for propensity score grouping
Input your choice ([1:3], <cr>=1): 3
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: rhcdsc4.txt
Reading data description file ...
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
35 N variables changed to S
Warning: model changed to linear in treatment
D variable is dth30
Reading data file ...
Number of records in data file: 5735
Length of longest entry in data file: 19
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
Number of classes: 2
Assigning integer codes to values of 18 categorical variables
Finished assigning codes to 10 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Finished processing 5000 of 5735 observations
Data checks complete
Creating missing value indicators ...
```

```
Rereading data ...
Treatment      #Cases    Proportion
NoRHC            3551    0.61918047
RHC              2184    0.38081953
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      5735          0       5157        9        0        0       35
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       18        0
Number of cases used for training: 5735
Number of split variables: 53
Number of cases excluded due to 0 W or missing D variable: 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
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: prop30.fit
Input 2 to write R function for predicting new cases, 1 otherwise ([1:2], <cr>=2):
Input file name: prop30.r
Input rank of top variable to split root node ([1:53], <cr>=1):
Input file is created!
Run GUIDE with the command: guide < prop30.in
```

17.2 Contents of prop30.out

```
Propensity score grouping and estimation of causal effects
Pruning by cross-validation
Data description file: rhcdsc4.txt
Training sample file: rhcdata.txt
Missing value code: NA
Records in data file start on line 2
R variable present
35 N variables changed to S
Warning: model changed to linear in treatment
D variable is dth30
Number of records in data file: 5735
Length of longest entry in data file: 19
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 2
```

```
Treatment      #Cases    Proportion
NoRHC            3551    0.61918047
RHC              2184    0.38081953

Summary information for training sample of size 5735
d=dependent, b=split and fit cat variable using indicator variables,
c=split-only categorical, i=fit-only categorical (via indicators),
s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
m=missing-value flag variable, p=periodic variable, w=weight
                                                  #Codes/
                                                  Levels/
 Column  Name           Minimum      Maximum    Periods   #Missing
      2  cat1       c                                  9
      3  cat2       c                                  6     4535
      4  ca         c                                  3
     10  cardiohx   s    0.000        1.000
     :
     26  das2d3pc   s    11.00        33.00
     28  dth30      d    0.000        1.000
     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     r                                      2
     46  wtkilo1    s    19.50        244.0                  515
      :
     58  ortho      c                                      2
     59  adld3p     s    0.000        7.000                 4296
     60  urin1      s    0.000        9000.                 3028
     61  race       c                                      3
     62  income     c                                      4

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

```
      5735          0       5157        9        0        0       35
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0       18        0
 Number of cases used for training: 5735
 Number of split variables: 53
 Number of cases excluded due to 0 W or missing D variable: 0

 Constant fitted to cases with missing values in regressor variables
 Pruning by v-fold cross-validation, with v = 10
 Selected tree is based on mean of CV estimates
 Number of SE's for pruned tree: 0.2500

 Simple node models
 Equal priors
 Unit misclassification costs
 Univariate split highest priority
 Interaction splits 2nd priority; no linear splits
 Split values for N and S variables based on exhaustive search
 Maximum number of split levels: 15
 Minimum node sample size: 6
 Top-ranked variables and 1-df 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
      :
     52  0.1052E+01   meta
     53  0.6357E+00   race

 Size and CV mean cost and SE of subtrees:
  Tree   #Tnodes  Mean Cost   SE(Mean)   BSE(Mean)  Median Cost  BSE(Median)
    1      354   3.581E-01   6.817E-03   6.202E-03   3.609E-01   5.354E-03
    2      353   3.581E-01   6.817E-03   6.202E-03   3.609E-01   5.354E-03
    :
  208       18   3.278E-01   6.421E-03   4.257E-03   3.277E-01   6.447E-03
  209**     16   3.255E-01   6.349E-03   5.516E-03   3.205E-01   9.186E-03
  210       14   3.287E-01   6.301E-03   5.926E-03   3.290E-01   9.957E-03
  211       12   3.285E-01   6.339E-03   5.849E-03   3.268E-01   8.241E-03
  212        8   3.330E-01   6.355E-03   7.153E-03   3.315E-01   8.781E-03
  213        6   3.360E-01   6.287E-03   6.883E-03   3.325E-01   9.229E-03
  214        5   3.527E-01   6.506E-03   7.212E-03   3.511E-01   5.489E-03
  215        4   3.690E-01   6.337E-03   7.280E-03   3.705E-01   9.859E-03
  216        2   4.131E-01   5.710E-03   3.745E-03   4.112E-01   3.751E-03
  217        1   5.000E-01   8.419E-03   2.585E-16   5.000E-01   2.764E-16

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

```
0-SE tree based on median is marked with + and has 16 terminal nodes
Selected-SE tree based on mean using naive SE is marked with **
Selected-SE tree based on mean using bootstrap SE is marked with --
Selected-SE tree based on median and bootstrap SE is marked with ++
* tree, ** tree, + tree, and ++ tree all the same

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

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

Node cost is node misclassification cost divided by number of training cases
       Node    Total    Train    Predicted        Node    Split          Interacting
      label    cases    cases    class            cost    variables      variable
          1     5735     5735    NoRHC         5.000E-01 cat1
          2     4572     4572    RHC           4.469E-01 pafi1
          4     2218     2218    RHC           3.640E-01 crea1
          8      823      823    RHC           4.738E-01 pafi1
        16T      370      370    RHC           3.757E-01 resp
         17      453      453    NoRHC         4.385E-01 trauma
        34T       14       14    RHC           9.298E-02 -
         35      439      439    NoRHC         4.193E-01 card
        70T      107      107    RHC           4.213E-01 crea1
        71T      332      332    NoRHC         3.624E-01 bili1 :aps1
          9     1395     1395    RHC           3.044E-01 adld3p
        18T     1144     1144    RHC           2.608E-01 wtkilo1
         19      251      251    NoRHC         4.675E-01 resp1
        38T      114      114    RHC           3.483E-01 resp1
        39T      137      137    NoRHC         2.852E-01 gastr
          5     2354     2354    NoRHC         4.682E-01 cat1
         10     1076     1076    RHC           4.030E-01 meanbp1
        20T      798      798    RHC           3.358E-01 bili1
        21T      278      278    NoRHC         3.753E-01 cat1 :age
         11     1278     1278    NoRHC         3.462E-01 cat2
         22      291      291    RHC           4.813E-01 wtkilo1
        44T      108      108    NoRHC         3.287E-01 pafi1
        45T      183      183    RHC           3.834E-01 resp
        23T      987      987    NoRHC         2.898E-01 wtkilo1
          3     1163     1163    NoRHC         2.615E-01 aps1
         6T      895      895    NoRHC         1.666E-01 card
          7      268      268    RHC           4.691E-01 cat2
        14T       72       72    RHC           3.052E-01 meanbp1
         15      196      196    NoRHC         4.635E-01 income
        30T       25       25    RHC           2.570E-01 wblc1
        31T      171      171    NoRHC         4.154E-01 card

Number of terminal nodes of final tree: 16
```

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

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

Node 1: cat1 = "ARF", "CHF", "MOSF w/Malignancy", "MOSF w/Sepsis"
  Node 2: pafi1 <= 188.43750
    Node 4: crea1 <= 1.2498779
      Node 8: pafi1 <= 116.48438
        Node 16: RHC
      Node 8: pafi1 > 116.48438 or NA
        Node 17: trauma = "Yes"
          Node 34: RHC
        Node 17: trauma /= "Yes"
          Node 35: card = "Yes"
            Node 70: RHC
          Node 35: card /= "Yes"
            Node 71: NoRHC
    Node 4: crea1 > 1.2498779 or NA
      Node 9: adld3p = NA
        Node 18: RHC
      Node 9: adld3p /= NA
        Node 19: resp1 <= 29.500000 or NA
          Node 38: RHC
        Node 19: resp1 > 29.500000
          Node 39: NoRHC
  Node 2: pafi1 > 188.43750 or NA
    Node 5: cat1 = "CHF", "MOSF w/Sepsis"
      Node 10: meanbp1 <= 98.500000 or NA
        Node 20: RHC
      Node 10: meanbp1 > 98.500000
        Node 21: NoRHC
    Node 5: cat1 /= "CHF", "MOSF w/Sepsis"
      Node 11: cat2 = "MOSF w/Sepsis"
        Node 22: wtkilo1 <= 66.449950
          Node 44: NoRHC
        Node 22: wtkilo1 > 66.449950 or NA
          Node 45: RHC
      Node 11: cat2 /= "MOSF w/Sepsis"
        Node 23: NoRHC
Node 1: cat1 /= "ARF", "CHF", "MOSF w/Malignancy", "MOSF w/Sepsis"
  Node 3: aps1 <= 61.500000
    Node 6: NoRHC
  Node 3: aps1 > 61.500000 or NA
    Node 7: cat2 = "MOSF w/Sepsis"
```

```
      Node 14: RHC
    Node 7: cat2 /= "MOSF w/Sepsis"
      Node 15: income = "$25-$50k", "> $50k"
        Node 30: RHC
      Node 15: income /= "$25-$50k", "> $50k"
        Node 31: NoRHC

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

Node 1: Intermediate node
 A case goes into Node 2 if cat1 = "ARF", "CHF", "MOSF w/Malignancy", "MOSF w/Sepsis"
 cat1 mode = "ARF"
 Number of observations in node = 5735
 Regressor     Coefficient  t-stat       p-value
 Constant      0.3064        38.80        0.000
 swang1.RHC    0.7364E-01    5.756       0.9026E-08
 Number of observations in node = 5735
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if pafi1 <= 188.43750
 pafi1 mean = 215.63083
 Number of observations in node = 4572
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if crea1 <= 1.2498779
 crea1 mean = 2.1359302
 Number of observations in node = 2218
 ----------------------------
Node 8: Intermediate node
 A case goes into Node 16 if pafi1 <= 116.48438
 pafi1 mean = 120.46293
 Number of observations in node = 823
 ----------------------------
Node 16: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.3115        8.801       0.7772E-15
 swang1.RHC    0.9494E-01    1.907       0.5729E-01
 Number of observations in node = 370
 ----------------------------
Node 17: Intermediate node
 A case goes into Node 34 if trauma = "Yes"
 trauma mode = "No"
 Number of observations in node = 453
 ----------------------------
```

```
 :
 ----------------------------
Node 30: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.4444        2.617       0.1543E-01
 swang1.RHC   -0.6944E-01  -0.3271       0.7466
 Number of observations in node = 25
 ----------------------------
Node 31: Terminal node
 Regressor     Coefficient  t-stat       p-value
 Constant      0.5294        11.96        0.000
 swang1.RHC    0.2206        2.748       0.6641E-02
 Number of observations in node = 171
 ----------------------------
 Regression estimates are weighted means over terminal nodes
 regressor     coefficient  z-stat       p-value
 constant      0.3160        38.52        0.000
 swang1.RHC    0.5191E-01    3.597       0.3222E-03

 average treatment effect of swang1 level "RHC" vs level "NoRHC" =  5.1909E-02

 Observed and fitted values are stored in prop30.fit
 LaTeX code for tree is in prop30.tex
 R code is stored in prop30.r
```

The results at the end of `prop30.out` show that the average treatment effect is 0.51909. The LaTeX tree is shown in Figure 37. The number beside each terminal node is the proportion of observations with `swang1 = RHC` ($Z = 1$). The pair below each node are the sample means of Y corresponding to $Z = 0$ and 1. GUIDE treats "`NoRHC`" as $Z = 0$ because it precedes "`RHC`" in alphabetical order.

The file `prop30.fit` gives the proportions of `swang1` in the rightmost two columns. Here are the top 5 rows of the file:

```
train   node  observed  predicted  "P(NoRHC)"    "P(RHC)"
    y      6  "NoRHC"    "NoRHC"   0.89050E+00  0.10950E+00
    y     20  "RHC"      "RHC"     0.45113E+00  0.54887E+00
    y     45  "RHC"      "RHC"     0.50273E+00  0.49727E+00
    y     18  "NoRHC"    "RHC"     0.36451E+00  0.63549E+00
    y     20  "RHC"      "RHC"     0.45113E+00  0.54887E+00
```

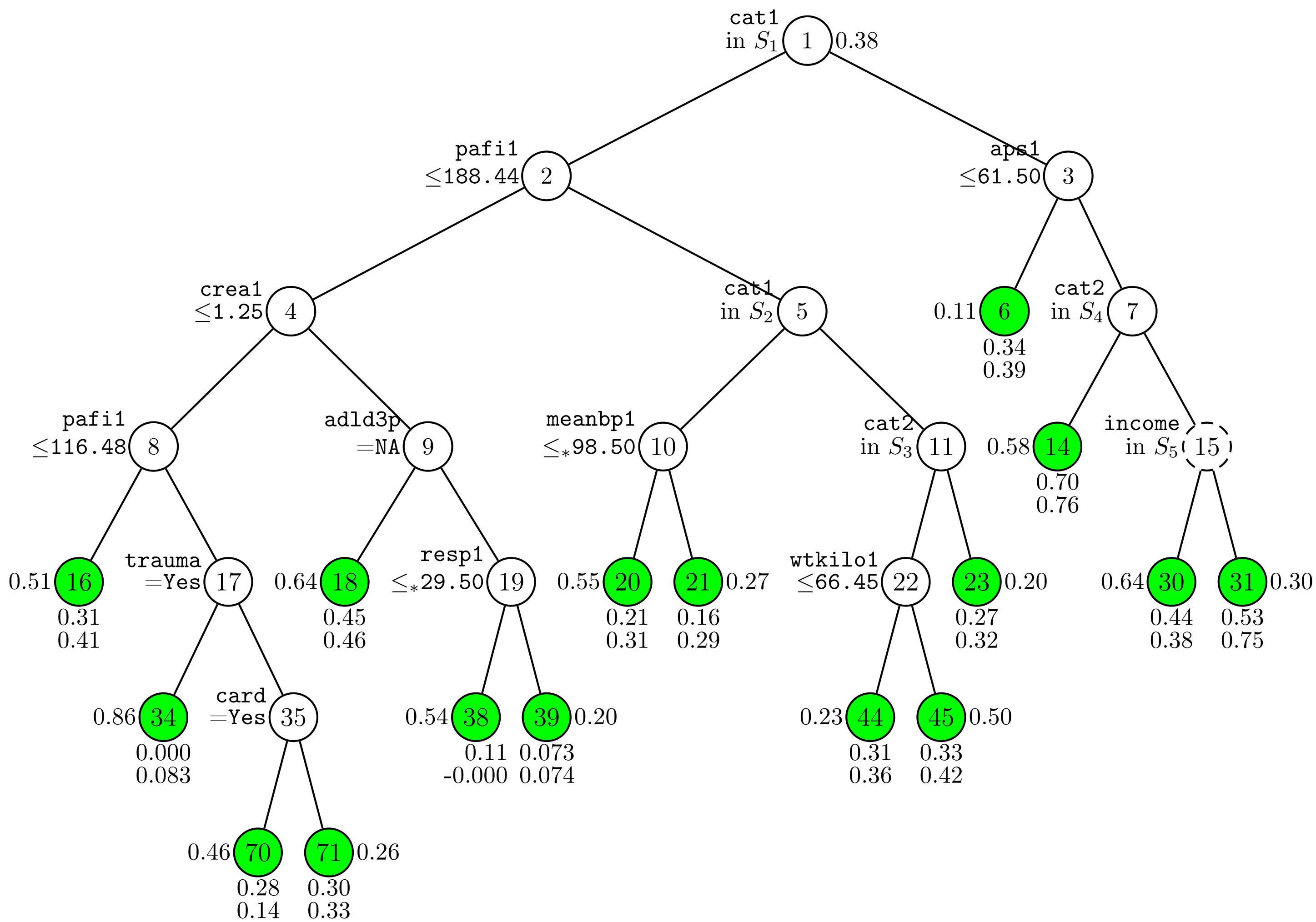

Figure 37: GUIDE v.41.1 0.250-SE tree for propensity score grouping and estimation of effects of `swang1` on `dth30`. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. S_1 = {`ARF`, `CHF`, `MOSF w/Malignancy`, `MOSF w/Sepsis`}. S_2 = {`CHF`, `MOSF w/Sepsis`}. S_3 = {`MOSF w/Sepsis`}. S_4 = {`MOSF w/Sepsis`}. S_5 = {`$25-$50k`, `> $50k`}. Circles with dashed lines are nodes with no significant split variables. Sample means of `dth30` for `swang1` levels `NoRHC` and `RHC`, respectively, printed below nodes. Sample proportion of `swang1` = `RHC` printed beside nodes. Second best split variable at root node is `aps1`.

18 Differential item functioning: GDS data

GUIDE has an experimental option to identify important predictor variables and items with differential item functioning (DIF) in a data set with two or more item (dependent variable) scores. We illustrate it with a data set from Broekman et al. (2011, 2008) and Marc et al. (2008). It consists of responses from 1978 subjects on 15 items. There are 3 predictor variables (age, education, and gender). The data and description files are `GDS.dat` and `GDS.dsc`. Although the item responses in this example are 0-1, GUIDE allows them to be in any ordinal (e.g., Likert) scale. The contents of `GDS.dsc` are:

```
GDS.dat
NA
1
1 rid x
2 satis d
3 drop d
4 empty d
5 bored d
6 spirit d
7 afraid d
8 happy d
9 help d
10 home d
11 memory d
12 alive d
13 worth d
14 energy d
15 hope d
16 better d
17 total x
18 gender c
19 education n
20 age n
21 dxcurren x
22 sumscore x
```

Here is the session log to create an input file for identifying DIF items and the important predictor variables:

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

```
Name of batch output file: dif.out
Input 1 for classification, 2 for regression, 3 for propensity score grouping
Input your choice ([1:3], <cr>=1): 2
Choose type of regression model:
 1=linear, 2=quantile, 3=Poisson, 4=censored response,
 5=multiresponse or itemresponse, 6=longitudinal data (with T variables).
Input choice ([1:6], <cr>=1): 5
Input 1 for default options, 2 otherwise ([1:2], <cr>=1):
Input name of data description file (max 100 characters);
enclose with matching quotes if it has spaces: GDS.dsc
Reading data description file ...
Training sample file: GDS.dat
Missing value code: NA
Records in data file start on line 1
2 N variables changed to S
Number of D variables: 15
D variables are:
satis
drop
empty
bored
spirit
afraid
happy
help
home
memory
alive
worth
energy
hope
better
Multivariate or univariate split variable selection:
Choose multivariate if there is an order among the D variables;
choose univariate otherwise or if item response
Input 1 for multivariate, 2 for univariate ([1:2], <cr>=1): 2
D variables can be normalized to have unit variance,
e.g., if they have different scales or units
Input 1 to normalize D variables, 2 for no normalization ([1:2], <cr>=1): 2
Input 1 for equal, 2 for unequal weighting of D variables ([1:2], <cr>=1):
Reading data file ...
Number of records in data file: 1978
Length of longest entry in data file: 4
Checking for missing values ...
Finished checking
Missing values found in D variables
```

```
Assigning integer codes to values of 1 categorical variables
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Some D variables have missing values
Rereading data ...
PCA can be used for variable selection
Do not use PCA if differential item functioning (DIF) scores are wanted
Input 1 to use PCA, 2 otherwise ([1:2], <cr>=2):
#cases w/ miss. D = number of cases with all D values missing
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      1978          0          0        4        0        0        2
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0        1        0
Number of cases used for training: 1977
Number of split variables: 3
Number of cases excluded due to 0 W or missing D variable: 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
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 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):
You can create a description file with the selected variables included or excluded
Input 2 to create such a file, 1 otherwise ([1:2], <cr>=1):
You can also output the importance scores and variable names to a file
Input 1 to create such a file, 2 otherwise ([1:2], <cr>=1):
Input file name: dif.scr
Input file is created!
Run GUIDE with the command: guide < dif.in
```

The importance scores are in the file `dif.scr`. They show that `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 38 shows the tree.

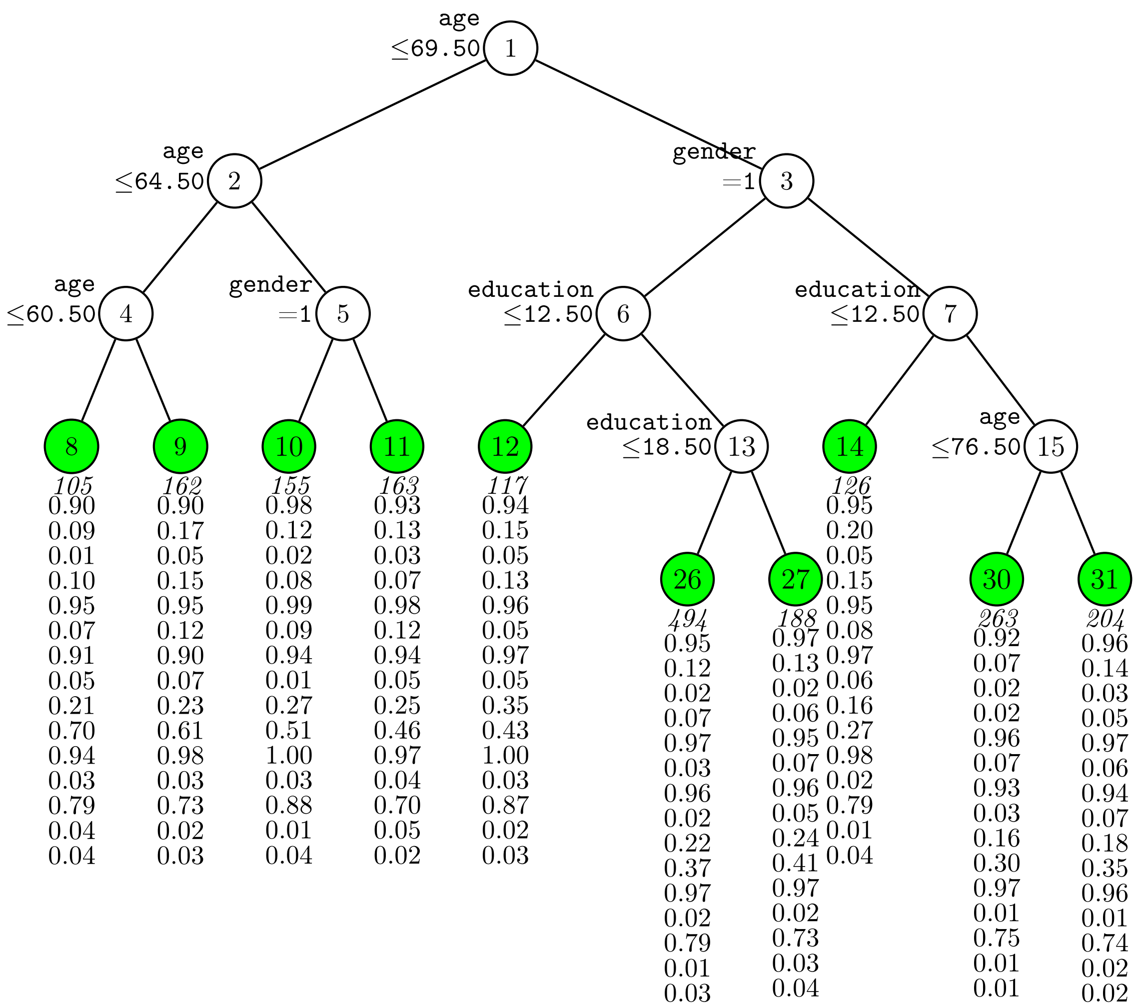

Figure 38: GUIDE v.41.1 importance scoring or DIF regression tree for predicting response variables `satis`, `drop`, `empty`, `bored`, `spirit`, `afraid`, `happy`, `help`, `home`, `memory`, `alive`, `worth`, `energy`, `hope`, and `better`, without using PCA at each node. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (in *italics*) and predicted values of `satis`, `drop`, `empty`, `bored`, `spirit`, `afraid`, `happy`, `help`, `home`, `memory`, `alive`, `worth`, `energy`, `hope`, and `better` printed below nodes. Second best split variable at root node is `gender`.

19 Bootstrap confidence intervals

Owing to the numerous procedures that are performed during tree construction (such as selection of the variable and the split set to partition each intermediate node), proper statistical inference must account for the multiple testing and estimation issues. Otherwise, the error variance will be underestimated. Suppose, for example, we wish to obtain confidence intervals for the proportion of "`RHC`" in each terminal node of the tree in Figure 1. Let n denote the sample size in a node and $\hat{p}$ the proportion of observations in it with the response value `RHC`. The usual $(1-\alpha)$ binomial interval is then $\hat{p} \pm z_{1-\alpha/2}\sqrt{\hat{p}(1-\hat{p})/n}$, where z_α is the α -quantile of the standard normal distribution. This formula yields intervals that are too short because it does not account for the extra variance due to model construction. Bonferroni corrections, which are traditionally used for multiple testing, are inapplicable here because the number of tests are not specified in advance. For example, the number of chi-squared tests at each node depends on the number of variables eligible to split the node and the number of levels of splits depends on the total sample size, extent of pruning, and other parameters such as the minimum sample size in each node.

As with the Bonferroni correction, a natural solution is to change the multiplier $z_{1-\alpha/2}$ to a larger value. The bootstrap method provides one simple solution. Called "bootstrap calibration", the procedure is described and analyzed in Loh (1987, 1991) in the context of estimating a nonparametric mean; it is extended to subgroup analysis from regression tree models in Loh et al. (2016, 2019c) and Loh and Zhou (2020). The R code below implements the procedure. It can be used by following these steps:

1. Change the name of the data file (`rhcdata.txt` here) to `realdata.txt`.
2. Change the name of the description file (`rhcdsc1.txt` here) to `real.dsc`.
3. Change the name of the GUIDE input file (`classin.txt` here) to `real.in`.
4. Change the word "`RHC`" in line 1 of the R code to the name of the desired class in the data file.
5. In Windows, change the word "`system`" in lines 32, 32, 74 and 75 to "`shell`" if necessary.
6. Source the program in R.

```
class.name <- "RHC"  ## name of desired class in realdata.txt
nboot <- 1000
probs <- c(0.80,0.90,0.95,0.98)
zstat <- rep(0,nboot)
### write bootstrap description file boot.dsc
file <- readLines("real.dsc")  ## read real description file
write("bootdata.txt",file="boot.dsc")
len <- length(file)
write(file[2:length(file)],"boot.dsc",append=TRUE)
write(paste(len-2,"w w"),"boot.dsc",append=TRUE)
### write bootstrap input file boot.in
file <- readLines("real.in")  ## read real input file
file2 <- gsub("real.","boot.",file) ## replace "real." with "boot."
write(file2,"boot.in")
### read real data
z0 <- read.table("realdata.txt",header=TRUE)
nobs <- nrow(z0)
zt <- cbind(z0,rep(0,nobs)) ### add column of weight 0
write("Bootstrap simultaneous intervals by linear interpolation of z",
      "results.txt")
write("trials  z80    z90    z95     z98      bias.err     sd.err",
      "results.txt", append=TRUE)
err.test <- rep(0,nboot) ## misclassification rates
bias <- 0
for(i in 1:nboot){
    zb <- z0[sample(nobs,nobs,replace=TRUE),]
    zb <- cbind(zb,rep(1,nobs)) ### add column of weight 1
    write.table(zb,"bootdata.txt",col.names=TRUE,row.names=FALSE)
    write.table(zt,"bootdata.txt",col.names=FALSE,row.names=FALSE,
                append=TRUE)
    system("rm -f log.txt boot.out boot.fit")
    system("guide < boot.in > log.txt")
    bfit <- read.table("boot.fit",header=TRUE)  ## read boot results
    test <- bfit$train == "n"
    err.test[i] <- sum(bfit$observed[test] != bfit$predicted[test])/nobs
    err.resub <- sum(bfit$observed[!test] != bfit$predicted[!test])/nobs
    bias <- bias+(err.resub-err.test[i])
    unodes <- unique(sort(bfit$node))
    for(j in 1:length(unodes)){
        gp <- bfit$node == unodes[j] & bfit$train == "y" ## training data
        n0 <- sum(bfit$observed[gp] != class.name)
        n1 <- sum(bfit$observed[gp] == class.name)
        ntot <- n0+n1
        estp <- n1/ntot
        if(n1 == 0 | n0 == 0){
            p <- (n1+0.5)/(ntot+1)
```

```
            sd <- sqrt(p*(1-p)/(ntot+1))
        } else {
            sd <- sqrt(estp*(1-estp)/ntot)
        }
        gp <- bfit$node == unodes[j] & bfit$train == "n"  ## real data
        n0 <- sum(bfit$observed[gp] != class.name)
        n1 <- sum(bfit$observed[gp] == class.name)
        realp <- n1/(n0+n1)
        zstat[i] <- max(zstat[i],abs(realp-estp)/sd)
    }
    if(i %% 100 == 0){
        sd.err <- sqrt(var(err.test[1:i])) ## linear interpolation
        q <- quantile(zstat[1:i],probs=probs,type=4)
        write(c(i,q,bias/i,sd.err),"results.txt",append=TRUE,ncol=7)
    }
}
### find calibrated z.alpha
write(paste("No.␣bootstraps␣=␣",nboot),"results.txt",append=TRUE)
write(c("Calibrated␣z␣at␣levels",probs),file="results.txt",ncol=5,
      append=TRUE)
q <- quantile(zstat,probs=probs,type=4) ## linear interpolation
write(q,"results.txt",append=TRUE,ncol=4)
write(paste("Bootstrap␣estimate␣of␣bias␣of␣error␣rate␣=␣",bias/nboot),
      "results.txt",append=TRUE)
write(paste("Bootstrap␣estimate␣of␣SD␣of␣error␣rate␣=␣",
      sqrt(var(err.test))),"results.txt",append=TRUE)
### fit real data
system("rm␣-f␣log.txt␣real.out␣real.fit")
system("guide␣<␣real.in␣>␣log.txt")
realfit <- read.table("real.fit",header=TRUE)
train <- realfit$train == "y"
err.obs <- sum(realfit$observed[train] != realfit$predicted[train])/nobs
write(paste("Real␣data␣observed␣error␣rate␣=␣",err.obs),"results.txt",
      append=TRUE)
k <- 3     ## 95% level
z0 <- q[k] ## 95% z value
write(c("Simultaneous␣intervals␣at␣level",probs[k]),
      file="results.txt",ncol=2,append=TRUE)
write(paste0("Node␣␣N␣␣␣P(",class.name,")␣halfwid␣␣␣left␣␣␣right"),
      "results.txt", append=TRUE)
unodes <- unique(sort(realfit$node))
for(j in 1:length(unodes)){
    gp <- realfit$node == unodes[j] & realfit$train == "y"
    n0 <- sum(realfit$observed[gp] != class.name)
    n1 <- sum(realfit$observed[gp] == class.name)
    ntot <- n0+n1
```

```
    if(n1 == 0 | n0 == 0){
        p <- (n1+0.5)/(ntot+1)
        sd <- sqrt(p*(1-p)/(ntot+1))
    } else {
        p <- n1/ntot
        sd <- sqrt(p*(1-p)/(ntot))
    }
    p <- n1/ntot
    halfwid <- z0*sd
    left <- p-halfwid
    rght <- p+halfwid
    write(c(unodes[j],ntot,p,halfwid,left,rght),"results.txt",
          append=TRUE,ncol=6)
}
## write(sort(zstat),"zstat.txt",ncol=1) ## output sorted zstat values
```

Figure 39 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 40.

20 Tree ensembles

A tree ensemble is a collection of trees. GUIDE has two methods of constructing an ensemble.

GUIDE forest. This the preferred method. Similar to Random Forest (Breiman, 2001), it fits *unpruned* trees to bootstrap samples and randomly selects a small subset of variables to search for splits at each node. There are, however, two important differences:

1. GUIDE forest uses the unbiased GUIDE method for split selection; Random Forest uses the biased CART method. One consequence is that GUIDE forest can be very much faster than Random Forest if the dependent variable is a class variable having more than two distinct values and some categorical predictor variables have many categories.
2. GUIDE forest is applicable to data with missing values. The R implementation of Random Forest (Liaw and Wiener, 2002) requires apriori imputation of missing values in the predictor variables.

```
Bootstrap simultaneous intervals by linear interpolation of z
trials  z80    z90    z95     z98      bias.err     sd.err
100 4.036962 4.458809 4.545827 4.922293 -0.03357803 0.005906056
200 4.123996 4.508203 4.777955 5.035208 -0.03335222 0.005670584
300 4.093978 4.513735 4.918732 5.117146 -0.0335048 0.00598086
400 4.108083 4.519645 4.835633 5.28808 -0.03360811 0.005930667
500 4.108083 4.508203 4.826329 5.117146 -0.03377507 0.005887693
600 4.144132 4.548011 4.895352 5.408027 -0.03397879 0.005812075
700 4.123996 4.529434 4.889087 5.408027 -0.03377357 0.005839512
800 4.117319 4.51814 4.845685 5.365021 -0.03369159 0.00588305
900 4.108552 4.50332 4.835633 5.408027 -0.03358888 0.005924705
1000 4.108083 4.495735 4.845685 5.397256 -0.03353304 0.005951228
No. bootstraps =  1000
Calibrated z at levels 0.8 0.9 0.95 0.98
4.108083 4.495735 4.845685 5.397256
Bootstrap estimate of bias of error rate =  -0.0335330427201395
Bootstrap estimate of SD of error rate =  0.00595122775778847
Real data observed error rate =  0.296251089799477
Simultaneous intervals at level 0.95
Node  N   P(RHC) halfwid   left   right
5 566 0.3816254 0.09894446 0.282681 0.4805699
7 2760 0.2355072 0.03913718 0.1963701 0.2746444
8 655 0.6961832 0.08707675 0.6091065 0.78326
18 244 0.6270492 0.1500158 0.4770334 0.7770649
19 218 0.3853211 0.1597212 0.2255999 0.5450423
25 66 0.3484848 0.2842088 0.06427609 0.6326936
26 110 0.6363636 0.2222518 0.4141119 0.8586154
27 601 0.3627288 0.09503228 0.2676965 0.4577611
48 438 0.6552511 0.1100458 0.5452053 0.7652969
49 77 0.3506494 0.2635033 0.08714608 0.6141526
```

Figure 39: Contents of `results.txt`

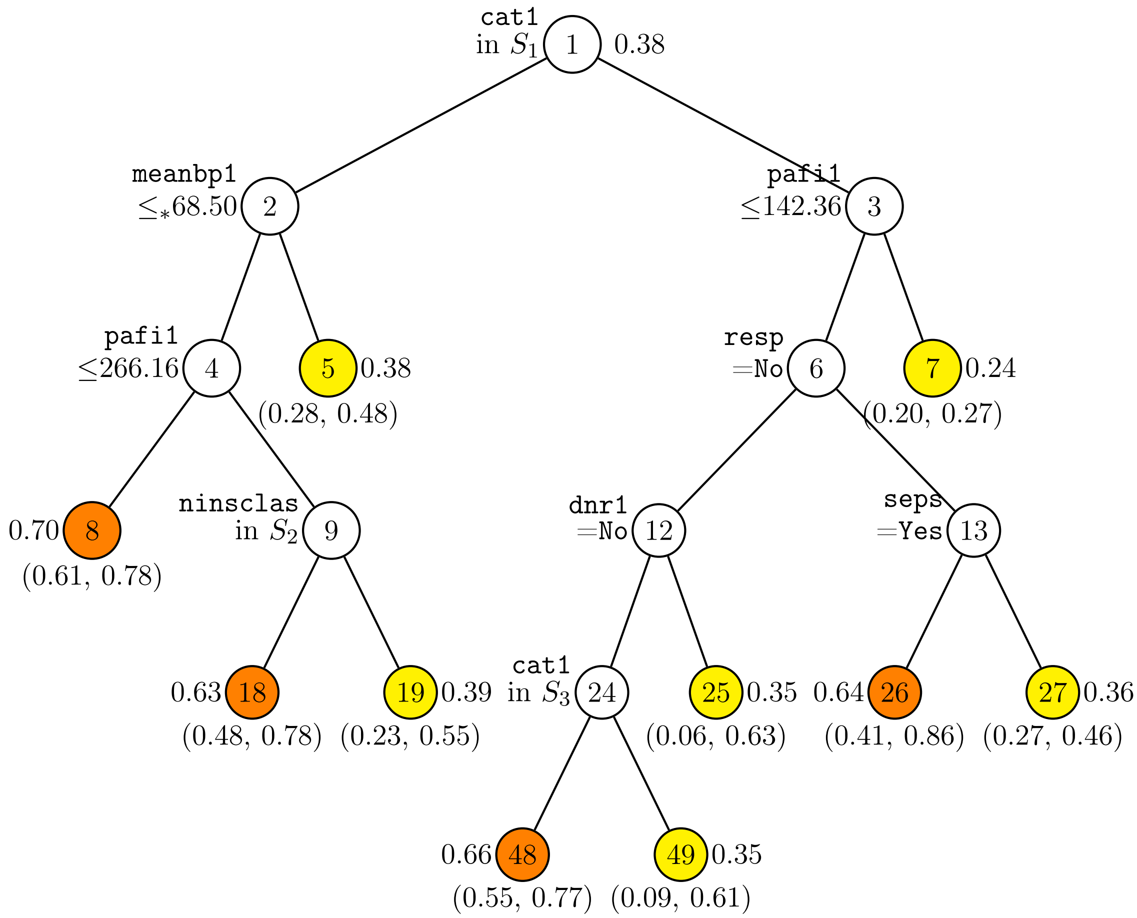

Figure 40: GUIDE v.38.0 0.25-SE classification tree for predicting `swang1` using estimated priors and unit misclassification costs. At each split, an observation goes to the left branch if and only if the condition is satisfied. Symbol '$\leq_*$' stands for '$\leq$ or missing'. Set S_1 = {`CHF`, `MOSF 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. 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 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: ce2021class.dsc
Reading data description file ...
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
384 N variables changed to S
D variable is INTRDVX_
Reading data file ...
Number of records in data file: 3965
Length of longest entry in data file: 11
Checking for missing values ...
Finished checking
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Recoding D values to integers
Finished recoding
```

```
Number of classes: 3
Finding number of levels of M variables associated with C variables ...
Assigning integer codes to values of 47 categorical variables
Finished assigning codes to 10 categorical variables
Finished assigning codes to 20 categorical variables
Finished assigning codes to 30 categorical variables
Finished assigning codes to 40 categorical variables
Associating missing values of N and S variables with M variable codes ...
Re-checking data ...
Allocating missing value information ...
Assigning codes to missing values, if any ...
Data checks complete
Creating missing value indicators ...
Rereading data ...
Warning: S variable DIRACC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable ROTHRFLC is constant
Warning: S variable WELFREBX is constant
Smallest positive weight:  1.0725E+03
Largest positive weight:   9.3902E+04
Class  #Cases     Proportion
C        1478     0.37276166
D        2431     0.61311475
T          56     0.01412358
     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965          0       3965        1        0        0      384
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Number of cases used for training: 3965
Number of split variables: 431
Number of cases excluded due to 0 W or missing D variable: 0
Finished reading data file
Warning: No linear splits; number of S variables must be < 225
Choose 1 for estimated priors, 2 for equal priors, 3 to input priors from a file
Input 1, 2, or 3 ([1:3], <cr>=1):
Choose 1 for unit misclassification costs, 2 to input costs from a file
Input 1 or 2 ([1:2], <cr>=1):
Warning: All positive weights treated as 1
Input name of file to store predicted class and probability: gf.pro
Input rank of top variable to split root node ([1:431], <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: ce2021class.dsc
Training sample file: ce2021.txt
Missing value code: NA
Records in data file start on line 2
Number of M variables associated with C variables: 19
384 N variables changed to S
D variable is INTRDVX_
Number of records in data file: 3965
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 DIRACC is constant
Warning: S variable TOTHVHRP is constant
Warning: S variable TOTHVHRC is constant
Warning: S variable ROTHRFLC is constant
Warning: S variable WELFREBX is constant
Smallest and largest positive weights are  1.0725E+03 and 9.3902E+04
Training sample class proportions of D variable INTRDVX_:
Class  #Cases     Proportion
C        1478     0.37276166
D        2431     0.61311475
T          56     0.01412358

Summary information for training sample of size 3965
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     s    1.000        1.000                   170
      2  DIRACC_    m                                 2
      3  AGE_REF    s    18.00        87.00
      4  AGE_REF_   m                                 0
      5  AGE2       s    21.00        87.00                  1734
      6  AGE2_      m                                 1
```

```
      :
    545  STOCKYR    c                                  1     3942
    546  STOCKYR_   m                                  1
    547  WHLFYR     c                                  1     3964
    548  WHLFYR_    m                                  1
    549  FFTAXOWE   s  -0.3368E+05   0.3997E+06
    550  FSTAXOWE   s   -3309.       0.7223E+05

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      3965          0       3965        1        0        0      384
    #P-var   #M-var   #B-var   #C-var   #I-var
         0      116        0       47        0
Number of cases used for training: 3965
Number of split variables: 431
Number of cases excluded due to 0 W or missing D variable: 0

Number of trees in ensemble: 500
Number of variables used for splitting: 144
Warning: No linear splits; number of S variables must be < 225
Simple node models
Estimated priors
Unit misclassification costs
Warning: All positive weights treated as 1
Univariate split highest priority
No interaction splits
No linear splits
Fraction of cases used for splitting each node: .0252
Maximum number of split levels: 18
Minimum node sample size: 19
Mean number of terminal nodes:    135.8

Classification matrix for training sample:
Predicted       True class
class              C         D         T
C               1047        54         6
D                431      2377        50
T                  0         0         0
Total           1478      2431        56

Number of cases used for tree construction: 3965
Number misclassified: 541
Resubstitution estimate of mean misclassification cost: .1364

Number of OOB cases: 3965
Number OOB misclassified: 961
```

```
OOB estimate of mean misclassification cost: .2424
Mean number of trees per OOB observation: 183.93

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   0.79713E+00  0.19766E+00  0.52107E-02    "C"    "C"
y   0.17413E+00  0.81800E+00  0.78689E-02    "D"    "D"
y   0.19204E+00  0.80242E+00  0.55365E-02    "D"    "D"
y   0.30491E+00  0.69031E+00  0.47743E-02    "D"    "D"
y   0.28991E+00  0.69812E+00  0.11970E-01    "D"    "D"
y   0.40932E+00  0.48080E+00  0.10988E+00    "D"    "T"
y   0.36835E+00  0.61102E+00  0.20625E-01    "D"    "D"
```

20.2 Bagged GUIDE

This option uses an ensemble of **pruned** GUIDE trees. It often takes longer to execute and does not appear to produce more accurate results. It is made available for research purposes.

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 `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 `wtgain`^{2} for fitting terminal node models only. Note: The line defines `wtgain`^{2} as `wtgain`$^2 \times$ `wtgain`^{0}. Since we can equivalently define the variable by `wtgain`$^2 =$ `wtgain`$^1 \times$ `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
```

A CE variables

Table 11: Some CE variables and their missing rates (if any)

Name	Definition	Missing
AGE_REF	Age of reference person	
AGE2	Age of spouse	0.44
ALCBEVCQ	Alcoholic beverages this quarter	
ALCBEVPQ	Alcoholic beverages last quarter	
ALLFULCQ	Fuel oil and other fuels this quarter	
ALLFULPQ	Fuel oil and other fuels last quarter	
APPARCQ	Apparel and services this quarter (MENBOYCQ + WOMGRLCQ + CHLDRNCQ + FOOTWRCQ + OTHAPLCQ)	
APPARPQ	Apparel and services last quarter (same composition as APPARCQ)	
AS_COMP1	Number of males age 16 and over in CU	
AS_COMP2	Number of females age 16 and over in CU	
AS_COMP3	Number of males age 2 through 15 in CU	
AS_COMP4	Number of females age 2 through 15 in CU	
AS_COMP5	Number of members under age 2 in CU	
BATHRMQ	Number of complete baths in this unit	
BBYDAYCQ	Babysitting and child day care this quarter	
BBYDAYPQ	Babysitting and child day care last quarter	
BEDROOMQ	Number of bedrooms in CU	0.01
BLS_URBN	Is this CU located in an urban or rural area? (1=urban, 2=rural)	
BUILDING	Which of these descriptions from the list best describes this building? (1–11)	
BUILT	Year property was built	0.23
BUSCREEN	Has household had business expenses that could be reimbursed? (1=yes, 2=no)	<0.01
CARTKNCQ	Cars and trucks, new (net outlay) this quarter	
CARTKNPQ	Cars and trucks, new (net outlay) last quarter	
CARTKUCQ	Cars and trucks, used (net outlay) this quarter	
CARTKUPQ	Cars and trucks, used (net outlay) last quarter	
CASHCOCQ	Cash contributions this quarter	
CASHCOPQ	Cash contributions last quarter	

CHILDAGE	Age of children of reference person (0=no children, 1=all children less than 6, 2=oldest child 6–11 and at least one child less than 6, 3=all children 6–11, 4=oldest child 12–17 and at least one child less than 12, 5=all children 12–17, 6=oldest child greater than 17 and at least one child less than 17, 7=all children greater than 17)	
CREDFINX	What was the total amount paid in finance, late charges, and interest for all cards in last month?	0.82
CREDITB	Could you tell me which range that best reflects the total amount owed on all major credit cards including store cards and gas cards? (1=0–499, 2=500–999, 3=1000-2499, 4=2500–9999, 5=10000–34999, 6=35K and over)	0.99
CREDITBX	Median bracket range of CREDITB	0.99
CREDITX	Total amount owed on all cards	0.81
CREDTYRX	Total amount owed on all cards one year ago today	0.90
CREDYR	Did you have any credit cards including store cards and gas cards one year ago today? (1=yes, 2=no)	0.99
CREDYRB	Range that best reflects the total amount owed on all major credit cards including store cards and gas cards one year ago today (1=0–499, 2=500–999, 3=1000–2499, 4=2500–9999, 5=10000–34999, 6=35K and over)	0.99
CREDYRBX	Median bracket range of CREDYRB	0.99
CUTENURE	Housing tenure (1=homeowner with mortgage, 2=homeowner without mortgage, 3=homeowner, mortgage not reported, 4=rented, 5=occupied without payment of rent, 6=student housing)	
DEFBENRP	Do you have a defined retirement plan, such as a pension, from an employer? (1=yes, 2=no)	0.77
DIRACC	Is access to the quarters direct or through another unit? (1=direct, 2=another)	0.04
DIVISION	Census division (1=New England, 2=Middle Atlantic, 3=East North Central, 4=West North Central, 5=South Atlantic, 6=East South Central, 7=West South Central, 8=Mountain, 9=Pacific)	0.07
DOMSRVCQ	Domestic services this quarter	
DMSXCCCQ	Domestic services excluding child care this quarter	
DMSXCCPQ	Domestic services excluding child care last quarter	
DOMSRVPQ	Domestic services last quarter	

EARNCOMP	Composition of earners (1=reference person only, 2=reference person and spouse, 3=reference person, spouse and others, 4=reference person and others, 5=spouse only, 6=spouse and others, 7=others, 8=no earners)	
ECARTKNC	Outlays for new vehicle purchases this quarter including down payment, principal and interest paid on loans, or if not financed, purchase amount	
ECARTKNP	Outlays for new vehicle purchases last quarter including down payment, principal and interest paid on loans, or if not financed, purchase amount	
ECARTKUC	Outlays for used vehicle purchases this quarter including down payment, principal and interest paid on loans, or if not financed, purchase amount	
ECARTKUP	Outlays for used vehicle purchases last quarter including down payment, principal and interest paid on loans, or if not financed, purchase amount	
EDUC_REF	Education of reference person (10=grades 1–8; 11=grades 9–12, no degree; 12=high school graduate; 13=some college, no degree; 14=Assocaite's degree in college; Bachelors degree; 16=Masters degree or professional/doctorate degree)	
EDUCA2	Education level of spouse (same levels as EDUC_REF)	0.44
EDUCACQ	Education this quarter	
EDUCAPQ	Education last quarter	
EENTMSCC	Miscellaneous entertainment outlays this quarter including photographic and sports equipment and boat and RV rentals	
EENTMSCP	Miscellaneous entertainment outlays last quarter including photographic and sports equipment and boat and RV rentals	
EENTRMTC	Total entertainment outlays this quarter including sound systems, sports equipment, toys, cameras, and down payments on boats and campers (FEEADMCQ + TVRDIOCQ + PETTOYCQ + EOTHENTC)	
EENTRMTP	Total entertainment outlays last quarter including sound systems, sports equipment, toys, cameras, and down payments on boats and campers (same composition as EENTRMTC)	

EHOUSNGC	Total housing outlays this quarter including maintenance, fuels, public services, household operations, house furnishings, and mortgage (lump sum home equity loan or line of credit home equity loan) principle and interest (ESHELTRC + UTILCQ + HOUSOPCQ + HOUSEQCQ)
EHOUSNGP	Total housing outlays last quarter including maintenance, fuels, public services, household operations, house furnishings, and mortgage (lump sum home equity loan or line of credit home equity loan) principle and interest (same composition as EHOUSNGC)
ELCTRCCQ	Electricity this quarter
ELCTRCPQ	Electricity last quarter
EMISCELC	Miscellaneous outlays this quarter including reduction of mortgage principal (lump sum home equity loan) on other property (MISCPQ + EMISCMTP)
EMISCELP	Miscellaneous outlays last quarter including reduction of mortgage principal (lump sum home equity loan) on other property (same composition as EMISCELC)
EMISCMTC	Mortgage principal outlays this quarter for other property
EMISCMTP	Mortgage principal outlays last quarter for other property
EMRTPNOC	Mortgage principal outlays this quarter for owned home
EMRTPNOP	Mortgage principal outlays last quarter for owned home
EMRTPNVC	Mortgage principal outlays this quarter for owned vacation home
EMRTPNVP	Mortgage principal outlays last quarter for owned vacation home
EMOTRVHC	Outlays for motored recreational vehicles this quarter
EMOTRVHP	Outlays for motored recreational vehicles last quarter
ENOMOTRC	Outlays for non-motored recreational vehicles this quarter
ENOMOTRP	Outlays for non-motored recreational vehicles last quarter
ENTERTCQ	Entertainment this quarter (FEEADMCQ + TVRDIOCQ + OTHEQPCQ)

ENTERTPQ	Entertainment last quarter (same composition as ENTERTCQ)	
EOTHENTC	Outlays for other entertainment supplies this quarter, equipment, and services including down payments on boats and campers (ENOMOTRC + EMOTRVHC + EENTMSCC)	
EOTHENTP	Outlays for other entertainment supplies last quarter, equipment, and services including down payments on boats and campers (same composition as EOTHENTC)	
EOTHLODC	Outlays for other lodging this quarter such as owned vacation home, including mortgage principal and interest, property taxes, maintenance, insurance, and other expenses (OTHLODCQ + EMRTPNVC)	
EOTHLODP	Outlays for other lodging last quarter such as owned vacation home, including mortgage principal and interest, property taxes, maintenance, insurance, and other expenses (same composition as EOTHLODC)	
EOTHVEHP	Outlays for other vehicle purchases last quarter such as motorcycles and airplanes including down payment, principal and interest paid on loans, or if not financed, purchase amount	
EOTHVEHC	Outlays for other vehicle purchases this quarter such as motorcycles and airplanes including down payment, principal and interest paid on loans, or if not financed, purchase amount	
EOTHVEHP	Outlays for other vehicle purchases last quarter such as motorcycles and airplanes including down payment, principal and interest paid on loans, or if not financed, purchase amount	
EOWNDWLC	Owned home outlays this quarter including mortgage principal and interest, property taxes, maintenance, insurance, and other expenses (OWNDWECQ + EMRTPNOC)	
EOWNDWLP	Owned home outlays last quarter including mortgage principal and interest, property taxes, maintenance, insurance, and other expenses (same composition as EOWNDWLC)	
ERANKH	Percent expenditure outlay rank	0.08

ERANKHM	Weighted cumulative percent expenditure outlay ranking of CU to total population
ESHELTRC	Shelter outlays this quarter including mortgage principle and interest for owned home and/or vacation home, rents, insurance, taxes, and maintenance (EOWNDWLC + RENDWECQ + EOTHLODC)
ESHELTRP	Shelter outlays last quarter including mortgage principle and interest for owned home and/or vacation home, rents, insurance, taxes, and maintenance (same composition as ESHELTRC)
ETOTALC	Total outlays this quarter, sum of outlays from all major expenditure categories (FOODCQ + ALCBEVCQ + EHOUSNGC + APPARCQ + ETRANPTC + HEALTHCQ + EENTRMTC + PERSCACQ + READCQ + EDUCACQ + TOBACCCQ + EMISCELC + CASHCOCQ + PERINSCQ)
ETOTALP	Total outlays last quarter, sum of outlays from all major expenditure categories (same composition as ETOTALC)
ETOTACX4	Adjusted total outlays this quarter, sum of outlays from all major expenditure categories (FOODCQ + ALCBEVCQ + EHOUSNGC + APPARCQ + ETRANPTC + HEALTHCQ + EENTRMTC + PERSCACQ + READCQ + EDUCACQ + TOBACCCQ + MISC1CQ + 4×MISC2CQ + EMISCMTC + PERINSCQ)
ETOTAPX4	Adjusted total outlays last quarter, sum of outlays from all major expenditure categories (same composition as ETOTACX4)
ETRANPTC	Total outlays for transportation this quarter including down payment, principal and finance charges paid on loans, gasoline and motor oil, maintenance and repairs, insurance, public and other transportation, and vehicle rental licenses and other charges (EVEHPURC + GASMOCQ + MAINRPCQ + VEHINSCQ + VRNTLOCQ + PUBTRACQ)

ETRANPTP	Total outlays for transportation last quarter including down payment, principal and finance charges paid on loans, gasoline and motor oil, maintenance and repairs, insurance, public and other transportation, and vehicle rental licenses and other charges (same composition as ETRANPTC)
EVEHPURC	Outlays for vehicle purchases this quarter including down payment, principal and interest paid on loans, or if not financed, purchase amount (ECARTKNC + ECARTKUC + EOTHVEHC)
EVEHPURP	Outlays for vehicle purchases last quarter including down payment, principal and interest paid on loans, or if not financed, purchase amount (same composition as EVEHPURC)
FAM_SIZE	Number of Members in CU
FAM_TYPE	Family type (1–9)
FDAWAYCQ	Food away from home this quarter
FDAWAYPQ	Food away from home last quarter
FDHOMECQ	Food at home this quarter
FDHOMEPQ	Food at home last quarter
FDMAPCQ	Meals as pay this quarter
FDMAPPQ	Meals as pay last quarter
FDXMAPCQ	Food away excluding meals as pay this quarter
FDXMAPPQ	Food away excluding meals as pay last quarter
FEEADMCQ	Fees and admissions this quarter
FEEADMPQ	Fees and admissions last quarter
FFTAXOWE	Weighted estimate for Federal tax liabilities for entire CU
FGOVRETM	Amount of government retirement deducted from last pay, annualized for all CU members
FGOVRETX	Amount of government retirement deducted from last pay annualized
FINCBTAX	Total family income before taxes in last 12 months (INTRDVX + INTRDVBX + ROYESTX + ROYESTBX + OTHREGX + OTHREGBX + WELFAREX + WELFREBX + RETSURVX + RETSRVBX + NETRENTX + NETRNTBX + OTHRINCX)
FINDRETX	Money placed in self-employed retirement plan in past year for all CU members

FINLWT21	Sampling weight	
FJSSDEDX	Estimated amount contributed to Social Security by all CU members past 12 mos.	
FLRCVRCQ	Floor coverings this quarter	
FLRCVRPQ	Floor coverings last quarter	
FMLPYYRX	Annual value of free meals received as part of pay	0.99
FOODCQ	Total food this quarter	
FOODPQ	Total food last quarter	
FPRIPENM	Amount of private pensions deducted from last pay, annualized, for all CU members	
FPRIPENX	Amount of private pensions	
FRRDEDM	Amount of Railroad Retirement deducted from last pay, annualized for all CU members	
FRRDEDX	Amount of railroad retirement deducted from last pay annualized	
FRRETIRM	Amount of social security and railroad retirement income, prior to deductions for medical insurance and Medicare, received by all CU members in the past 12 months	
FRRETIRX	Social security and railroad retirement income	
FS_MTHI	In how many of the last 12 months were food stamps or EBTs received?	0.98
FSALARYX	Wage and salary income of all members past 12 mos.	
FSMPFRMX	Family level summation for new variable SEMPFRMX and SMPFRMBX	
FSSIX	Amount supplemental security income from all sources received by all CU members in past 12 months	
FSTAXOWE	Weighted estimate for State tax liabilities for entire CU	
FURNTRCQ	Furniture this quarter	
FURNTRPQ	Furniture last quarter	
GASMOCQ	Gasoline and motor oil this quarter	
GASMOPQ	Gasoline and motor oil last quarter	
FULOILCQ	Fuel oil this quarter	
FULOILPQ	Fuel oil last quarter	
FURNTRCQ	Furniture this quarter	
FURNTRPQ	Furniture last quarter	
HEALTHCQ	Health care this quarter (HLTHINCQ + MEDSRVCQ + PREDRGCQ + MEDSUPCQ)	

HEALTHPQ	Health care last quarter (same composition as HEALTHCQ)	
HIGH_EDU	Highest level of education within the CU (0=never attended, 10=1–8 grade, 11=9–12 grade, 12=HS grad, 13=some college, 14=AA degree, 15=Bachelors, 16=Masters/professional/doctorate)	
HISP_REF	Hispanic origin of reference person (1=Hispanic, 2=non-Hispanic)	
HISP2	Hispanic origin of spouse (1=Hispanic, 2=non-Hispanic)	
HH_CU_Q	Count of CUs in household	
HLFBATHQ	How many half bathrooms are there in this unit?	0.01
HLTHINCQ	Health insurance this quarter	
HLTHINPQ	Health insurance last quarter	
HORREF1	Hispanic origin of reference person (1=Mexican, 2=Mexican-American, 3=Chicano, 4=Puerto Rican, 5=Cuban, 6=Other)	0.96
HORREF2	Hispanic origin of spouse (same codes as HORREF1)	0.98
HOUSCQ	Housing this quarter	
HOUSEQCQ	House furnishings and equipment this quarter (TEXTILCQ + FURNTRCQ + FLRCVRCQ + MAJAPPCQ + SMLAPPCQ + MISCEQCQ)	
HOUSEQPQ	House furnishings and equipment last quarter (same composition as HOUSEQCQ)	
HOUSOPCQ	Household operations this quarter	
HOUSPQ	Housing last quarter	
HOUSOPPQ	Household operations last quarter	
INC_HRS1	Number hours worked per week by reference person	0.38
INC_HRS2	Number hours worked per week by spouse	0.66
INC_RANK	Income rank of CU to total population	
INCLASS2	Income class based on INC_RANK (1=0–0.1667, 2=0.1667–0.3333, 3=0.3334–0.4999, 4=0.5000–0.6666 5=0.6667–0.8333, 6=0.8334–1, 7=incomplete reporting)	
INCNONW1	Reason for not working during past 12 months (1=retired, 2=take care of home, 3=going to school, 4=ill, disabled, unable to work, 5=unable to find work, 6=doing something else)	0.62
INCNONW2	Reason spouse did not work during past 12 months (same codes as INCNONW1)	0.78

INCOMEY1	Employer paying most earnings in past 12 months (1=private company, business or individual, 2=Federal govt, 3=State govt, 4=local govt, 5=self-employed, 6=family business or farm, working without pay)	0.38
INCOMEY2	Employer from which spouse received most earnings during the past 12 months	0.66
INCWEEK1	Weeks worked full or part time in last 12 months	
INCWEEK2	Weeks worked by spouse full or part time last 12 months	0.44
INTRDVX	Amount received in interest or dividend during past 12 mos.	0.37
IRA	Do you have any retirement accounts such as 401(k)s, IRAs, thrift saving plans? (1=yes, 2=no)	0.76
IRAB	Range that best reflects the total value of all retirement accounts such as 401(k)s, IRAs, and thrift savings plans (1=0–1999, 2=2000-9999, 3=10K–49999, 4=50K–199999, 5=200K–449999, 6=450K or more)	0.97
IRAX	Total amount put into retirement accounts past 12 mos.	0.87
IRAYRB	Range which best reflects the total value of all retirement accounts one year ago today (same codes as IRAB)	0.96
IRAYRBX	Median value of bracket range for IRAYRB	0.96
IRAYRX	Total value of retirement accounts one year ago	0.88
JFS_AMT	Annual value of food stamps	
LIFINSCQ	Life and other personal insurance this quarter	
LIFINSPQ	Life and other personal insurance last quarter	
LIQDYRBX	Median value of bracket range for LIQUDYRB	0.96
LIQUDYR	Did you have any checking savings money market accounts, or CDs one year ago? (1=yes, 2=no)	>0.99
LIQUID	Do you have any checking, saving, money market accounts, or CDs? (1=yes, 2=no)	0.76
LIQUIDB	Range that best reflects total value of checking, savings, money market accounts, CDs (1=0–499, 2=500-999, 3=1000–2499, 4=2.5K–9999, 5=10K–34999, 6=35K and over)	0.97
LIQUIDBX	Median value of bracket range LIQUIDB	0.97
LIQUIDX	Total value of all checking, savings, money market, and CD accounts	0.83

LIQUDYRB	Range that best reflects the total value of all checking, savings, money market accounts, and CDs one year ago today (same codes as LIQUIDB)	0.97
LIQUDYRX	Total value of all checking, savings, money market accounts, and CDs one year ago today	0.84
MAINRPCQ	Maintenance and repairs this quarter	
MAINRPPQ	Maintenance and repairs last quarter	
MAJAPPCQ	Major appliances this quarter	
MAJAPPPQ	Major appliances last quarter	
MARITAL1	Marital status of reference person (1=married, 2=widowed, 3=divorced, 4=separated, 5=never married)	
MEALSPAY	Have you received any free meals at work as part of your pay? (1=yes, 2=no)	<0.01
MEDSRVCQ	Medical services this quarter	
MEDSRVPQ	Medical services last quarter	
MEDSUPCQ	Medical supplies this quarter	
MEDSUPPQ	Medical supplies last quarter	
MENBOYCQ	Clothing for men and boys this quarter	
MENSIXCQ	Clothing for men, 16 and over this quarter	
MENSIXPQ	Clothing for men, 16 and over last quarter	
MENBOYPQ	Clothing for men and boys last quarter	
MISC1CQ	Miscellaneous expenditures this quarter	
MISC1PQ	Miscellaneous expenditures last quarter	
MISCEQCQ	Miscellaneous household equipment this quarter	
MISCEQPQ	Miscellaneous household equipment last quarter	
MISCCQ	Miscellaneous expenditures this quarter (MISC1CQ + MISC2CQ)	
MISCPQ	Miscellaneous expenditures last quarter (same composition as MISCCQ)	
MISCTAXX	During past 12 months, what was total amount paid for personal property taxes and other taxes not reported elsewhere by all CU members?	0.99
MISCX4CQ	Adjusted miscellaneous expenditures this quarter (MISC1CQ + 4×MISC2CQ)	
MISCX4PQ	Adjusted miscellaneous expenditures last quarter (same composition as MISCX4CQ)	
MLPAYWKX	About what was the weekly dollar value of these meals?	0.99

MLPYQWKS	For how many weeks did members of your household receive these meals during the past 12 months?	0.99
MRPINSCQ	Maintenance, repairs, insurance, and other expenses this quarter	
MRPINSPQ	Maintenance, repairs, insurance, and other expenses last quarter	
MRTINTCQ	Mortgage interest this quarter	
MRTINTPQ	Mortgage interest last quarter	
MRTPRNOC	Outlays on owned vacation home mortgage principle this quarter	
MRTPRNOP	Outlays on owned vacation home mortgage principle last quarter	
NETRENTB	Range that best reflects the total net rental income or loss during the past 12 months (1=0–999, 2=1–2K, 3=2–3K, 4=3–4K, 5=4–5K, 6=5–10K, 7=10–15K, 8=15–20K, 9=20–30K, 10=30–40K, 11=40–50K, 12=50K and over)	0.99
NETRENTX	What was the amount of net rental income or loss?	0.92
NETRNTBX	Median value of bracket range of NETRENTB	0.99
NTLGASCQ	Natural gas this quarter	
NTLGASPQ	Natural gas last quarter	
NO_EARNR	Number of earners	
NONINCMX	Amount of other money receipts excluded from CU income before taxes received by CU in past 12 months	
NUM_AUTO	Total number of owned cars	
NUM_TVAN	Total number of owned trucks and vans	
OCCUCOD1	Highest paid occupation last 12 months (15 coded values)	0.38
OCCUCOD1	Job in which reference person received most earnings during past 12 months (15 coded values)	0.66
OCCUCOD2	Job in which spouse received most earnings during past 12 months (15 coded values)	0.66
OTHAPLCQ	Other apparel products and services this quarter	
OTHAPLPQ	Other apparel products and services last quarter	
OTHASTB	Range which best reflects the total value of these other financial assets (1=0–2K, 2=2–10K, 3=10–50K, 4=50–200K, 5=200–450K, 6=450K and over)	>0.99
OTHASTBX	Median value of bracket range for OTHASTB	>0.99
OTHASTX	Total value of these other financial assets as of today	0.99
OTHENTCQ	Other entertainment this quarter	

OTHENTPQ	Other entertainment last quarter	
OTHEQPCQ	Other equipment and services this quarter (PETTOYCQ + OTHENTCQ)	
OTHEQPPQ	Other equipment and services last quarter (same composition as OTHEQPCQ)	
OTHFINX	Total amount paid in finance, late charges, and interest for all other loans in the last month	0.99
OTHFLSCQ	Other fuels this quarter	
OTHFLSPQ	Other fuels last quarter	
OTHHEXCQ	Other household expenses this quarter	
OTHHEXPQ	Other household expenses last quarter	
OTHLNYR	Did you have any other debt such as medical loans or personal loans one year ago today? (1=yes, 2=no)	>0.99
OTHLNYRB	Range which best reflects the total amount owed on all other loans one year ago today (1=0–499, 2=500-999, 3=1–2.5K, 4=2.5–10K, 5=10–35K, 6=35K and over)	>0.99
OTHLODCQ	Other lodging this quarter	
OTHLODPQ	Other lodging last quarter	
OTHLONX	Total amount owed on all other loans	0.99
OTHLYRBX	Median value of bracket range for OTHLONBX	>0.99
OTHREGB	Range best reflects total amount received in Veteran's Administration (VA) payments, unemployment compensation, child support, or alimony during the past 12 months (1=0–1K, 2=1–2K, 3=2–3K, 4=3–4K, 5=4–5K, 6=5–10K, 7=10–15K, 8=15–20K, 9=20–30K, 10=30–40K, 11=40–50K, 12=50K and over)	0.99
OTHREGBX	Median value of bracket range for OTHREGB	0.99
OTHRINCX	Amount received in other income including money from care of foster children, cash scholarships and fellowships, or stipends not based on working	0.97
OTHREGX	Income on a regular basis from any other source such as Veteran's Administration (VA) payments, unemployment compensation, child support, or alimony	0.92
OTHSTYRB	Range which best reflects total value of these other financial assets one year ago today (1=0–2K, 2=2–10K, 3=10–50K, 4=50–200K, 5=200–450K, 6=450K and over)	>0.99
OTHSTYRX	Value of these other financial assets one year ago today	0.99
OTHSYRBX	Median value of bracket range for OTHSTYRB	>0.99

OTHVEHCQ	Other vehicles this quarter	
OTHVEHPQ	Other vehicles last quarter	
OWNDWECQ	Owned dwellings this quarter (MRTINTCQ + PROPTXCQ + MRPINSCQ)	
OWNDWEPQ	Owned dwellings last quarter (same composition as OWNDWECQ)	
OWNVACC	Expenditures on owned vacation homes this quarter including mortgage interest, insurance, taxes, maintenance, and miscellaneous household equipment (VOTHRLOC + VMISCHEC)	
OWNVACP	Expenditures on owned vacation homes last quarter including mortgage interest, insurance, taxes, maintenance, and miscellaneous household equipment (same composition as OWNVACC)	
PERINSCQ	Personal insurance and pensions this quarter (LIFINSCQ + RETPENCQ)	
PERINSPQ	Personal insurance and pensions last quarter (same composition as PERINSCQ)	
PERSCACQ	Personal care this quarter	
PERSCAPQ	Personal care last quarter	
PERSLT18	Number of CU members less than 18	
PERSOT64	Number of CU members over 64	
PETTOYCQ	Pets, toys, and playground equipment this quarter	
PETTOYPQ	Pets, toys, and playground equipment last quarter	
POPSIZE	Population size of the PSU (1=more than 5M, 2=1–5M, 3=0.5–1M, 4=100–500K, 5=less than 100K)	
PREDRGCQ	Prescription drugs this quarter	
PREDRGPQ	Prescription drugs last quarter	
PRINEARN	Member number of principal earner (5 coded values)	
PROPTXCQ	Property taxes this quarter	
PROPTXPQ	Property taxes last quarter	
PSU	Primary sampling unit	0.52
PUBTRACQ	Public and other transportation this quarter (TRNTRPCQ + TRNOTHCQ)	
PUBTRAPQ	Public and other transportation last quarter (same composition as PUBTRACQ)	
RACE2	Race of spouse (same codes as REF_RACE)	0.44
READCQ	Reading this quarter	

READPQ	Reading last quarter	
REF_RACE	Race of reference person (1=white, 2=black, 3=native American, 4=Asian, 5=Pacific islander, 6=multi-race)	
REFGEN	Generation of reference person (1=Greatest/Silent: born 1945 or earlier, 3=Baby boomers: 1946–64, 4=Gen X: 1965–80, 5=Millennials: 1981 or later)	
REGION	Region (1=Northeast, 2=Midwest, 3=South, 4=West)	0.01
RELECTRC	Expenditures on electricity for rented vacation homes this quarter	
RELECTRP	Expenditures on electricity for rented vacation homes last quarter	
RENDWECQ	Rented dwelling this quarter (RNTXRPCQ + RNTAPYCQ)	
RENDWEPQ	Rented dwelling last quarter (same composition as RENDWECQ)	
RENTEQVX	Monthly rent if home rented today	0.20
RETPENCQ	Retirement, pensions, social security this quarter	
RETPENPQ	Retirement, pensions, social security last quarter	
RETSRVBX	Median value of bracket range for RETSURVB	0.99
RETSURV	Did you receive income from retirement, survivor, or disability pensions during past 12 months? (1=yes, 2=no)	
RETSURVX	Retirement, survivor, disability pensions received past 12 mos.	0.78
RNATLGAC	Expenditures on natural gas for rented vacation homes this quarter	
RNATLGAP	Expenditures on natural gas for rented vacation homes last quarter	
RNTAPYCQ	Rent as pay this quarter	
RNTAPYPQ	Rent as pay last quarter	
RNTXRPCQ	Rent excluding rent as pay this quarter	
RNTXRPPQ	Rent excluding rent as pay last quarter	
ROOMSQ	Number of rooms in CU living quarters, including finished living areas, excluding all baths	0.01
ROTHRFLC	Expenditures on other fuels for rented vacation homes this quarter	

ROYESTB	Range that best reflects total amount received in royalty income or income from estates and trusts during past 12 months (1=0–1K, 2=1–2K, 3=2–3K, 4=3–4K, 5=4–5K, 6=5–10K, 7=10–15K, 8=15–20K, 9=20–30K, 10=30–40K, 11=40–50K, 12=50K and over)	>0.99
ROYESTBX	Median value of bracket range for ROYESTB	>0.99
ROYESTX	Amount received in royalty income or income from estates and trusts	0.96
RWATERPC	Expenditures on water and public services for rented vacation homes this quarter	
RWATERPP	Expenditures on water and public services for rented vacation homes last quarter	
SEX_REF	Sex of reference person (1=male, 2=female)	
SEX2	Sex of spouse (1=male, 2=female)	0.44
SHELTCQ	Shelter this quarter (OWNDWECQ + RENDWECQ + OTHLODCQ)	
SHELTPQ	Shelter last quarter (same composition as SHELTCQ)	
SMLAPPCQ	Small appliances, miscellaneous housewares this quarter	
SMLAPPPQ	Small appliances, miscellaneous housewares last quarter	
SMSASTAT	Does CU reside inside a Metropolitan Statistical Area (MSA)? (1=yes, 2=no)	
ST_HOUS	Are these living quarters presently used as student housing by a college or university? (1=yes, 2=no)	
STATE	1=AL, 2=AK, 4=AZ, 5=AR, 6=CA, 8=CO, 9=CT, 10=DE, 11=DC, 12=FL, 13=GA, 15=HI, 16=ID, 17=IL, 18=IN, 19=IA, 20=KS, 21=KY, 22=LA, 23=ME, 24=MD, 25=MA, 26=MI, 27=MN, 28=MS, 29=MO, 30=MT, 31=NE, 32=NV, 33=NH, 34=NJ, 36=NY, 37=NC, 39=OH, 40=OK, 41=OR, 42=PA, 44=RI, 45=SC, 46=SD, 47=TN, 48=TX, 49=UT, 51=VA, 53=WA, 54=WV, 55=WI	0.08
STCKYRBX	Median value of bracket range for STOCKYRB	0.98
STDNTYR	Did you have student loans one years ago today? (1=yes, 2=no)	>0.99
STDNTYRB	Range which best reflects the total amount owed on all student loans one year ago today (1=0–499, 2=500–999, 3=1–2.5K, 4=2.5–10K, 5=10–35K, 6=35K and over)	>0.99

STDNTYRX	Total amount owed on all student loans one year ago today	0.97
STDTYRBX	Median value of bracket range for STDNTYRB	>0.99
STOCKB	Range which best reflects total value of all directly-held stocks, bonds, and mutual funds (1=0–2K, 2=2–10K, 3=10–50K, 4=50–200K, 5=200–450K, 6=450K and over)	0.99
STOCKBX	Median value of bracket range for STOCKB	0.99
STOCKX	Value of directly-held stocks, bonds, mutual funds (median=59,950, mean=411,867)	0.93
STOCKYR	Did you have any directly-held stocks, bonds, or mutual funds one year ago? (1=yes, 2=no)	>0.99
STOCKYRB	Range which best reflects total value of all directly-held stocks, bonds, and mutual funds one year ago today (same codes as STOCKB)	0.98
STOCKYRX	Median value of bracket range of STOCKX	0.93
STUDFINX	Total amount paid in finance, late charges, and interest for all student loans in the last month	0.97
STUDNTB	Range which best reflects the total amount owed on all student loans (1=0–499, 2=500–999, 3=1–2.5K, 4=2.5–10K, 5=10–35K, 6=35K and over)	>0.99
STUDNTBX	Median value of bracket range for STUDNTB	>0.99
STUDNTX	Total amount owed on all student loans	0.97
TAIRFARC	Trip expenditures on airfare this quarter	
TAIRFARP	Trip expenditures on airfare last quarter	
TALCBEVC	Total trip expenditures this quarter on alcoholic beverages at restaurants, cafes, and bars	
TALCBEVP	Total trip expenditures last quarter on alcoholic beverages at restaurants, cafes, and bars	
TELEPHCQ	Telephone services this quarter	
TELEPHPQ	Telephone services last quarter	
TENTRMNC	Total trip expenditures on entertainment this quarter including sporting events, movies, and recreational vehicle rentals (TFEESADC + TOTHENTC)	
TENTRMNP	Total trip expenditures on entertainment last quarter including sporting events, movies, and recreational vehicle rentals (same composition as TENTRMNC)	
TEXTILCQ	Household textiles this quarter	
TEXTILPQ	Household textiles last quarter	

TFAREC	Trip expenditures this quarter on transportation fares including airfare, intercity bus, train, and ship fare (TAIRFARC + TOTHFARC)
TFAREP	Trip expenditures last quarter on transportation fares including airfare, intercity bus, train, and ship fare (same composition as TFAREC)
TFEESADC	Trip expenditures on miscellaneous entertainment this quarter including recreation expenses, participation sport fees, and admission fees to sporting events and movies
TFEESADP	Trip expenditures on miscellaneous entertainment last quarter including recreation expenses, participation sport fees, and admission fees to sporting events and movies
TFOODAWC	Food and non-alcoholic beverages this quarter at restaurants, cafes, and fast food places during out-of-town trips
TFOODAWP	Food and non-alcoholic beverages last quarter at restaurants, cafes, and fast food places during out-of-town trips
TFOODHOC	Food and beverages purchased and prepared by CU this quarter during out-of-town trips
TFOODHOP	Food and beverages purchased and prepared by CU last quarter during out-of-town trips
TFOODTOC	Total trip expenditures on food this quarter including both restaurant food and food prepared by CU (TFOODAWC + TFOODHOC)
TFOODTOP	Total trip expenditures on food last quarter including both restaurant food and food prepared by CU (same composition as TFOODTOC)
TGASMOTC	Trip expenditures on gas and oil this quarter
TGASMOTP	Trip expenditures on gas and oil last quarter
TLOCALTC	Trip expenditures this quarter on local transportation including taxis, buses etc.
TLOCALTP	Trip expenditures last quarter on local transportation including taxis, buses etc.
TOBACCCQ	Tobacco and smoking supplies this quarter
TOBACCPQ	Tobacco and smoking supplies last quarter
TOTEX4CQ	Adjusted total expenditures this quarter (TOTEXPCQ - MISCCQ + MISC1CQ + 4 × MISC2CQ)
TOTEX4PQ	Adjusted total expenditures last quarter (same composition as TOTEX4CQ)

TOTEXPCQ	Total expenditures this quarter (FOODCQ + ALCBEVCQ + HOUSCQ + APPARCQ + TRANSCQ + HEALTHCQ + ENTERTCQ + PERSCACQ + READCQ + EDUCACQ + TOBACCCQ + MISCCQ + CASHCOCQ + PERINSCQ)
TOTEXPPQ	Total expenditures last quarter (same composition as TOTEXPCQ)
TOTHENTC	Trip expenditures on recreational vehicle rentals this quarter including campers, boats, and other vehicles
TOTHENTP	Trip expenditures on recreational vehicle rentals last quarter including campers, boats, and other vehicles
TOTHFARC	Tip expenditures this quarter on other transportation fares including intercity bus and train fare, and ship fare
TOTHFARP	Tip expenditures last quarter on other transportation fares including intercity bus and train fare, and ship fare
TOTHRLOC	Total trip expenditures on lodging this quarter including rent for vacation home, and motels
TOTHRLOP	Total trip expenditures on lodging last quarter including rent for vacation home, and motels
TOTHTREC	Trip expenditures this quarter for other transportation expenses including parking fees, and tolls
TOTHTREP	Trip expenditures last quarter for other transportation expenses including parking fees, and tolls
TOTHVHRC	Trip expenditures on other vehicle rentals this quarter
TOTHVHRP	Trip expenditures on other vehicle rentals last quarter
TOTXEST	Estimated total taxes paid (FFTAXOWE + FSTAXOWE + MISCTAXX)
TRANSCQ	Transportation this quarter (CARTKNCQ + CARTKUCQ + OTHVEHCQ + GASMOCQ + VEHFINCQ + MAINRPCQ + VEHINSCQ + VRNTLOCQ + PUBTRACQ)
TRANSPQ	Transportation last quarter (same composition as TRANSCQ)
TRNOTHCQ	Local public transportation, excluding on trips this quarter
TRNOTHPQ	Local public transportation, excluding on trips last quarter
TRNTRPCQ	Public and other transportation on trips this quarter

TRNTRPPQ	Public and other transportation on trips last quarter
TTOTALC	Total of all trip expenditures this quarter (TFOODTOC + TALCBEVC + TOTHRLOC + TTRANPRC + TENTRMNPC)
TTOTALP	Total of all trip expenditures last quarter (same composition as TTOTALC)
TTRANPRC	Total trip expenditures on transportation this quarter including airfare, local transportation, tolls and parking fees, and car rentals (TGASMOTC + TVRENTLC + TTRNTRIC)
TTRANPRP	Total trip expenditures on transportation last quarter including airfare, local transportation, tolls and parking fees, and car rentals (same composition as TTRANPRC)
TTRNTRIC	Trip expenditures this quarter for public transportation, including airfares (TFAREC + TLOCALTC)
TTRNTRIP	Trip expenditures last quarter for public transportation, including airfares (same composition as TTRNTRIC)
TVRDIOCQ	Televisions, radios, and sound equipment this quarter
TVRDIOPQ	Televisions, radios, and sound equipment last quarter
TVRENTLC	Trip expenditures on vehicle rentals and other fees this quarter (TCARTRKC + TOTHVHRC + TOTHTREC)
TVRENTLP	Trip expenditures on vehicle rentals and other fees last quarter (same composition as TVRENTLC)
UNISTRQ	How many housing units, both occupied and vacant, are in this structure? (1=only other units, 2=mobile home or trailer, 3=one, detached, 4=one, attached, 5=2, 6=3–4, 7=5–9, 8=10–19, 9=20–49, 10=50 or more)
UTILCQ	Utilities, fuels and public services this quarter (NTLGASCQ + ELCTRCCQ + ALLFULCQ + TELEPHCQ + WATRPSCQ)
UTILOWNC	Expenditures on owned vacation home utilities this quarter including water, trash, electricity, and fuels (VFUELOIC + VOTHRFLC + VELECTRC + VNATLGAC + VWATERPC)
UTILOWNP	Expenditures on owned vacation home utilities last quarter including water, trash, electricity, and fuels (same composition as UTILOWNC)
UTILPQ	Utilities, fuels and public services last quarter

UTILRNTC	Expenditures on rented vacation home utilities this quarter including water, trash, electricity, and fuels (RFUELOIC + ROTHRFLC + RELECTRC + RNATLGAC + RWATERPC)	
UTILRNTP	Expenditures on rented vacation home utilities last quarter including water, trash, electricity, and fuels (same composition as UTILRNTC)	
VEHFINCQ	Vehicle finance charges this quarter	
VEHFINPQ	Vehicle finance charges last quarter	
VEHICTAX	Personal property taxes for vehicles	0.92
VEHINSCQ	Vehicle insurance this quarter	
VEHINSPQ	Vehicle insurance last quarter	
VEHQ	Total number of owned vehicles	
VEHQL	Total number of leased autos, trucks and vans	
VELECTRC	Expenditures on electricity for owned vacation homes this quarter	
VELECTRP	Expenditures on electricity for owned vacation homes last quarter	
VFUELOIC	Expenditures on fuel oil for owned vacation homes this quarter	
VFUELOIP	Expenditures on fuel oil for owned vacation homes last quarter	
VNATLGAC	Expenditures on natural gas for owned vacation homes this quarter	
VNATLGAP	Expenditures on natural gas for owned vacation homes last quarter	
VOTHRFLC	Expenditures on other fuels for owned vacation homes this quarter	
VOTHRFLP	Expenditures on other fuels for owned vacation homes last quarter	
VOTHRLOP	Expenditures on owned vacation homes last quarter including mortgage interest, insurance, taxes, and maintenance	
VOTHRLOC	Expenditures on owned vacation homes this quarter including mortgage interest, insurance, taxes, and maintenance	
VRNTLOCQ	Vehicle rental, leases, licenses, and other charges this quarter	

VRNTLOPQ	Vehicle rental, leases, licenses, and other charges last quarter	
VWATERPC	Expenditures on water and public services for owned vacation homes this quarter	
VWATERPP	Expenditures on water and public services for owned vacation homes last quarter	
WATRPSCQ	Water and other public services this quarter	
WATRPSPQ	Water and other public services last quarter	
WELFAREX	Amount received from public assistance or welfare including money received from job training grants	0.99
WELFREBX	Median of bracket range of WELFAREB	0.99
WHLFYR	Did you own any whole life insurance or other life insurance policies that can be surrendered for cash or borrowed against prior to the death of the person insured one year ago today? (1=yes, 2=no)	>0.99
WHLFYRB	Range which best reflects total surrender value of these policies one year ago today (1=0–499, 2=500–999, 3=1–2.5K, 4=2.5–10K, 5=10–35K, 6=35K and over)	0.99
WHLFYRBX	Median value of bracket range for WHLFYRB	0.99
WHLFYRX	Total surrender value of these policies one year ago today	0.98
WHOLIFB	Range which best reflects the total surrender value of these policies (same codes as WHLFYRB) >0.99	
WHOLIFBX	Median value of bracket range for WHOLIFB	>0.99
WHOLIFX	Total surrender value of these policies as of today	0.98

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.

Table 12: PSU codes

S11A	Boston-Cambridge-Newton, MA-NH
S12A	New York-Newark-Jersey City, NY-NJ-PA
S12B	Philadelphia-Camden-Wilmington, PA-NJ-DE-MD
S23A	Chicago-Naperville-Elgin, IL-IN-WI
S23B	Detroit-Warren-Dearborn, MI
S24A	Minneapolis-St. Paul-Bloomington, MN-WI
S24B	St. Louis, MO-IL
S35A	Washington-Arlington-Alexandria, DC-VA-MD-WV
S35B	Miami-Fort Lauderdale-West Palm Beach, FL
S35C	Atlanta-Sandy Springs-Roswell, GA
S35D	Tampa-St. Petersburg-Clearwater, FL
S35E	Baltimore-Columbia-Towson, MD
S37A	Dallas-Fort Worth-Arlington, TX
S37B	Houston-The Woodlands-Sugar Land, TX
S48A	Phoenix-Mesa-Scottsdale, AZ
S48B	Denver-Aurora-Lakewood, CO
S49A	Los Angeles-Long Beach-Anaheim, CA
S49B	San Francisco-Oakland-Hayward, CA
S49C	Riverside-San Bernardino-Ontario, CA
S49D	Seattle-Tacoma-Bellevue, WA
S49E	San Diego-Carlsbad, CA
S49F	Honolulu, HI
S49G	Anchorage, AK

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth, Belmont.

Broekman, B. F. P., Niti, M., Nyunt, M. S. Z., Ko, S. M., Kumar, R., and Ng, T. P. (2011). Validation of a brief seven-item response bias-free geriatric depression scale. *American Journal of Geriatric Psychiatry*, 19:589–596.

Broekman, B. F. P., Nyunt, S. Z., Niti, M., Jin, A. Z., Ko, S. M., Kumar, R., Fones, C. S. L., and Ng, T. P. (2008). Differential item functioning of the geriatic depression scale in an Asian population. *Journal of Affective Disorders*, 108:285–290.

Cameron, A. A. and Trivedi, P. K. (1998). *Regression Analysis of Count Data*. Cambridge University Press.

Chambers, J. M. and Hastie, T. J. (1992). An appetizer. In Chambers, J. M. and Hastie, T. J., editors, *Statistical Models in S*, pages 1–12. Wadsworth & Brooks/Cole, Pacific Grove.

Chan, K.-Y. and Loh, W.-Y. (2004). LOTUS: An algorithm for building accurate and comprehensible logistic regression trees. *Journal of Computational and Graphical Statistics*, 13:826–852. http://www.stat.wisc.edu/~loh/treeprogs/lotus/lotus.pdf.

Chaudhuri, P., Huang, M.-C., Loh, W.-Y., and Yao, R. (1994). Piecewise-polynomial regression trees. *Statistica Sinica*, 4:143–167. http://www3.stat.sinica.edu.tw/statistica/j4n1/j4n18/j4n18.htm.

Chaudhuri, P., Lo, W.-D., Loh, W.-Y., and Yang, C.-C. (1995). Generalized regression trees. *Statistica Sinica*, 5:641–666. http://www3.stat.sinica.edu.tw/statistica/j5n2/j5n217/j5n217.htm.

Chaudhuri, P. and Loh, W.-Y. (2002). Nonparametric estimation of conditional quantiles using quantile regression trees. *Bernoulli*, 8:561–576. http://www.stat.wisc.edu/~loh/treeprogs/guide/quantile.pdf.

Chen, P. Y. and Tsiatis, A. A. (2001). Causal inference on the difference of the restricted mean lifetime between two groups. *Biometrics*, 57:1030–1038.

Choi, Y., Ahn, H., and Chen, J. J. (2005). Regression trees for analysis of count data with extra Poisson variation. *Computational Statistics & Data Analysis*, 49(3):893–915.

Connors, Jr., A. F., , Speroff, T., Dawson, N. V., et al. (1996). The effectiveness of right heart catheterization in the initial care of critically ill patients. *JAMA*, 276(11):889–897.

Deb, P. and Trivedi, P. K. (1997). Demand for medical care by the elderly: a finite mixture approach. *Journal of Applied Econometrics*, 12:313–336.

Hothorn, T. (2017). *TH.data: TH's Data Archive*. R package version 1.0-8.

Hothorn, T., Hornik, K., and Zeileis, A. (2006). Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15:651–674.

Hothorn, T. and Zeileis, A. (2015). partykit: A modular toolkit for recursive partytioning in r. *Journal of Machine Learning Research*, 16:3905–3909.

Italiano, A. (2011). Prognostic or predictive? It's time to get back to definitions! *Journal of Clinical Oncology*, 29:4718.

Kim, H. and Loh, W.-Y. (2001). Classification trees with unbiased multiway splits. *Journal of the American Statistical Association*, 96:589–604. http://www.stat.wisc.edu/~loh/treeprogs/cruise/cruise.pdf.

Kim, H. and Loh, W.-Y. (2003). Classification trees with bivariate linear discriminant node models. *Journal of Computational and Graphical Statistics*, 12:512–530. http://www.stat.wisc.edu/~loh/treeprogs/cruise/jcgs.pdf.

Kim, H., Loh, W.-Y., Shih, Y.-S., and Chaudhuri, P. (2007). Visualizable and interpretable regression models with good prediction power. *IIE Transactions*, 39:565–579. http://www.stat.wisc.edu/~loh/treeprogs/guide/iie.pdf.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

Loh, W.-Y. (1987). Calibrating confidence coefficients. *Journal of the American Statistical Association*, 82:155–162.

Loh, W.-Y. (1991). Bootstrap calibration for confidence interval construction and selection. *Statistica Sinica*, 1:477–491.

Loh, W.-Y. (2002). Regression trees with unbiased variable selection and interaction detection. *Statistica Sinica*, 12:361–386. http://www3.stat.sinica.edu.tw/statistica/j12n2/j12n21/j12n21.htm.

Loh, W.-Y. (2006a). Logistic regression tree analysis. In Pham, H., editor, *Handbook of Engineering Statistics*, pages 537–549. Springer. http://www.stat.wisc.edu/~loh/treeprogs/lotus/springer.pdf.

Loh, W.-Y. (2006b). Regression tree models for designed experiments. In Rojo, J., editor, *The Second Erich L. Lehmann Symposium–Optimality*, volume 49, pages 210–228. Institute of Mathematical Statistics Lecture Notes-Monograph Series. arxiv.org/abs/math.ST/0611192.

Loh, W.-Y. (2008a). Classification and regression tree methods. In Ruggeri, F., Kenett, R., and Faltin, F. W., editors, *Encyclopedia of Statistics in Quality and Reliability*, pages 315–323. Wiley, Chichester, UK. http://www.stat.wisc.edu/~loh/treeprogs/guide/eqr.pdf.

Loh, W.-Y. (2008b). Regression by parts: Fitting visually interpretable models with GUIDE. In Chen, C., Härdle, W., and Unwin, A., editors, *Handbook of Computational Statistics*, pages 447–469. Springer. http://www.stat.wisc.edu/~loh/treeprogs/guide/handbk.pdf.

Loh, W.-Y. (2009). Improving the precision of classification trees. *Annals of Applied Statistics*, 3:1710–1737. http://www.stat.wisc.edu/~loh/treeprogs/guide/aoas260.pdf.

Loh, W.-Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1:14–23. http://www.stat.wisc.edu/~loh/treeprogs/guide/wires11.pdf.

Loh, W.-Y. (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*, 30:1697–1722. http://www.stat.wisc.edu/~loh/treeprogs/guide/LZZZ20.pdf.

Loh, W.-Y. and Zheng, W. (2013). Regression trees for longitudinal and multiresponse data. *Annals of Applied Statistics*, 7:495–522. http://www.stat.wisc.edu/~loh//treeprogs/guide/AOAS596.pdf.

Loh, W.-Y. and Zhou, P. (2020). The GUIDE approach to subgroup identification. In Ting, N., Cappelleri, J. C., Ho, S., and Chen, D.-G., editors, *Design and analysis of Subgroups with Biopharmaceutical Applications*, pages 147–165. Springer. http://www.stat.wisc.edu/~loh/treeprogs/guide/LZ20.pdf.

Loh, W.-Y. and Zhou, P. (2021). Variable importance scores. *Journal of Data Science*, 19(4):569–592. http://www.stat.wisc.edu/~loh/treeprogs/guide/LZ21.pdf.

Marc, L. G., Raue, P. J., and Bruce, M. L. (2008). Screening performance of the 15-item geriatric depression scale in a diverse elderly home care population. *American Journal of Geriatric Psychiatry*, 16:914–921.

Murnane, R. J., Boudett, K. P., and Willett, J. B. (1999). Do male dropouts benefit from obtaining a GED, postsecondary education, and training? *Evaluation Reviews*, 23:475–502.

Quinlan, J. R. (1992). Learning with continuous classes. In *5th Australian Joint Conference on Artificial Intelligence*, pages 343–348.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Schmoor, C., Olschewski, M., and Schumacher, M. (1996). Randomized and non-randomized patients in clinical trials: experiences with comprehensive cohort studies. *Statistics in Medicine*, 15:263–271.

Singer, J. D. and Willett, J. B. (2003). *Applied Longitudinal Data Analysis*. Oxford University Press, New York, NY.

Therneau, T., Atkinson, B., and Ripley, B. (2017). *rpart: Recursive Partitioning and Regression Trees*. CRAN.R-project.org/package=rpart.

Tian, L., Zhao, L., and Wei, L. J. (2014). Predicting the restricted mean event time with the subject's baseline covariates in survival analysis. *Biostatistics*, 15:222–233.

Witten, I. and Frank, E. (2000). *Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations*. Morgan Kaufmann, San Fransico, CA. http://www.cs.waikato.ac.nz/ml/weka.

Zeileis, A. (2006). Object-oriented computation of sandwich estimators. *Journal of Statistical Software*, 16(9):1–16.