

# Package ‘rpms’

October 14, 2022

**Type** Package

**Title** Recursive Partitioning for Modeling Survey Data

**Version** 0.5.1

**Date** 2021-06-16

**Maintainer** Daniell Toth <danielltoth@yahoo.com>

**Description** Functions to allow users to build and analyze design consistent tree and random forest models using survey data from a complex sample design. The tree model algorithm can fit a linear model to survey data in each node obtained by recursively partitioning the data. The splitting variables and selected splits are obtained using a randomized permutation test procedure which adjusted for complex sample design features used to obtain the data. Likewise the model fitting algorithm produces design-consistent coefficients to any specified least squares linear model between the dependent and independent variables used in the end nodes. The main functions return the resulting binary tree or random forest as an object of “rpms” or “rpms\_forest” type. The package also provides methods modeling a “boosted” tree or forest model and a tree model for zero-inflated data as well as a number of functions and methods available for use with these object types.

**License** CC0

**Depends** R (>= 2.10)

**Imports** Rcpp (>= 0.12.3), stats

**LinkingTo** Rcpp, RcppArmadillo

**Suggests** parallel

**RoxygenNote** 7.1.1

**Encoding** UTF-8

**NeedsCompilation** yes

**LazyData** true

**Author** Daniell Toth [aut, cre]

**Repository** CRAN

**Date/Publication** 2021-06-25 23:40:02 UTC

## R topics documented:

rpms-package	2
boxes	3
box_ind	3
CE	4
end_nodes	6
grow_rpms	7
in_node	8
linearize	9
node_plot	9
predict.rpms	10
predict.rpms_boost	11
predict.rpms_forest	12
predict.rpms_proj	12
predict.rpms_zinf	13
print.rpms	13
print.rpms_forest	14
print.rpms_zinf	14
prune_rpms	15
qtree	15
r2stat	16
rpms	17
rpms_boost	18
rpms_forest	19
rpms_proj	20
rpms_zinf	21
<b>Index</b>	<b>23</b>

---

 rpms-package

*Recursive Partitioning for Modeling Survey Data (rpms)*


---

### Description

This package provides a function `rpms` to produce an `rpms` object and method functions that operate on them. The `rpms` object is a representation of a regression tree achieved by recursively partitioning the dataset, fitting the specified linear model on each node separately. The recursive partitioning algorithm has an unbiased variable selection and accounts for the sample design. The algorithm accounts for one-stage of stratification and clustering as well as unequal probability of selection. There are also functions for producing random forest estimator (a list of `rpms` objects), a boosted regression tree and tree based zero-inflated model.

---

boxes	<i>boxes</i>
-------	--------------

---

**Description**

returns end boxes that partition the data

**Usage**

```
boxes(x)
```

**Arguments**

x                    rpms object

**Value**

data.frame including end\_node, sample size, splits, and values for each end node

---

box_ind	<i>box_ind</i>
---------	----------------

---

**Description**

For each row of data, returns a vector indicators whether observation is in that box or not

**Usage**

```
box_ind(x, newdata)
```

**Arguments**

x                    rpms object  
newdata            dataframe containing the variables used for the recursive partitioning.

**Value**

Matrix where each row is a vector of indicators whether observation is in box or not.

CE

*CE Consumer expenditure data 2015***Description**

A dataset containing consumer unit characteristics, assets and expenditure data from the Bureau of Labor Statistics' Consumer Expenditure Survey public use interview data file.

**Usage**

CE

**Format**

A data frame with 68,415 observations on 47 variables:

**Sample-design information**

**NEWID** Consumer unit identifying variable, constructed using the first seven digits of NEWID BLS derived

**PSU** Primary Sampling Unit code for the 21 biggest clusters

**CID** Cluster Identifier for all clusters, (created using PSU, REGION, STATE, and POPSIZE) not part of CE data

**QINTRVMO** Month for which data was collected

**FINLWT21** Final sample weight to make inference to total population

**Location of Consumer Unit**

**STATE** State FIPS code

**REGION** Region code: 1 Northeast; 2 Midwest; 3 South; 4 West

**BLS\_URBN** Urban = 1, Rural = 2

**POPSIZE** Population size class of PSU: 1-biggest 5-smallest

**Housing and transportation**

**CUTENURE** Housing tenure: 1 Owned with mortgage; 2 Owned without mortgage 3 Owned mortgage not reported; 4 Rented; 5 Occupied without payment of cash rent; 6 Student housing

**ROOMSQ** Number of rooms, including finished living areas and excluding all baths

**BATHRMQ** Number of bathrooms

**BEDROOMQ** Number of bedrooms

**VEHQ** Number of owned vehicles

**VEHQL** Number of leased vehicles

**Family Information**

**FAM\_TYPE** CU code based on relationship of members to reference person (children include blood-related, step and adopted): 1 Married Couple only; 2 Married Couple, children (oldest < 6 years old); 3 Married Couple, children (oldest 6 to 17 years old); 4 Married Couple, children (oldest > 17 years old); 5 All other Married Couple CUs 6 One parent (male), children (at least one child < 18 years old); 7 One parent (female), children (at least one child < 18 years old); 8 Single consumers; 9 Other CUs

**FAM\_SIZE** Number of members in CU

**PERSLT18** Number of people <18 yrs old

**PERSOT64** Number of people >64 yrs old

**NO\_EARNR** Number of earners

**Primary Earner Information**

**AGE** Age of primary earner

**EDUCA** Education level coded: 1 None; 2 1st-8th Grade; 3 some HS; 4 HS; 5 Some college; 6 AA degree; 7 Bachelors degree; 8 Advanced degree

**SEX** Gender Code: F (Female); M (Male)

**MARITAL** Marital Status Coded: 1 Married; 2 Widowed; 3 Divorced; 4 Separated; 5 Never Married

**MEMBRACE** Race code: 1 White; 2 Black; 3 Native American; 4 Asian; 5 Pacific Islander; 6 Multi-race

**HORIGIN** Hispanic, Latino, or Spanish origin? Y (Yes); N (No)

**ARM\_FORC** Member of armed forces? Y (Yes); N (No)

**IN\_COLL** Currently enrolled in college? Full (full time); Part (part time); No

**Labor Status of Primary Earner**

**EARNER** Earn income: Y (Yes); N (No)

**EARNTYPE** 1 Full time all year; 2 Part time all year; 3 Full time part of the year; 2 Part time part of the year;

**OCCUCODE** The job in which the member received the most earnings during the past 12 months fits best in the following category: 01 Administrator, manager; 02 Teacher; 03 Professional Administrative support, technical, sales; 04 Administrative support, including clerical; 05 Sales, retail; 06 Sales, business goods and services; 07 Technician; 08 Protective service; 09 Private household service; 10 Other service; 11 Machine operator, assembler, inspector; 12 Transportation operator; 13 Handler, helper, laborer; 14 Mechanic, repairer, precision production; 15 Construction, mining; 16 Farming; 17 Forestry, fishing, grounds-keeping; 18 Armed forces

**INCOMEY** Type of employment: 1 An employee of a PRIVATE company, business, or individual 2 A Federal government employee 3 A State government employee 4 A local government employee 5 Self-employed in OWN business, professional practice or farm 6 Working WITHOUT PAY in family business or farm

**INCNONWK** Reason did not work during the past 12 months: 1 Retired; 2 Home maker; 3 School; 4 health; 5 Unable to find work; 6 Doing something else

**Income**

**FINCBTAX** Amount of CU income before taxes in past 12 months

**SALARYX** Amount of wage or salary income received in past 12 months, before any deductions

**SOCRRX** Amount income received from Social Security and Railroad Retirement in past 12 months

**Assets and Liabilities**

**IRAX** Total value of all retirement accounts

**LIQUIDX** Value of liquid assets

**STOCKX** Total value of all directly-held stocks, bonds

**STUDNTX** Amount owed on all student loans

**Expenditures**

**TOTEXPCQ** Total expenditures for current quarter

**TOTXEST** Total taxes paid (estimated)

**EHOUSNGC** Total expenditures for housing paid this quarter

**HEALTHCQ** Expenditures on health care quarter

**FOODCQ** Expenditure on food this quarter

**TOBACCCQ** Tobacco and smoking supplies this quarter

**FOOTWRCQ** Expenditure on footwear1 this quarter

end describe

**Source**

[https://www.bls.gov/cex/pumd\\_data.htm](https://www.bls.gov/cex/pumd_data.htm)

---

end\_nodes

*end\_nodes*

---

**Description**

Either a vector of end-node labels for each opbservation in newdata or a vector of the endnodes in the tree model if newdata is not provided.

**Usage**

```
end_nodes(object, newdata = NULL)
```

**Arguments**

object	rpms object
newdata	data.frame

**Value**

vector of end\_node labels

**Examples**

```
{
# model mean of retirement account value for households with reported
# retirement account values > 0 using a binary tree while accounting for
# clustered data and sample weights.

s1<- which(CE$IRAX > 0)
r1 <-rpms(IRAX~EDUCA+AGE+BLS_URBN, data = CE[s1,], weights=~FINLWT21, clusters=~CID)

  end_nodes(r1)
}
```

---

grow\_rpms

*grow\_rpms*

---

**Description**

grow an rpms tree from a given node

**Usage**

```
grow_rpms(
  x,
  node,
  data,
  weights = ~1,
  strata = ~1,
  clusters = ~1,
  pval = NA,
  bin_size = NA
)
```

**Arguments**

x	rpms object
node	node from which to grow tree further
data	data.frame that includes variables used in rp_equ, e_equ, and design information
weights	formula or vector of sample weights for each observation
strata	formula or vector of strata labels
clusters	formula or vector of cluster labels
pval	numeric p-value used to reject null hypothesis in permutation test
bin_size	numeric minimum number of observations in each node

**Value**

rpms tree expanded from node.

---

in_node	<i>in_node</i>
---------	----------------

---

**Description**

Get index of elements in dataframe that are in the specified end-node of an rpms object. A "which" function for end-nodes.

**Usage**

```
in_node(x, node, data)
```

**Arguments**

x	rpms object
node	integer label of the desired end-node.
data	dataframe containing the variables used for the recursive partitioning.

**Value**

vector of indexes for observations in the end-node.

**Examples**

```
{
# model mean of retirement account value for households with reported
# retirement account values > 0 using a binary tree while accounting for
# clustered data and sample weights.

s1<- which(CE$IRAX > 0)
r1 <-rpms(IRAX~EDUCA+AGE+BLS_URBN, data = CE[s1,], weights=~FINLWT21, clusters=~CID)

# Get summary statistics of CUTENURE for households in end-nodes 7 and 8 of the tree

if(7 %in% end_nodes(r1))
  summary(CE$CUTENURE[in_node(node=7, r1, data=CE[s1,])])
if(8 %in% end_nodes(r1))
  summary(CE$CUTENURE[in_node(node=8, r1, data=CE[s1,])])
}
```



---

linearize	<i>linearize</i>
-----------	------------------

---

**Description**

returns a linerized version of the splits. The coefficients represent the effect that each split has on the mean

**Usage**

```
linearize(x, data, weights = ~1, strata = ~1, clusters = ~1, type = "part")
```

**Arguments**

x	rpms object
data	data.frame
weights	formula or vector of sample weights for each observation
strata	formula or vector of strata labels
clusters	formula or vector of cluster labels
type	is on of "part" or "lin"

**Value**

data.frame including splits and estimates for the coefficient and their standard errors

---

node_plot	<i>node_plot</i>
-----------	------------------

---

**Description**

plots end-node of object of class rpms

**Usage**

```
node_plot(object, node, data, variable = NA, ...)
```

**Arguments**

object	rpms object
node	integer label of the desired end-node.
data	data.frame that includes variables used in rp_equ, e_equ, and design information
variable	string name of variable in data to use as x-axis in plot
...	further arguments passed to plot function.

**Examples**

```

{
# model mean of retirement account value for households with reported
# retirement account values > 0 using a binary tree while accounting for
# clustered data and sample weights.

s1<- which(CE$IRAX > 0)
r1 <-rpms(IRAX~EDUCA+AGE+BLS_URBN, data = CE[s1,], weights=~FINLWT21, clusters=~CID)

# plot node 6 if it is an end-node of the tree
if(6 %in% end_nodes(r1))
  node_plot(object=r1, node=6, data=CE[s1,])

# plot node 8 if it is an end-node of the tree
if(8 %in% end_nodes(r1))
  node_plot(object=r1, node=8, data=CE[s1,])

}

```

---

predict.rpms

*predict.rpms*

---

**Description**

Predicted values based on rpms object

**Usage**

```

## S3 method for class 'rpms'
predict(object, newdata, ...)

```

**Arguments**

object	Object inheriting from rpms
newdata	data frame with variables to use for predicting new values.
...	further arguments passed to or from other methods.

**Value**

vector of predicted values for each row of newdata

### Examples

```
{  
  
# get rpms model of mean Soc Security income for families headed by a  
# retired person by several factors  
r1 <-rpms(SOCRRX~EDUCA+AGE+BLS_URBN+REGION,  
          data=CE[which(CE$INCNONWK==1),], clusters=~CID)  
  
r1  
  
# first 10 predicted means  
predict(r1, CE[10:20, ])  
}
```

---

*predict.rpms\_boost*      *predict.rpms\_boost*

---

### Description

Predicted values based on *rpms\_boost* object

### Usage

```
## S3 method for class 'rpms_boost'  
predict(object, newdata, ...)
```

### Arguments

<i>object</i>	Object inheriting from <i>rpms_boost</i>
<i>newdata</i>	data frame with variables to use for predicting new values.
<i>...</i>	further arguments passed to or from other methods.

### Value

vector of predicted values for each row of *newdata*

---

predict.rpms\_forest     *predict.rpms\_forest*

---

### Description

Gets predicted values given new data based on rpms\_forest model.

### Usage

```
## S3 method for class 'rpms_forest'
predict(object, newdata, ...)
```

### Arguments

object	Object inheriting from rpms_forest
newdata	data frame with variables to use for predicting new values.
...	further arguments passed to or from other methods.

### Value

vector of predicted values for each row of newdata

---

predict.rpms\_proj     *predict.rpms\_proj*

---

### Description

Predicted values based on rpms\_zinf model

### Usage

```
## S3 method for class 'rpms_proj'
predict(object, newdata, ...)
```

### Arguments

object	Object inheriting from rpms_zinf
newdata	data frame with variables to use for predicting new values.
...	further arguments passed to or from other methods.

### Value

vector of predicted values for each row of newdata

---

predict.rpms_zinf	<i>predict.rpms_zinf</i>
-------------------	--------------------------

---

**Description**

Predicted values based on rpms\_zinf model

**Usage**

```
## S3 method for class 'rpms_zinf'  
predict(object, newdata, ...)
```

**Arguments**

object	Object inheriting from rpms_zinf
newdata	data frame with variables to use for predicting new values.
...	further arguments passed to or from other methods.

**Value**

vector of predicted values for each row of newdata

---

print.rpms	<i>print.rpms</i>
------------	-------------------

---

**Description**

print method for class rpms

**Usage**

```
## S3 method for class 'rpms'  
print(x, ...)
```

**Arguments**

x	rpms object
...	further arguments passed to or from other methods.

---

`print.rpms_forest`      *print.rpms\_forest*

---

**Description**

Prints information for a given `rpms_forest` model.

**Usage**

```
## S3 method for class 'rpms_forest'  
print(x, ...)
```

**Arguments**

`x`                      Object inheriting from `rpms_forest`  
`...`                    further arguments passed to or from other methods.

**Value**

vector of predicted values for each row of newdata

---

`print.rpms_zinf`      *print.rpms\_zinf*

---

**Description**

print method for class `rpms_zinf`

**Usage**

```
## S3 method for class 'rpms_zinf'  
print(x, ...)
```

**Arguments**

`x`                      `rpms_zinf` object  
`...`                    further arguments passed to or from other methods.

---

prune_rpms	<i>prune_rpms</i>
------------	-------------------

---

**Description**

prune rpms tree to given node

**Usage**

```
prune_rpms(x, node)
```

**Arguments**

x	rpms object
node	number of node to prune to.

**Value**

subtree ending clipping off any splits after given node.

---

qtree	<i>qtree</i>
-------	--------------

---

**Description**

Code to write a latex qtree plot takes a rpm frame and returns latex code to produce qtree uses linearize as a guide Produces text code to produce tree structure in tex document Requires using LaTeX packages and the following commands in preamble of LaTeX doc: `\usepackage{lscap}` and `\usepackage{tikz-qtree}`

**Usage**

```
qtree(
  t1,
  title = NULL,
  label = NA,
  caption = "",
  digits = 2,
  s_size = TRUE,
  scale = 1,
  lscap = FALSE,
  subnode = 1
)
```

**Arguments**

t1	rpms object created by rpms function
title	string for the top node of the tree
label	string used for labeling the tree figure
caption	string used for caption
digits	integer number of displayed digits
s_size	boolean indicating whether or not to include sample size
scale	numeric factor for scaling size of tree
lscope	boolean to display tree in landscape mode
subnode	starting node of subtree to plot

**Examples**

```
{
# model mean of retirement account value for households with reported
# retirement account values > 0 using a binary tree while accounting for
# clustered data and sample weights.

s1<- which(CE$IRAX > 0)
r1 <-rpms(IRAX~EDUCA+AGE+BLS_URBN, data = CE[s1,], weights=~FINLWT21, clusters=~CID)

# get Latex code
qtrees(r1)

}
```

---

r2stat

*r2*


---

**Description**

Returns the estimated  $R^2$  statistic for determining the fit of the given model to the data

**Usage**

```
r2stat(t1, data, adjusted = TRUE)
```

**Arguments**

t1	Object inheriting from rpms rpms_forest rpms_boost or rpms_zinf
data	data frame with variables used to estimate model
adjusted	TRUE/FALSE whether to compute adjusted $R^2$

**Value**

$R^2$  statistic computed using the model and provided data



---

 rpms

 rpms
 

---

## Description

main function producing a regression tree using variables from `rp_equ` to partition the data and fit the model `e_equ` on each node. Currently only uses data with complete cases of continuous variables.

## Usage

```
rpms(
  rp_equ,
  data,
  weights = ~1,
  strata = ~1,
  clusters = ~1,
  e_equ = ~1,
  e_fn = "survLm",
  l_fn = NULL,
  bin_size = NULL,
  gridpts = 3,
  perm_reps = 1000L,
  pval = 0.05
)
```

## Arguments

<code>rp_equ</code>	formula containing all variables for partitioning
<code>data</code>	data.frame that includes variables used in <code>rp_equ</code> , <code>e_equ</code> , and design information
<code>weights</code>	formula or vector of sample weights for each observation
<code>strata</code>	formula or vector of strata labels
<code>clusters</code>	formula or vector of cluster labels
<code>e_equ</code>	formula for modeling data in each node
<code>e_fn</code>	string name of function to use for modeling (only "survLm" is operational)
<code>l_fn</code>	loss function (ignored)
<code>bin_size</code>	integer specifying minimum number of observations in each node
<code>gridpts</code>	integer number of middle points to do in search; set to n for categorical variables when <code>e_equ</code> is used.
<code>perm_reps</code>	integer specifying the number of thousands of permutation replications to use to estimate p-value
<code>pval</code>	numeric p-value used to reject null hypothesis in permutation test

**Value**

object of class "rpms"

**Examples**

```
{
# model mean of retirement account value for households with reported
# retirement account values > 0 using a binary tree while accounting for
# clustered data and sample weights.

s1<- which(CE$IRAX > 0)
rpms(IRAX~EDUCA+AGE+BLS_URBN, data=CE[s1,], weights=~FINLWT21, clusters=~CID)

# model linear fit between retirement account value and amount of income
# conditioning on education and accounting for clustered data for households
# with reported retirement account values > 0

rpms(IRAX~EDUCA, e_equ=IRAX~FINCBTAX, data=CE[s1,], weights=~FINLWT21, clusters=~CID)
}
```

---

rpms\_boost

*rpms\_boost*

---

**Description**

function for producing boosted rpms models (trees or random forests)

**Usage**

```
rpms_boost(
  rp_equ,
  data,
  weights = ~1,
  strata = ~1,
  clusters = ~1,
  e_equ = ~1,
  bin_size = NULL,
  gridpts = 3,
  perm_reps = 100L,
  pval = 0.05,
  f_size = 200L,
  model_type = "tree",
  times = 2L
)
```

**Arguments**

rp_equ	formula containing all variables for partitioning
data	data.frame that includes variables used in rp_equ, e_equ, and design information
weights	formula or vector of sample weights for each observation
strata	formula or vector of strata labels
clusters	formula or vector of cluster labels
e_equ	formula for modeling data in each node
bin_size	numeric minimum number of observations in each node
gridpts	integer number of middle points to do in search
perm_reps	integer specifying the number of thousands of permutation replications to use to estimate p-value
pval	numeric p-value used to reject null hypothesis in permutation test
f_size	integer specifying the number of trees in the forest (only used if model_type is "forest")
model_type	string: one of "tree" or "forest"
times	integer specifying number of boosting levels to try.

**Value**

object of class "rpms\_boost"

**Examples**

```
{
# model mean of retirement contributions with a binary tree while accounting
# for clustered data and sample weights.

rpms_boost(IRAX~EDUCA+AGE+BLS_URBN, data = CE, weights=~FINLWT21, clusters=~CID, pval=.01)

}
```

---

rpms\_forest

*rpms\_forest*

---

**Description**

produces a random forest using rpms to create the individual trees.

**Usage**

```
rpms_forest(
  rp_equ,
  data,
  weights = ~1,
  strata = ~1,
  clusters = ~1,
  e_fn = "survLm",
  l_fn = NULL,
  bin_size = 5,
  f_size = 500,
  cores = 1
)
```

**Arguments**

rp_equ	formula containing all variables for partitioning
data	data.frame that includes variables used in rp_equ, e_equ, and design information
weights	formula or vector of sample weights for each observation
strata	formula or vector of strata labels
clusters	formula or vector of cluster labels
e_fn	string name of function to use for modeling (only "survLm" is operational)
l_fn	loss function (ignored)
bin_size	numeric minimum number of observations in each node
f_size	integer specifying the number of trees in the forest
cores	integer number of cores to use in parallel if > 1 (doesn't work with Windows operating systems)

**Value**

object of class "rpms"

---

rpms\_proj

*rpms\_proj*

---

**Description**

Returns a survLm\_fit object with coefficients projecting new data onto splits from the given rpms model.

**Usage**

```
rpms_proj(object, newdata, weights = ~1, strata = ~1, clusters = ~1)
```

**Arguments**

object	Object inheriting from rpms
newdata	data frame with variables used to estimate model
weights	formula or vector of sample weights for each observation
strata	formula or vector of strata labels
clusters	formula or vector of cluster labels

**Value**

survLm\_fit object

---

rpms_zinf	<i>rpms_zinf</i>
-----------	------------------

---

**Description**

main function producing a regression tree using variables from rp\_equ to partition the data and fit the model e\_equ on each node. Currently only uses data with complete cases.

**Usage**

```
rpms_zinf(
  rp_equ,
  data,
  weights = ~1,
  strata = ~1,
  clusters = ~1,
  e_equ = ~1,
  e_fn = "survLm",
  l_fn = NULL,
  bin_size = NULL,
  gridpts = 3,
  perm_reps = 1000L,
  pval = 0.05
)
```

**Arguments**

rp_equ	formula containing all variables for partitioning
data	data.frame that includes variables used in rp_equ, e_equ, and design information
weights	formula or vector of sample weights for each observation
strata	formula or vector of strata labels
clusters	formula or vector of cluster labels
e_equ	formula for modeling data in each node

<code>e_fn</code>	string name of function to use for modeling (only "survLm" is operational)
<code>l_fn</code>	loss function (does nothing yet)
<code>bin_size</code>	numeric minimum number of observations in each node
<code>gridpts</code>	integer number of middle points to do in search
<code>perm_reps</code>	integer specifying the number of thousands of permutation replications to use to estimate p-value
<code>pval</code>	numeric p-value used to reject null hypothesis in permutation test

**Value**

object of class "rpms"

# Index

## \* datasets

CE, [4](#)

box\_ind, [3](#)

boxes, [3](#)

CE, [4](#)

end\_nodes, [6](#)

grow\_rpms, [7](#)

in\_node, [8](#)

linearize, [9](#)

node\_plot, [9](#)

predict.rpms, [10](#)

predict.rpms\_boost, [11](#)

predict.rpms\_forest, [12](#)

predict.rpms\_proj, [12](#)

predict.rpms\_zinf, [13](#)

print.rpms, [13](#)

print.rpms\_forest, [14](#)

print.rpms\_zinf, [14](#)

prune\_rpms, [15](#)

qtree, [15](#)

r2stat, [16](#)

rpms, [17](#)

rpms-package, [2](#)

rpms\_boost, [18](#)

rpms\_forest, [19](#)

rpms\_proj, [20](#)

rpms\_zinf, [21](#)