Package 'splinetree'

August 13, 2018

Title Builds and Plots Longitudinal Regression Trees

Version 0.0.0.9000

Description This package builds longitudinal regression trees and longitudinal random forests using a spline projection method.

Depends R (>= 3.5.0), rpart, nlme, splines

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports mosaic,
ggplot2,
treeClust

RoxygenNote 6.0.1.9000

Suggests knitr,
rmarkdown,
testthat

VignetteBuilder knitr

R topics documented:

`av_size` *Average tree size in forest*

Description

Returns the average number of terminal nodes for tree in forest

Usage

```
av_size(forest)
```

Arguments

`forest`	A splineforest object

Value

avergae number of terminal nodes

Examples

```
data(sample_forest)
av_size(sample_forest)
```

flatten_predictors *Flattens predictor variable data into one row per person*

Description

Assumes that splitting explanatory variables do not vary with time. Spline Tree is not meant to handle time-varying covariates.

Usage

```
flatten_predictors(idvar, data)
```

Arguments

idvar	The string name of the ID variable (used to group observations)
data	The full dataset to be flattened (long form)

Value

A wide format dataset with spline coefficients as the responses.

Examples

```
flatten_predictors('ID', nlsySample)
```

forest_projection_R2 *Computes a level-based or shape-based evaluation metric for a splineforest.*

Description

Computes an R-squared-like evaluation metric for a splineforest object. Goal is to see how well the predicted spline coefficients for each individual match the spline coefficients obtained when fitting a spline only to this individual's data (we call these coefficients the true coefficients). Computes 1-SSE/SST, where SSE is the total sum of squared projection errors of the true coefficients compared to the predicted coefficients, and SST is the total sum of squared projection erros of the true coefficients compared to the population mean coefficients. If this is an intercept forest, have the option to compute these sum of squares either with the intercept included or with the intercept ignored to isolate the shape.

Usage

```
forest_projection_R2(forest, method = "oob", removeIntercept = TRUE)
```

Arguments

forest	A splineforest object
method	How would you like to compute this metric? The choices are "oob", "itb", or "all". "oob" means that predictions for a datapoint can only be made using trees for which that datapoint was "out of the bag" (not in the bootstrap sample). "all" means that all trees are used in the prediction for every datapoint. "itb" means that predictions for a datapoint are made using only the trees for which this datapoint was IN the bootstrap sample.
removeIntercept	If true, the projection sum of squared error is computed while ignoring the intercept coefficient. This will help capture the tree's performance at clustering based on shape, not based on level. This parameter is only meaningful if this tree was built using an intercept.

Value

Returns 1-SSE/SST, where SSE is the total sum of squared projection errors of the true coefficients compared to the predicted coefficients, and SST is the total sum of squared projection erros of the true coefficients compared to the population mean coefficients.

Examples

```
forest_projection_R2(sample_forest, method="oob")
forest_projection_R2(sample_forest, method="all")
forest_projection_R2(sample_forest, method="itb")
forest_projection_R2(sample_intercept_forest, method="all")
forest_projection_R2(sample_intercept_forest, method="all", removeIntercept=FALSE)
```

---

forest_Y_R2 *Computes a level-based evaluation metric for a splineforest that was built WITH an intercept.*

Description

Computes the R-squared metric for a splineforest object. Goal is to see how well the predicted response values match the actual response values. Note that this function should only be used on forests where the intercept parameter is TRUE. A simple 1-SSE/SST calculation.

Usage

```
forest_Y_R2(forest, method = "oob")
```

Arguments

forest	A splineforest object
method	How would you like to compute this metric? The choices are "oob", "itb", or "all". "oob" means that predictions for a datapoint can only be made using trees for which that datapoint was "out of the bag" (not in the bootstrap sample). "all" means that all trees are used in the prediction for every datapoint. "itb" means that predictions for a datapoint are made using only the trees for which this datapoint was IN the bootstrap sample.

Value

Returns 1-SSE/SST, where SSE is the total sum of squared errors of the true responses and predicted responses, and SST is the total sum of squared erros of the responses around their mean. If this forest was not built with an intercept, returns NULL.

Examples

```
forest_Y_R2(sample_intercept_forest, method="oob")
forest_Y_R2(sample_intercept_forest, method="all")
forest_Y_R2(sample_intercept_forest, method="itb")
```

---

getBasisMat *Get the basis matrix to be used for this spline tree*

Description

Using the user-specified parameters or the default parameters, computes the basis matrix that will be used for building the tree.

Usage

```
getBasisMat(yvar, tvar, idvar, data, knots = NULL, df, degree, intercept,
  gridPoints, nGrid = 7)
```

Arguments

yvar	Name of response variable (string)
tvar	Name of time variable (string)
idvar	Name of ID variable (string)
data	Full dataset
knots	Knots arguement specified by user. Specifies location of INTERNAL knots.
df	Degrees of freedom arguement specified by user
degree	The degree of the spline polynomial
intercept	Whether or not to use an intercept
gridPoints	Optional. A vector of numbers that will be used as the grid on which to evaluate the projection sum of squares. Should fall roughly within the range of the time variable.
nGrid	Number of grid points to evaluate split function at.

Value

The basis matrix to be used for the tree building process

getNodeData *Returns the portion of the data found at a given terminal node*

Description

Given a terminal node number, this function returns the dataset found at this terminal node. If the dataType arguement is 'all', then all rows of data (with original response values) that fall in this node are returned. Otherwise, the flattened data is returned (one row of data per person/unit, original responses replaced by spline coefficients).

Usage

```
getNodeData(tree, node, dataType = "all")
```

Arguments

tree	a splinetree object
node	The number of the node that you want the data for. Node numbers for your model can be seen using stPrint(tree) or treeSummary(tree). Note that this node number should correspond to a terminal node.
dataType	If "all", the data returned is the original data (one row per individual observation with original response values). If "flat", the data returned is the flattened data (one row per person/unit), with spline coefficients instead of response values.

Value

A dataframe which holds all the data that falls into this node of the tree.

individual_spline *Get spline coefficients for a single person*

Description

Get spline coefficients for a single person

Usage

```
individual_spline(person, idvar, yvar, tvar, data, boundaryKnots, innerKnots,
  degree, intercept)
```

Arguments

person	ID of this person
idvar	name of the id variable (string)
yvar	the name of the response variable
tvar	name of time variable (string)
data	full dataset
boundaryKnots	the boundary knots for the bspline

innerKnots	the inner knots for the bspline
degree	the degree of the bspline
intercept	whether or not to include an intercept

`nlsySample` *Baseline socioeconomic information and BMI of 100 individuals.*

Description

A dataset containing the body mass index (BMI) and baseline socioeconomic information of 500 individuals from the National Longitudinal Survey of Youth 1979 (NLSY), a freely available longitudinal dataset. The 1000 individuals were drawn randomly from among all NLSY respondents with at least 10 non-missing height/weight responses spread out over at least 20 years. This dataset is used in the package vignettes and code examples. Only a small subset of the variables available from the NLSY are included here- see https://www.bls.gov/nls/nlsy79.htm for more

Usage

```
nlsySample
```

Format

A data frame with 16126 rows and 34 columns.

ID Unique identifier for each NLSY respondent

News Did anyone in respondent's family subscribe to a newspaper when respondent was 14?

SEX Respondent's sex. 1 denotes male, 2 denotes female.

AGE Respondent's age

BLACK Indicator for whether or not respondent's identified as Black

BMI Respondent's body mass index - calculated from reported height and weight

Father_14 Was the respondent's father present when respondednt was 14

HEIGHTIN Respondent's height

HGC_FATHER Highest grade completed by respondent's father

HGC_MOTHER Highest grade completed by respondent's mother

HISP Indicator for whether or not respondent's race identified as Hispanic

Lib Did anyone in respondent's family have a library card when respondent was 14?

Mag Did anyone in respondent's family subscribe to a magazine when respondent was 14?

Mother_14 Was respondent's mother present when respondent was 14?

Num_sibs Number of siblings of respondent

Two_Adults_14 Were two adults present in respondent's household at age 14?

WEIGHT Respondent's weight

WHITE Indicator for whether or not respondent identified as white.

HGC Highest grade completed by respondent

POVSTAT Indicator denoting whether or not respondent is in poverty

Dad_Full_Work Did respondent's father work full time (>35 hours) at time of first interview? NA if father not on record.

Mom_Full_Work Did respondent's mother work full time (>35 hours) at time of first interview? NA if mother not on record.

Age_first_weed Age that respondent reported first using marijuana. If they reported never using marijuana, recorded as 100.

Age_first_smoke Age that respondent reported first using tobacco. If they reported never using tobacco, recorded as 100.

Live_with_parents Did respondent report living in same household as the parents on record?

Age_first_alc Age that respondent reported first drinking alcohol. If they reported never drinking alcohol, recorded as 100.

Mom_dad_household Did the respondent's mother and father live in same household at first interview?

Father_on_record Did the respondent have a father listed in the survey record at first interview?

Mother_on_record Did the respondent have a mother listed in the survey record at first interview?

STABLE_RESIDENCE Did the respondent live in the same house from birth to age 14?

URBAN_14 Did the respondent live in an urban area at age 14? 1 denotes town or city, 2 denotes country not farm, 3 denotes farm

RACE Race, as recorded by NLSY. 1 denotes Hispanic, 2 denotes Black, 3 denotes White.

South_birth Indicator for whether or not respondent was born in the south (south as defined by NLSY).

HOUSEHOLD_14 Household identifier for survey respondent

Source

https://www.bls.gov/nls/nlsy79.htm

`plotNodeTraj` *Plot the predicted trajectory for a single node*

Description

Make a simple plot to view the trajectory predicted at a given node. Option to include or not include the individual trajectories of people in the node as well.

Usage

```
plotNodeTraj(tree, node, includeData = FALSE)
```

Arguments

`tree`	a splinetree object
`node`	a node number that must correspond to a terminal node
`includeData`	would you like to see the data from the node plotted along with the predicted trajectory?

plot_varimp *Create a variable importance plot*

Description

Pass in a named vector of variable importance measures. This function will make a barplot of the importances. The importances are scaled, so only relative importance is shown.

Usage

```
plot_varimp(importance_vector)
```

Arguments

importance_vector	a named vector where the names are the variables and the vector stores the importances.

predict_coeffs_RF *Predict spline coefficients using random forest.*

Description

Uses the forest to make predictions of spline coefficients for individuals in the training sample or on a new sample. If the testdata parameter is null, then predictions are given on the training sample according to one of three possible methods. The supplied method parameter must be either "oob", "itb", or "all".

Usage

```
predict_coeffs_RF(forest, method = "oob", testdata = NULL)
```

Arguments

forest	a splinetree forest object
method	a string; either "oob", "itb", or "all". "oob" is the default value. if "oob", predictions for a given data point are made only using trees for which this data point was "out of the bag" (not in the bootstrap sample). If "itb", predictions for a given data point are made using onle the trees for which this datapoint was in the bag (in the bootstrap sample). If "all", all trees are used.
testdata	the test data to make preditions for. If this is provided, then all trees are used for all datapoints.

Value

a matrix of predicted coefficients.

predict_coeffs_tree *Predict spline coefficients for a testset using a single tree.*

Description

Predict spline coefficients for a testset using a single tree.

Usage

```
predict_coeffs_tree(tree, testset = tree$data)
```

Arguments

tree	a splinetree object
testset	the dataset to return predictions for. If omitted, defaults to the data used to build this tree.

Value

a matrix of predicted coefficients

predict_spline_coeffs *Predict spline coefficients from a splinetree object*

Description

Returns a matrix of spline coefficients for each observation in the testset. If no testset is provided, returns predicted coefficients for the individuals in the dataset used to build the tree.

Usage

```
predict_spline_coeffs(tree, testset = tree$parms$flat_data)
```

Arguments

tree	A splinetree object
testset	The dataset to predict coefficients for. Default is the dataset used to make the tree.

Details

importFrom treeClust rpart.predict.leaves

Value

A matrix of spline coefficients. Dimension is number of units in test set by degrees of freedom of the spline.

Examples

```
## Not run:
split_formula <- BMI ~ HISP + WHITE + BLACK + SEX + Dad_Full_Work
  + Mom_Full_Work   + Age_first_weed + Age_first_smoke + Age_first_alc
  + Num_sibs + HGC_FATHER + HGC_MOTHER + Mag + News + Lib + Two_Adults_14
  + Mother_14 + Father_14 + STABLE_RESIDENCE + URBAN_14 + South_Birth
tree <- splineTree(split_formula, BMI~AGE, 'ID', nlsySample, degree=1,
  df=3, intercept=TRUE, cp=0.006, minNodeSize=20)

## End(Not run)
predict_spline_coeffs(tree)
```

---

predict_y *Returns predicted responses.*

Description

Returns predicted responses. Note that this function is most meaningful when used on spline tree objects that have an intercept.

Usage

```
predict_y(model, testData = NULL)
```

Arguments

model	a SplineTree object
testData	The data to predict on. By default, uses the training set.

Value

A vector of predictions with rows corresponding to the testdata.

Examples

```
## Not run:
split_formula <- BMI ~ HISP + WHITE + BLACK + SEX + Dad_Full_Work
  + Mom_Full_Work   + Age_first_weed + Age_first_smoke + Age_first_alc
  + Num_sibs + HGC_FATHER + HGC_MOTHER + Mag + News + Lib + Two_Adults_14
  + Mother_14 + Father_14 + STABLE_RESIDENCE + URBAN_14 + South_Birth
tree <- splineTree(split_formula, BMI~AGE, 'ID', nlsySample, degree=1,
  df=3, intercept=TRUE, cp=0.006, minNodeSize=20)

## End(Not run)
plot(predict_y(tree), tree$parms$data[[tree$parms$yvar]])
```

predict_y_RF *Predict responses using random forest.*

Description

Uses the forest to make predictions of responses for individuals at given times in the training sample or on a new sample. If the testdata parameter is null, then predictions are given on the training sample according to one of three possible methods. The supplied method parameter must be either "oob", "itb", or "all". Note that this method should only be used on trees that have an intercept. Otherwise, the Y predictions will not be accurate at all.

Usage

```
predict_y_RF(forest, method = "oob", testdata = NULL)
```

Arguments

forest	a splinetree forest object
method	a string; either "oob", "itb", or "all". "oob" is the default value. if "oob", predictions for a given data point are made only using trees for which this data point was "out of the bag" (not in the bootstrap sample). If "itb", predictions for a given data point are made using onle the trees for which this datapoint was in the bag (in the bootstrap sample). If "all", all trees are used.
testdata	the test data to make preditions for. If this is provided, then all trees are used for all datapoints.

Value

a matrix of predicted responses.

predict_y_training *Predict responses for the training data*

Description

Returns a vector of predicted responses for the dataset used to build the tree

Usage

```
predict_y_training(model)
```

Arguments

model	a splinetree object

Details

Calling predict_y(model) and predict_y_training(model) return identical results, because when no test data is provided to predict_y(), the default is to use the training set. This is a slightly faster version that can be used when you know that you wish to predict on the training data.

Value

A vector of predicted responses

`predict_Y_tree` *Predict responses for a testset using a single tree.*

Description

Predict responses for a testset using a single tree.

Usage

```
predict_Y_tree(tree, testset = tree$data)
```

Arguments

`tree`	a splinetree object
`testset`	the dataset to return predictions for. Defaults to the dataaset used to build this tree.

Value

a matrix of predicted responses

`prune_forest` *Prunes each tree in forest using a given complexity parameter.*

Description

Prunes each tree in forest using a given complexity parameter.

Usage

```
prune_forest(forest, cp)
```

Arguments

`forest`	A spline forest object
`cp`	The complexity parameter that will be used to prune each tree (see rpart package documentation for detailed description of complexity parameter)

Value

A new splinetree forest object, where each tree has a new size

Examples

```
data(sample_forest)
print(av_size(sample_forest))
print(av_size(prune_forest(sample_forest, cp=0.007)))
print(av_size(prune_forest(sample_forest, cp=0.01)))
```

R2_projected *Computes an R^2-like measure that is based on the projected sum of squared errors.*

Description

Computes an R^2-like measure that is based on the projected sum of squared errors. Can be used on trees whether or not they were built with an intercept. If the tree was built with an intercept, there is the option to ignore the intercept in this projection to isolate how well the tree clusters based on shape.

Usage

```
R2_projected(model, includeIntercept = FALSE)
```

Arguments

model	a splinetree tree object
includeIntercept	If FALSE and if the model was built with an intercept, the projected squared errors are computed while ignoring the intercept. If the model was built without an intercept, this parameter does not do anything.

R2_y *Percent of variation in response explained by spline tree.*

Description

Computes an R^2 measure for the spline tree with respect to prediction. Note that this metric is only meaningful if the spline tree object includes an intercept. If the tree includes an intercept, the measure will be between 0 and 1.

Usage

```
R2_y(model)
```

Arguments

model	a splinetree tree object

Value

An R^2 goodness measure. 1-SSE/SST where SSE is the sum of squared errors between predicted responses and true responses, and SST is sum of squared errors of true responses around population mean. Note that if the tree passed in was built without an intercept, this function will return NULL.

Examples

```
## Not run:
split_formula <- BMI ~ HISP + WHITE + BLACK + SEX + Dad_Full_Work
  + Mom_Full_Work   + Age_first_weed + Age_first_smoke + Age_first_alc
  + Num_sibs + HGC_FATHER + HGC_MOTHER + Mag + News + Lib + Two_Adults_14
  + Mother_14 + Father_14 + STABLE_RESIDENCE + URBAN_14 + South_Birth
 tree <- splineTree(split_formula, BMI~AGE, 'ID', nlsySample, degree=1,
  df=3, intercept=TRUE, cp=0.006, minNodeSize=20)

## End(Not run)
R2_y(tree)
```

---

rpartco *Calculates coordinates for tree plot*

Description

Figures out the coordinates on the tree plot for the little mini trajectory plots. Originally from longRpart.

Usage

```
rpartco(tree, parms = paste(".rpart.parms", dev.cur(), sep = "."))
```

Arguments

tree	a SplineTree object
parms	a string

---

sample_forest *A pre-built random forest (with no intercet) with 20 trees, built to the NLSY sample.*

Description

The purpose of this pre-built forest is to demonstrate the forest evaluation functions without needing to rebuild a forest (slow) every time. This forest does not use an intercept.

Usage

```
sample_forest
```

Format

An object of class `list` of length 16.

sample_intercept_forest

A pre-built random forest (with intercept) with 20 trees, built to the NLSY sample.

Description

The purpose of this pre-built forest is to demonstrate the forest evaluation functions without needing to rebuild a forest (slow) every time. This forest uses an intercept.

Usage

```
sample_intercept_forest
```

Format

An object of class `list` of length 16.

spaghettiPlot

Create a facetted spaghetti plot of a splinetree model

Description

Uses ggplot to create a paneled spaghetti plot of the data, where each panel corresponds to a terminal node in the tree. Allows users to visualize homogeneity of trajectories within the terminal nodes of the tree.

Usage

```
spaghettiPlot(model, colors = NULL)
```

Arguments

`model`	a splinetree object
`colors`	optional arguement specifying colors to be used for each panel.

Examples

```
## Not run:
split_formula <- BMI ~ HISP + WHITE + BLACK + SEX + Dad_Full_Work
  + Mom_Full_Work   + Age_first_weed + Age_first_smoke + Age_first_alc
  + Num_sibs + HGC_FATHER + HGC_MOTHER + Mag + News + Lib + Two_Adults_14
  + Mother_14 + Father_14 + STABLE_RESIDENCE + URBAN_14 + South_Birth
tree <- splineTree(split_formula, BMI~AGE, 'ID', nlsySample, degree=1,
  df=3, intercept=TRUE, cp=0.006, minNodeSize=20)

## End(Not run)
spaghettiPlot(tree)
```

`splineForest` *Build a splinetree random forest*

Description

Builds an ensemble of regression trees for longitudinal or functional data using the spline projection method. The resulting object is a list of splinetree objects along with some additional information. All parameters are used in the same way that they are used in the splineTree function. The additional parameter ntree specifies how many trees should be in the ensemble, and prob controls the probability of selecting a given variable for split consideration at a node.

Usage

```
splineForest(splitFormula, tformula, idvar, data, knots = NULL, df = NULL,
  degree = 3, intercept = FALSE, nGrid = 7, ntree = 50, prob = 0.3,
  cp = 0.001, minNodeSize = 1)
```

Arguments

`splitFormula`	Formula specifying the longitudinal response variable and the time-constant variables that will be used for splitting in the tree.
`tformula`	Formula specifying the longitudinal response variable and the variable that acts as the time variable.
`idvar`	The name of the variable that serves as the ID variable for grouping observations. Must be in quotes
`data`	dataframe that contains all variables specified in the formulas- in long format.
`knots`	Specified locations for internal knots in the spline basis. Defaults to NULL, which corresponds to no internal knots.
`df`	Degrees of freedom of the spline basis. If this is specified but the knots parameter is NULL, then the appropriate number of internal knots will be added at quantiles of the training data. If both df and knots are unspecified, the spline basis will have no internal knots.
`degree`	Specifies degree of spline basis used in the tree.
`intercept`	Specifies whether or not the splitting process will consider the intercept coefficient of the spline projections. Defaults to FALSE, which means that the tree will split based on trajectory shape, ignoring response level.
`nGrid`	Number of grid points to evaluate projection sum of squares at. The default is 7, which corresponds to evaluating projections at the endpoints and quintiles of the time variable.
`ntree`	Number of trees in the forest
`prob`	Probability of selecting a variable to included as a candidate for each split.
`cp`	Complexity parameter passed to the rpart building process.
`minNodeSize`	Minimum number of observational units that can be in a terminal node. Controls tree size and helps avoid overfitting.

Details

The ensemble method is highly similar to the random forest methodology of Brieman (2001). Each tree in the ensemble is fit to a bootstrap sample of the data. At each node of each tree, only a subset of the split variables are considered candidates for the split. In our methodology, the subset of variables considered at each node is determined by a random process. The prob parameter specifies the probability that a given variable will be selected at a certain node. Because the method is based on probability, the same number of variables are not considered for splitting at each node (as in the randomForest package). Note that if prob is small and the number of variables in the splitFormula is also small, there is a high probability that no variables will be considered for splitting at a certain node, which is problematic. The fewer total variables there are, the larger prob should be to ensure good results.

Value

A splineforest object, which stores a list of splinetree objects (in model$Trees), along with information about the spline basis used (model$intercept, model$innerKnots, model$boundaryKnots, etc.), and information about which datapoints were used to build each tree (model$oob_indices and model$index).

Examples

```
splitForm <-BMI~HISP+WHITE+BLACK+HGC_MOTHER+HGC_FATHER+SEX+Num_sibs
forest <- splineForest(splitForm, BMI~AGE, 'ID', nlsySample, degree=1, cp=0.005, ntree=30)
```

---

splineforest_split *Custom rpart split function for spline random forests*

Description

Wrapper for split function required for the random forest functionality. This function is called once per covariate at each potential split. Implements the random selection of variables; each variable is randomly selected to be included or excluded.

Usage

```
splineforest_split(y, wt, x, parms = NULL, continuous)
```

Arguments

y	the responses at this node
wt	the weight of the responses
x	the X data for this covariate
parms	the basis matrix for the spline and the proportion of variables randomly sampled (diceProb)
continuous	value is handled internally by rpart - tells us if this covariate is continuous or categorical (factor).

splineTree *Build a splinetree object*

Description

Builds a regression tree for longitudinal or functional data using the spline projection method. The underlying tree building process uses the rpart package, and the splinetree object is an rpart object with additional stored information. The parmaters df, knots, degree, interept, and nGrid allow for flexibility in the spline basis used for splitting. The parameters minNodeSize and cp allow for flexibility in controlling the size of the final tree.

Usage

```
splineTree(splitFormula, tformula, idvar, data, knots = NULL, df = NULL,
  degree = 3, intercept = FALSE, nGrid = 7, gridPoints = NULL,
  minNodeSize = 10, cp = 0.01)
```

Arguments

splitFormula	Formula specifying the longitudinal response variable and the time-constant variables that will be used for splitting in the tree.
tformula	Formula specifying the longitudinal response variable and the variable that acts as the time variable.
idvar	The name of the variable that serves as the ID variable for grouping observations. Must be in quotes
data	dataframe that contains all variables specified in the formulas- in long format.
knots	Specified locations for internal knots in the spline basis. Defaults to NULL, which corresponds to no internal knots.
df	Degrees of freedom of the spline basis. If this is specified but the knots parameter is NULL, then the appropriate number of internal knots will be added at quantiles of the training data. If both df and knots are unspecified, the spline basis will have no internal knots.
degree	Specifies degree of spline basis used in the tree.
intercept	Specifies whether or not the splitting process will consider the intercept coefficient of the spline projections. Defaults to FALSE, which means that the tree will split based on trajectory shape, ignoring response level.
nGrid	Number of grid points to evaluate projection sum of squares at. If gridPoints is not supploed, this arguement will be used and the projection sum of squares will be evaluated at quantiles of the time variable.
gridPoints	Optional. A vector of numbers that will be used as the grid on which to evaluate the projection sum of squares. Should fall roughly within the range of the time variable.
minNodeSize	Minimum number of observational units that can be in a terminal node. Controls tree size and helps avoid overfitting.
cp	Complexity parameter passed to the rpart building process.

Value

An rpart object with additional splinetree-specific information stored in model$parms.

Examples

```
splitForm <-BMI~HISP+WHITE+BLACK+HGC_MOTHER+HGC_FATHER+SEX+Num_sibs
tree1 <- splineTree(splitForm, BMI~AGE, 'ID', nlsySample, degree=1, intercept=FALSE, cp=0.005)
tree2 <- splineTree(splitForm, BMI~AGE, 'ID', nlsySample, degree=3, intercept=TRUE, cp=0.005)
stPrint(tree1)
treeSummary(tree1)
stPlot(tree1)
stPlot(tree2)
R2_projected(tree1)
R2_projected(tree2)
```

---

splineTreePlot *Tree plot of a spline tree*

Description

Creates a tree plot of a SplineTree object.

Usage

```
splineTreePlot(model, colors = NULL)
```

Arguments

model	a splinetree object
colors	a list of colors that will be used for the terminal nodes (if NULL, will use a rainbow)

---

stPlot *Plots a Spline Tree, showing the tree and the trajectories for comparison.*

Description

Creates a two paneled plot of a SplineTree object that shows both the tree and the trajectories side by side.

Usage

```
stPlot(model, colors = NULL)
```

Arguments

model	a SplineTree object
colors	a list of colors that will be used for the trajectories (if NULL, will automatically select colors from rainbow color scheme.

Examples

```
## Not run:
split_formula <- BMI ~ HISP + WHITE + BLACK + SEX + Dad_Full_Work
  + Mom_Full_Work   + Age_first_weed + Age_first_smoke + Age_first_alc
  + Num_sibs + HGC_FATHER + HGC_MOTHER + Mag + News + Lib + Two_Adults_14
  + Mother_14 + Father_14 + STABLE_RESIDENCE + URBAN_14 + South_Birth
tree <- splineTree(split_formula, BMI~AGE, 'ID', nlsySample, degree=1,
  df=3, intercept=TRUE, cp=0.006, minNodeSize=20)

## End(Not run)
stPlot(tree, colors = c("red", "orange", "green", "blue", "cyan", "magenta"))
```

---

stPrint *Print a spline tree object*

Description

Code adapted only slightly from the rpart base code for print.rpart to support the printing of all coefficients.

Usage

```
stPrint(tree, cp, digits = getOption("digits"))
```

Arguments

tree	The splinetree object
cp	Optional- if provided, a pruned version of the tree will be printed. The tree will be pruned using the provided cp as the complexity parameter.
digits	Specifies how many digits of each coefficient should be printed

Value

A printout of the tree. The printout provides numbered labels for the terminal nodes, a description of the split at each node, the number of observations found at each node, and the predicted spline coefficients for each node.

Examples

```
## Not run:
split_formula <- BMI ~ HISP + WHITE + BLACK + SEX + Dad_Full_Work
  + Mom_Full_Work   + Age_first_weed + Age_first_smoke + Age_first_alc
  + Num_sibs + HGC_FATHER + HGC_MOTHER + Mag + News + Lib + Two_Adults_14
  + Mother_14 + Father_14 + STABLE_RESIDENCE + URBAN_14 + South_Birth
 tree <- splineTree(split_formula, BMI~AGE, 'ID', nlsySample, degree=1,
   df=3, intercept=TRUE, cp=0.006, minNodeSize=20)

## End(Not run)
stPrint(tree)
```

terminalNodeSummary *Prints a summary of a terminal node in a tree*

Description

If no arguement is provided for the parameter node, summaries are printed for every terminal node. Otherwise, the summary of just the requested node is printed.

Usage

```
terminalNodeSummary(tree, node = NULL)
```

Arguments

tree	A splinetree object
node	The number of the node that you want summarized. To see which nodes correspond to which numbers, see stPrint(tree).

tree *Sample tree used in examples*

Description

Sample tree used in examples

Usage

```
tree
```

Format

An object of class rpart of length 14.

treeSize *Returns number of terminal nodes in a tree.*

Description

Returns number of terminal nodes in a tree.

Usage

```
treeSize(model)
```

Arguments

model	A splinetree object, or any rpart object

Value

Number of terminal nodes in tree

Examples

```
## Not run:
split_formula <- BMI ~ HISP + WHITE + BLACK + SEX + Dad_Full_Work
  + Mom_Full_Work   + Age_first_weed + Age_first_smoke + Age_first_alc
  + Num_sibs + HGC_FATHER + HGC_MOTHER + Mag + News + Lib + Two_Adults_14
  + Mother_14 + Father_14 + STABLE_RESIDENCE + URBAN_14 + South_Birth
tree <- splineTree(split_formula, BMI~AGE, 'ID', nlsySample, degree=1,
  df=3, intercept=TRUE, cp=0.006, minNodeSize=20)

## End(Not run)
treeSize(tree)
```

---

treeSummary *Prints the tree frame.*

Description

Prints the tree frame.

Usage

```
treeSummary(model)
```

Arguments

model	A splinetree object.

---

varImp_coeff_RF *Random Forest Variable Importance based on spline coefficients*

Description

Returns the random forest variable importance based on the permutation accruacy measure, which is calculated as the difference in mean squared error between the original data and from randomly permutating the values of a variable.

Usage

```
varImp_coeff_RF(forest, removeIntercept = TRUE, method = "oob")
```

Arguments

forest	a random forest, generated from splineForest()
removeIntercept	a boolean value, TRUE if you want to exclude the intercept in the calculations, FALSE otherwise.
method	the method to be used. This must be one of "oob" (out of bag), "all", "itb" (in the bag).

Value

a matrix of variable importance metrics.

`varImp_Y_RF` *Random Forest Variable Importance based on Y*

Description

Returns the random forest variable importance based on the permutation accruacy measure, which is calculated as the difference in mean squared error between the original data and from randomly permutating the values of a variable.

Usage

```
varImp_Y_RF(forest, method = "oob")
```

Arguments

`forest`	a random forest, generated from splineForest()
`method`	the method to be used. This must be one of "oob" (out of bag), "all", "itb" (in the bag).

Details

The "method" parameter deals with the way in which forest performance should be measured. Since varaible importance is based on a change in performance, the "method" parameter is necessary for a variable importance measure. The choices are "oob" (out of bag), "all", or "itb" (in the bag).

Value

a matrix storing variable importance metrics.

Index