Package 'mvtboost'

December 5, 2016

Type Package

Title Tree Boosting for Multivariate Outcomes

Version 0.5.0

Date 2016-05-25

Author Patrick Miller [aut, cre]

Maintainer Patrick Miller <patrick.mil10@gmail.com>

Description Fits a multivariate model of decision trees for multiple, continuous outcome variables. A model for each outcome variable is fit separately, selecting predictors that explain covariance in the outcomes. Built on top of 'gbm', which fits an ensemble of decision trees to univariate outcomes.

License GPL (>= 2) | file LICENSE

URL https://github.com/patr1ckm/mvtboost

BugReports https://github.com/patr1ckm/mvtboost/issues

Depends R (>= 3.0.0)

Suggests testthat, plyr, MASS, parallel, lars, ggplot2, knitr, rmarkdown

Imports gbm, RColorBrewer, stats, graphics, grDevices, utils,

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-12-05 18:28:47

R topics documented:

`mvtboost-package` *Tree Boosting for Multivariate Outcomes*

Description

Fits a multivariate model of decision trees for multiple, continuous outcome variables. A model for each outcome variable is fit separately, selecting predictors that explain covariance in the outcomes. Built on top of 'gbm', which fits an ensemble of decision trees to univariate outcomes.

Details

The most important function is `mvtb`, which fits the multivariate tree boosting model. See `?mvtb` for details. The fitted model objects have `summary`, `print`, `plot` and `predict` methods. Additionally, `mvtb.ri` to computes the relative influence of each predictor, and `mvtb.covex` computes an estimate of the covariance explained in pairs of outcomes by predictors. These tables can be displayed as heatmaps using `mvtb.heat`. Examples for fitting, tuning and interpreting the models are available in the help pages and two package vignettes:

vignette("mvtboost_vignette") vignette("mvtboost_wellbeing")

Author(s)

Patrick Miller [aut, cre]

Maintainer: Patrick Miller <patrick.mil10@gmail.com>

References

Miller P.J., Lubke G.H, McArtor D.B., Bergeman C.S. (2015) Finding structure in data: A data mining alternative to multivariate multiple regression. Psychological Methods.

See Also

`gbm`

Examples

```
data(wellbeing)
Y <- wellbeing[,21:26]
X <- wellbeing[,1:20]
Ys <- scale(Y)
cont.id <- unlist(lapply(X,is.numeric))
Xs <- scale(X[,cont.id])

res <- mvtb(Y=Ys,X=Xs)

summary(res)
plot(res,predictor.no = 8)
predict(res,newdata=Xs)

covex <- mvtb.covex(res, Y=Ys, X=Xs)
mvtb.cluster(covex)
par(mar=c(4,7,1,1))
mvtb.heat(covex,cexRow=.8)
par(mar=c(5,5,1,1))
mvtb.heat(t(mvtb.ri(res)),cexRow=.8,cexCol=1,dec=0)
```

---

addalpha *Add alpha*

Description

Internal to colorRampPaletteALpha but exported in case a user wants to modify

Usage

```
addalpha(colors, alpha = 1)
```

Arguments

colors	list of colors
alpha	value of alpha

---

colorRampPaletteAlpha *Add alpha and ramps between colors*

Description

Internal to mvtb.heat, but exported for easy modification

Usage

```
colorRampPaletteAlpha(colors, n = 32, interpolate = "linear")
```

Arguments

colors	original colors
n	number of colors
interpolate	linear, otherwise spline interpolation is used

---

mvtb *Fitting a Multivariate Tree Boosting Model*

Description

Builds on gbm (Ridgeway 2013; Friedman, 2001) to fit a univariate tree model for each outcome, selecting predictors at each iteration that explain (co)variance in the outcomes. The number of trees included in the model can be chosen by minimizing the multivariate mean squared error using cross validation or a test set.

Usage

```
mvtb(Y, X,
     n.trees = 100,
     shrinkage = 0.01,
     interaction.depth = 1,
     distribution="gaussian",
     train.fraction = 1,
     bag.fraction = 1,
     cv.folds = 1,
     s = NULL,
     seednum = NULL,
     compress = FALSE,
     save.cv = FALSE,
     iter.details = TRUE,
     verbose=FALSE,
     mc.cores = 1, ...)

mvtb.fit(Y,X,
         n.trees=100,
         shrinkage=.01,
         interaction.depth=1,
         bag.fraction=1,
         s=1:nrow(X),
         seednum=NULL,...)
```

Arguments

Y	vector, matrix, or data.frame for outcome variables with no missing values. To easily compare influences across outcomes and for numerical stability, outcome variables should be scaled to have unit variance.

X	vector, matrix, or data.frame of predictors. For best performance, continuous predictors should be scaled to have unit variance. Categorical variables should converted to factors.
n.trees	maximum number of trees to be included in the model. Each individual tree is grown until a minimum number observations in each node is reached.
shrinkage	a constant multiplier for the predictions from each tree to ensure a slow learning rate. Default is .01. Small shrinkage values may require a large number of trees to provide adequate fit.
interaction.depth	fixed depth of trees to be included in the model. A tree depth of 1 corresponds to fitting stumps (main effects only), higher tree depths capture higher order interactions (e.g. 2 implies a model with up to 2-way interactions)
distribution	Character vector specifying the distribution of all outcomes. Default is "gaussian" see ?gbm for further details.
train.fraction	proportion of the sample used for training the multivariate additive model. If both cv.folds and train.fraction are specified, the CV is carried out within the training set.
bag.fraction	proportion of the training sample used to fit univariate trees for each response at each iteration. Default: 1
cv.folds	number of cross validation folds. Default: 1. Runs k + 1 models, where the k models are run in parallel and the final model is run on the entire sample. If larger than 1, the number of trees that minimize the multivariate MSE averaged over k-folds is reported in object$best.trees
s	vector of indices denoting observations to be used for the training sample. If s is given, train.fraction is ignored.
seednum	integer passed to set.seed
compress	TRUE/FALSE. Compress output results list using bzip2 (approx 10% of original size). Default is FALSE.
save.cv	TRUE/FALSE. Save all k-fold cross-validation models. Default is FALSE.
iter.details	TRUE/FALSE. Return training, test, and cross-validation error at each iteration. Default is FALSE.
verbose	If TRUE, will print out progress and performance indicators for each model. Default is FALSE.
mc.cores	Number of cores for cross validation.
...	additional arguments passed to gbm. These include distribution, weights, var.monotone, n.minobsinnode, keep.data, verbose, class.stratify.cv. Note that other distribution arguments have not been tested.

Details

This function selects predictors that explain covariance in multivariate outcomes. This is done efficiently by fitting separate gbm models for each outcome (contained in $models).

(Relative) influences can be retrieved using summary or mvtb.ri, which are the usual reductions in SSE due to splitting on each predictor. The covariance explained in pairs of outcomes by each

predictor can be computed using mvtb.covex. Partial dependence plots can be obtained from mvtb.plot.

The model is tuned jointly by selecting the number of trees that minimize multivariate mean squared error in a test set (by setting train.fraction) or averaged over k folds in k-fold cross-validation (by setting cv.folds > 1). The best number of trees is available via $best.trees. If both cv.folds and train.fraction is specified, cross-validation is carried out within the training set. If s is specified, train.fraction is ignored but cross-validation will be carried out for observations in s.

Cross-validation models are usually discarded but can be saved by setting save.cv = TRUE. CV models can be accessed from $ocv of the output object. Observations can be specifically set for inclusion in the training set by passing a vector of integers indexing the rows to include to s. Multivariate mean squared training, test, and cv error are available from $trainerr, $testerr, $cverr from the output object when iter.details = TRUE.

Since the output objects can be large, automatic compression is available by setting compress=TRUE. All methods that use the mvtb object automatically uncompress this object if necessary. The function mvtb.uncomp is available to manually decompress the object.

Note that trees are grown until a minimum number of observations in each node is reached. If the number of training samples*bag.fraction is less the minimum number of observations, (which can occur with small data sets), this will cause an error. Adjust the n.minobsinnode, train.fraction, or bag.fraction.

Cross-validation can be parallelized by setting mc.cores > 1. Parallel cross-validation is carried out using parallel::mclapply, which makes mc.cores copies of the original environment. For models with many trees (> 100K), memory limits can be reached rapidly. mc.cores will not work on Windows.

Value

Fitted model. This is a list containing the following elements:

- models - list of gbm models for each outcome. Functions from the gbm package (e.g. to compute relative influence, print trees, obtain predictions, etc) can be directly applied to each of these models
- best.trees - A list containing the number of trees that minimize the multivariate MSE in a test set or by CV, and n.trees. Many of the functions in the package default to using the minimum value of the three.
- params - arguments to mvtb
- trainerr - multivariate training error at each tree (If iter.details = TRUE)
- testerr - multivariate test error at each tree (if train.fraction < 1 and iter.details = TRUE)
- cverr - multivariate cv error at each tree (if cv.folds > 1 and iter.details = TRUE)
- ocv - the CV models if save.cv=TRUE
- s - indices of training sample
- n - number of observations
- xnames
- ynames

Functions

- `mvtb.fit`:

References

Miller P.J., Lubke G.H, McArtor D.B., Bergeman C.S. (Accepted) Finding structure in data with multivariate tree boosting.

Ridgeway, G., Southworth, M. H., & RUnit, S. (2013). Package 'gbm'. Viitattu, 10, 2013.

Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. Journal of Animal Ecology, 77(4), 802-813.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.

See Also

`summary.mvtb`, `predict.mvtb`

`mvtb.covex` to estimate the covariance explained in pairs of outcomes by predictors

`mvtb.nonlin` to help detect nonlinear effects or interactions

`plot.mvtb`, `mvtb.perspec` for partial dependence plots

`mvtb.uncomp` to uncompress a compressed output object

Examples

```
data(wellbeing)
Y <- wellbeing[,21:26]
X <- wellbeing[,1:20]
Ys <- scale(Y)
cont.id <- unlist(lapply(X,is.numeric))
Xs <- scale(X[,cont.id])

## Fit the model
res <- mvtb(Y=Ys,X=Xs)

## Interpret the model
summary(res)
covex <- mvtb.covex(res, Y=Ys, X=Xs)
plot(res,predictor.no = 8)
predict(res,newdata=Xs)
mvtb.cluster(covex)
mvtb.heat(t(mvtb.ri(res)),cexRow=.8,cexCol=1,dec=0)
```

`mvtb.cluster` *Clustering the covariance explained or relative influence matrix*

Description

The 'covariance explained' by each predictor is the reduction in covariance between each pair of outcomes due to splitting on each predictor over all trees (`$covex`). To aid in the interpretability of the covariance explained matrix, this function clusters the rows (pairs of outcomes) and the columns (predictors) of `object$covex` so that groups of predictors that explain similar pairs of covariances are closer together. This function can also be used to cluster the relative influence matrix. In this case, the rows (usually outcomes) and columns (usually predictors) with similar values will be clustered together.

Usage

```
mvtb.cluster(x, clust.method = "complete", dist.method = "euclidean",
  plot = FALSE, ...)
```

Arguments

`x`	Any matrix, such as `mvtb.covex(object)`, or `mvtb.ri(object)`.
`clust.method`	clustering method for rows and columns. This should be (an unambiguous abbreviation of) one of `"ward.D"`, `"ward.D2"`, `"single"`, `"complete"`, `"average"` (= UPGMA), `"mcquitty"` (= WPGMA), `"median"` (= WPGMC) or `"centroid"` (= UPGMC).
`dist.method`	method for computing the distance between two lower triangular covariance matrices. This must be one of `"euclidean"`, `"maximum"`, `"manhattan"`, `"canberra"`, `"binary"` or `"minkowski"`. Any unambiguous substring can be given.
`plot`	Produces a heatmap of the covariance explained matrix. see `?mvtb.heat`
`...`	Arguments passed to `mvtb.heat`

Details

The covariance explained by each predictor is only unambiguous if the predictors are uncorrelated and interaction.depth = 1. If predictors are not independent, the decomposition of covariance explained is only approximate (like the decomposition of R^2 by each predictor in a linear model). If interaction.depth > 1, the following heuristic is used: the covariance explained by the tree is assigned to the predictor with the largest influence in each tree.

Note that different distances measures (e.g. `"manhattan"`, `"euclidean"`) provide different ways to measure (dis)similarities between the covariance explained patterns for each predictor. See `?dist` for further details. After the distances have been computed, `hclust` is used to form clusters. Different clustering methods (e.g. `"ward.D"`, `"complete"`) generally group rows and columns differently (see `?hclust` for further details). It is suggested to try different distance measures and clustering methods to obtain the most interpretable solution. The defaults are for `"euclidean"` distances and `"complete"` clustering. Transposing the rows and columns may also lead to different results.

A simple heatmap of the clustered matrix can be obtained by setting `plot=TRUE`. Details of the plotting procedure are available via `mvtb.heat`.

`covex` values smaller than `getOption("digits")` are truncated to 0. Note that it is possible to obtain negative variance explained due to sampling fluctuation. These can be truncated or ignored.

Value

clustered covariance matrix, with re-ordered rows and columns.

See Also

`mvtb.heat`

`mvtb.covex` *Estimate the covariance explained matrix*

Description

Estimate the covariance explained matrix

Usage

```
mvtb.covex(object, Y, X, n.trees = NULL, iter.details = FALSE)
```

Arguments

`object`	an object of class `mvtb`
`Y`	vector, matrix, or data.frame for outcome variables with no missing values. To easily compare influences across outcomes and for numerical stability, outcome variables should be scaled to have unit variance.
`X`	vector, matrix, or data.frame of predictors. For best performance, continuous predictors should be scaled to have unit variance. Categorical variables should converted to factors.
`n.trees`	number of trees to use. Defaults to the minimum number of trees by CV, test, or training error
`iter.details`	`TRUE/FALSE`. Return the loss, relative loss, and selected predictors at each iteration as a list

Value

Covariance explained matrix, or a list if `iter.details` is `TRUE`.

mvtb.heat *Clustered heatmap of tables from* mvtb

Description

Simple (clustered) heatmap of tables from mvtb (relative influence, covariance explained)

Usage

```
mvtb.heat(x, clust.method = "ward.D", dist.method = "manhattan", dec = 2,
  numformat = NULL, col = NULL, cexRow = NULL, cexCol = NULL, ...)
```

Arguments

x	Any table. For example: the covariance explained from mvtb.covex, or relative influence mvtb.ri(res).
clust.method	clustering method for rows and columns. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC). If NULL, unclustered.
dist.method	method for computing the distance between two lower triangular covariance matrices. This must be one of "euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski". Any unambiguous substring can be given.
dec	Number of decimal places to round to if numformat is unspecified. Defaults to 2.
numformat	function to format the covex values into strings. Defaults to removing leading 0 and rounding to dec = 2 decimal places.
col	A list of colors mapping onto covex explained values. A white to black gradient is default.
cexRow,	See cex.axis from par. The magnification used for the row axis labels. A useful default is provided.
cexCol,	See cex.axis from par. The magnification used for the col axis labels. The default is set equal to the row axis labels.
...	extra arguments are passed to image, then to plot. See ?image, ?par

Details

The row and column names of x are used for the labels. See the examples for modifying the default colors.

Value

heatmap of x, usually a covariance explained matrix or a matrix of (relative) influences.

See Also

plot.mvtb, mvtb.perspec

Examples

```
data(wellbeing)
Y <- wellbeing[,21:26]
X <- wellbeing[,1:20]
Ys <- scale(Y)
cont.id <- unlist(lapply(X,is.numeric))
Xs <- scale(X[,cont.id])

res <- mvtb(Y=Ys,X=Xs)

covex <- mvtb.covex(res, Y=Ys, X=Xs)
par(mar=c(4,7,1,1))
mvtb.heat(covex,cexRow=.8)

col <- colorRampPaletteAlpha(RColorBrewer::brewer.pal(9,"Greys"),100)
mvtb.heat(covex, Y=Ys, X=Xs, col=col, cexRow=.8)

par(mar=c(5,5,1,1))
mvtb.heat(t(mvtb.ri(res)),cexRow=.8,cexCol=1,dec=0)
```

---

mvtb.nonlin *Detect departures from linearity from a multivariate tree boosting model.*

Description

Detect departures from linearity from a multivariate tree boosting model.

Usage

```
mvtb.nonlin(object, Y, X, n.trees = NULL, detect = "grid", scale = TRUE)
```

Arguments

object	object of class mvtb
Y	matrix of predictors
X	matrix of responses
n.trees	number of trees. Defaults to the minimum number of trees given that minimize CV, test, training error.
detect	method for testing possible non-linear effects or interactions. Possible values are "grid", "influence", and "lm". See details.
scale	For method "influence", whether the resulting influences are scaled to sum to 100.

Details

This function provides a statistic to detect departures from linearity in the multivariate boosting model for any outcome as a function of pairs of predictors. These departures could be interactions between pairs of variables, or more general non-linear effects. Please note that these methods should be interpreted as exploratory only.

Several methods are provided for detecting departures from non-linearity from pairs of predictors. The `"grid"` method computes a grid of the model implied predictions as a function of two predictors, averaging over the others. A linear model predicting the observed outcomes from the predicted values is fit, and the mean squared residuals (times 1000) are reported. Large residuals indicate deviations from linearity.

The `"influence"` method computes the reductions in SSE attributable to predictors after the first split on the tree. These reductions in sums of squared error (or influences) indicate to what extent individual predictors capture deviations from linear, main effects.

The `"lm"` method is the same as the `"grid"` method, but produces the grid of predicted values by conditioning on the average values of the other predictors rather than averaging over the values of the other predictors (see Elith et al., 2008) . Like the `"grid"` approach, large residuals from a linear model (times 1000) indicate departures from linearity.

A final option is to use `gbm::interact.gbm` from the `gbm` package to detect interactions. It can be used directly on individual `mvtb` output models from `object$models`.

These methods are not necessarily overlapping, and can produce different results. We suggest using several approaches, followed by plotting the model implied effects of the two predictors.

Value

For each outcome, a list is produced showing the interactions in two forms. The first is `$rank.list`, which shows the nonlinear effect for each pair of predictors ranked according to the size of the departure from non-linearity. The second, `$interactions`, shows the departure from non-linearity for all pairs of predictors.

References

Miller P.J., Lubke G.H, McArtor D.B., Bergeman C.S. (Submitted) Finding structure in data: A data mining alternative to multivariate multiple regression. Psychological Methods.

Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. Journal of Animal Ecology, 77(4), 802-813.

Friedman, J. H., & Meulman, J. J. (2003). Multiple additive regression trees with application in epidemiology. Statistics in medicine, 22(9), 1365-1381.

See Also

`interact.gbm`, `mvtb.perspec`, `plot.gbm`

mvtb.perspec *Perspective plot for 2 predictors and 1 response.*

Description

This is a plot of the model implied function of 2 predictors averaged over the other predictors included in the model. This is called a partial dependence plot. As an alternative to the perspective (3D) plot, a 2D heat plot can be obtained directly using ?plot.gbm.

Usage

```
mvtb.perspec(object, response.no = 1, predictor.no = 1:2, n.trees = NULL,
  phi = 15, theta = -55, r = sqrt(10), d = 3, xlab = NULL,
  ylab = NULL, zlab = NULL, ticktype = "detailed", ...)
```

Arguments

object	mvtb output object
response.no	index of the response variable
predictor.no	vector containing indices of the predictor variables to plot
n.trees	desired number of trees. Defaults to the minimum number of trees by CV, test, or training error
phi	angle of viewing direction. See ?persp.
theta	angle of viewing direction See ?persp.
r	distance from eye to center. See ?persp.
d	strength of perspective. See ?persp.
xlab,	title for x axis, must be character strings.
ylab,	title for y axis, must be character strings.
zlab,	title for z axis, must be character strings.
ticktype	'detailed' gives axis points. See ?persp for other options.
...	extra arguments are passed to persp. See ?persp

Value

Function returns a plot.

See Also

plot.gbm, plot.mvtb, mvtb.heat

mvtb.ri *Computes the relative influence of each predictor for each outcome*

Description

The relative influence of a predictor is the reduction in sums of squares attributable to splits on individual predictors. It is often expressed as a percent (sums to 100).

Usage

```
mvtb.ri(object, n.trees = NULL, relative = "col", ...)
```

Arguments

object	mvtb output object
n.trees	number of trees to use. Defaults to the minimum number of trees by CV, test, or training error
relative	How to scale the multivariate influences. If "col", each column sums to 100. If "tot", the whole matrix sums to 100 (a percent). Otherwise, the raw reductions in SSE are returned.
...	Additional arguments passed to gbm::relative.influence

Value

Matrix of (relative) influences.

mvtb.uncomp *Uncompress a compressed mvtb output object*

Description

This function uncompresses a compressed mvtb output object. All elements are uncompressed.

Usage

```
mvtb.uncomp(object)
```

Arguments

object	an object of class mvtb

plot.mvtb *Plots the model implied effect of 1 predictor for one outcome*

Description

Plots the model implied effect of 1 predictor for one outcome

Usage

```
## S3 method for class 'mvtb'
plot(x, predictor.no = 1, response.no = 1, n.trees = NULL,
  X = NULL, xlab = NULL, ylab = NULL, return.grid = FALSE, ...)
```

Arguments

x	mvtb output object
predictor.no	index of the predictor variable
response.no	index of the response variable
n.trees	desired number of trees. Defaults to the minimum number of trees by CV, test, or training error
X	optional vector, matrix, or data.frame of predictors. If included, a 'rug' (a small vertical line for each observation) is plotted on the x-axis showing the density of predictor.no.
xlab	label of the x axis
ylab	label of the y axis
return.grid	TRUE/FALSE return the prediction grid from gbm. Default is FALSE.
...	extra arguments are passed to plot. See ?par

Details

This is the classic partial dependence plot, where the model implied effect of the chosen predictor is plotted controlling for the other predictors. In addition to the model-implied effect, the relative influence of the predictor is included in the x-axis label. If this is not desired, a custom label can be provided via xlab.

Value

Produces a plot of the model implied effect along with the relative influence of the predictor. If return.grid=TRUE, returns the plotting matrix as well.

See Also

plot.gbm, mvtb.perspec, for other plots, mvtb.heat to plot the covariance explained by predictors in a heatmap

predict.mvtb *Predicted values*

Description

Predicted values

Usage

```
## S3 method for class 'mvtb'
predict(object, n.trees = NULL, newdata, drop = TRUE, ...)
```

Arguments

object	mvtb object
n.trees	number of trees. If a vector, returns predictions in an array. Defaults to the minimum number of trees by CV, test, or training error
newdata	matrix of predictors.
drop	TRUE/FALSE Drop any dimensions of length 1
...	not used

Value

Returns an (array, matrix, vector) of predictions for all outcomes. The third dimension corresponds to the predictions at a given number of trees, the second dimension corresponds to the number of outcomes.

print.mvtb *Simple default printing of the mvtb output object*

Description

Simple default printing of the mvtb output object

Usage

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

Arguments

x	mvtb output object
...	unused

summary.mvtb *Computes a summary of the multivariate tree boosting model*

Description

Computes a summary of the multivariate tree boosting model

Usage

```
## S3 method for class 'mvtb'
summary(object, print = TRUE, n.trees = NULL,
  relative = "col", ...)
```

Arguments

object	mvtb output object
print	result (default is TRUE)
n.trees	number of trees used to compute relative influence. Defaults to the minimum number of trees by CV, test, or training error
relative	relative If 'col', each column sums to 100. If 'tot', the whole matrix sums to 100 (a percent). If 'n', the raw reductions in SSE are returned.
...	additional arguments affecting the summary produced.

Value

Returns the best number of trees, the univariate relative influence of each predictor for each outcome, and covariance explained in pairs of outcomes by each predictor

See Also

mvtb.ri, gbm.ri, mvtb.cluster

wellbeing *Psychological well-being*

Description

This data set contains 984 survey responses on the psychological well being (Ryff & Keyes, 1995) of middle-aged and older adults from the Notre Dame Study of Health and Well Being (Bergeman & Deboek, 2014). Psychological well-being has 6 sub-scales - autonomy, environmental mastery (envmast), personal growth (prsgrwth), positive relationships with others (posrel), purpose in life (prpsnlf), and self acceptance. These sub-scales are columns 21:26.

The following predictors of psychological well-being are included:

- gender
- age (18-91)
- education (educ, with levels high-school, vocational, some-college, graduate degree, other)
- income (inc, with binned levels)
- chronic health problems (chrhlth)
- somatic health problems (somhlth)
- self-reported health problems (slfhlth)
- positive affect (posaff)
- negative affect (negaff)
- positive social control (pscntrl)
- control of internal states (cistot)
- commitment (comit)
- control (cntrl)
- challenge (chlng)
- ego resilience (egores)
- positive social support from friends/family (pssfrnd, pssfam)
- perceived stressful emotions (pssemot)
- perceived stressful problems (pssprob)
- loneliness (lonlnes)

* Note: This data set is only included for demonstration of the software. No other use of the data is permitted. Additional noise has been added to this data set to protect privacy.

Usage

```
wellbeing
```

Format

A data frame of 984 observations and 26 variables (6 outcomes and 20 predictors)

Source

Bergeman, C. S., & Deboeck, P. R. (2014). Trait stress resistance and dynamic stress dissipation on health and well-being: The reservoir model. Research in Human Development, 11(2), 108-125.

References

Ryff, C. D., & Keyes, C. L. M. (1995). The structure of psychological well-being revisited. Journal of Personality and Social Psychology, 69(4), 719.

Index