

# Package: featurefinder (via r-universe)

October 16, 2024

**Title** Feature Finder

**Version** 1.1

**Description** Finds modelling features through a detailed analysis of model residuals using 'rpart' classification and regression trees. Scans the residuals of a model across subsets of the data to identify areas where the model prediction differs from the actual target variable. S. Chatterjee, A. S. Hadi (2006) <[doi:10.1002/0470055464](https://doi.org/10.1002/0470055464)>.

**Depends** R (>= 3.2.0)

**License** MIT + file LICENSE

**LazyData** true

**RoxygenNote** 6.0.1

**Suggests** png, knitr, Metrics, mlr, gbm, randomForest

**VignetteBuilder** knitr

**Imports** rpart, rpart.plot, utils, plyr, grDevices

**NeedsCompilation** no

**Author** Richard Davis [aut, cre]

**Maintainer** Richard Davis <[davisconsulting@gmail.com](mailto:davisconsulting@gmail.com)>

**Date/Publication** 2018-12-03 05:20:03 UTC

**Repository** <https://richard987.r-universe.dev>

**RemoteUrl** <https://github.com/cran/featurefinder>

**RemoteRef** HEAD

**RemoteSha** 808414cdabb6417c8c3e3ad309aa170002286651

## Contents

|                       |   |
|-----------------------|---|
| addFeatures . . . . . | 2 |
| dat . . . . .         | 4 |
| dat0 . . . . .        | 4 |
| data . . . . .        | 5 |

|   |           |
|---|-----------|
| doAllFactors . . . . .                          | 5         |
| expr . . . . .                                  | 6         |
| fileConn . . . . .                              | 6         |
| filename . . . . .                              | 7         |
| findFeatures . . . . .                          | 7         |
| futuresdata . . . . .                           | 9         |
| generateResidualCutoffCode . . . . .            | 10        |
| generateTrees . . . . .                         | 11        |
| getVarAv . . . . .                              | 12        |
| i . . . . .                                     | 12        |
| mainfaclevels . . . . .                         | 13        |
| maxFactorLevels . . . . .                       | 13        |
| mpgdata . . . . .                               | 14        |
| names . . . . .                                 | 14        |
| parseSplits . . . . .                           | 15        |
| pathterms . . . . .                             | 15        |
| printResiduals . . . . .                        | 16        |
| runname . . . . .                               | 17        |
| saveTree . . . . .                              | 17        |
| splitlist . . . . .                             | 18        |
| t . . . . .                                     | 19        |
| tree . . . . .                                  | 19        |
| treeGenerationMinBucket . . . . .               | 20        |
| trees . . . . .                                 | 20        |
| treesAll . . . . .                              | 21        |
| treeSummaryMinBucket . . . . .                  | 21        |
| treeSummaryResidualMagnitudeThreshold . . . . . | 22        |
| treeSummaryResidualThreshold . . . . .          | 22        |
| vars . . . . .                                  | 23        |
| <b>Index</b>                                    | <b>24</b> |

---

|             |                    |
|-------------|--------------------|
| addFeatures | <i>addFeatures</i> |
|-------------|--------------------|

---

## Description

Use the results of findFeatures to append promising features to a dataframe for further testing

## Usage

```
addFeatures(df, path, prefix)
```

## Arguments

|        |                                       |
|--------|---------------------------------------|
| df     | A dataframe                           |
| path   | A string                              |
| prefix | A list of trees generated by saveTree |

**Value**

A dataframe with extra features appended

**Examples**

```

require(featurefinder)
data(futuresdata)
data=futuresdata
data$SMIfactor=paste("smi",as.matrix(data$SMIfactor),sep="")
n=length(data$DAX)
nn=floor(length(data$DAX)/2)

# Can we predict the relative movement of DAX and SMI?
data$y=data$DAX*0 # initialise the target to 0
data$y[1:(n-1)]=((data$DAX[2:n])-(data$DAX[1:(n-1)]))/
  (data$DAX[1:(n-1)]-(data$SMI[2:n]-(data$SMI[1:(n-1)])))/(data$SMI[1:(n-1)])

# Fit a simple model
thismodel=lm(formula=y ~ .,data=data)
expected=predict(thismodel,data)
actual=data$y
residual=actual-expected
data=cbind(data,expected, actual, residual)

CSVPath=tempdir()
fcsv=paste(CSVPath,"/futuresdata.csv",sep="")
write.csv(data[(nn+1):(length(data$y))],,file=fcsv,row.names=FALSE)
exclusionVars=""residual\","expected\","actual\","y""
factorToNumericList=c()

# Now the dataset is prepared, try to find new features
tempDir=findFeatures(outputPath="NoPath", fcsv, exclusionVars,
factorToNumericList,
treeGenerationMinBucket=50,
treeSummaryMinBucket=20,
useSubDir=FALSE)

newfeat1=((data$SMIfactor==0) & (data$CAC < 2253) & (data$CAC< 1998) & (data$CAC>=1882)) * 1.0
newfeat2=((data$SMIfactor==1) & (data$SMI < 7837) & (data$SMI >= 7499)) * 1.0
newfeatures=cbind(newfeat1, newfeat2) # create columns for the newly found features
datanew=cbind(data,newfeatures)
thismodel=lm(formula=y ~ .,data=datanew)
expectednew=predict(thismodel,datanew)

requireNamespace("Metrics")
OriginalRMSE = Metrics::rmse(data$y,expected)
NewRMSE = Metrics::rmse(data$y,expectednew)

print(paste("OriginalRMSE = ",OriginalRMSE))
print(paste("NewRMSE = ",NewRMSE))

# Append new features to a dataframe automatically

```

```
dataWithNewFeatures = addFeatures(df=data, path=tempDir, prefix="auto_")  
head(df)
```

---

|     |            |
|-----|------------|
| dat | <i>dat</i> |
|-----|------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

A data frame with 234 rows and 11 variables

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(dat)  
head(dat)
```

---

|      |             |
|------|-------------|
| dat0 | <i>dat0</i> |
|------|-------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

A data frame with 234 rows and 11 variables

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(dat0)  
head(dat0)
```

---

|      |             |
|------|-------------|
| data | <i>data</i> |
|------|-------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

A data frame with 234 rows and 11 variables

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(data)
head(data)
```

---

|              |                     |
|--------------|---------------------|
| doAllFactors | <i>doAllFactors</i> |
|--------------|---------------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

A boolean to indicate whether to scan over all categorical factor partitions.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(doAllFactors)
head(doAllFactors)
```

---

|      |             |
|------|-------------|
| expr | <i>expr</i> |
|------|-------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

A string describing the formula defining a leaf node.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(expr)
head(expr)
```

---

|          |                 |
|----------|-----------------|
| fileConn | <i>fileConn</i> |
|----------|-----------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

A text output object.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(fileConn)
head(fileConn)
```

---

|          |                 |
|----------|-----------------|
| filename | <i>filename</i> |
|----------|-----------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

A filename for output.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(filename)
head(filename)
```

---

|              |                     |
|--------------|---------------------|
| findFeatures | <i>findFeatures</i> |
|--------------|---------------------|

---

**Description**

Perform analysis of residuals grouped by factor to identify features which explain the target variable

**Usage**

```
findFeatures(outputPath = "NoPath", fcsv, exclusionVars, factorToNumericList,
  treeGenerationMinBucket = 20, treeSummaryMinBucket = 50,
  treeSummaryResidualThreshold = 0,
  treeSummaryResidualMagnitudeThreshold = 0, doAllFactors = TRUE,
  maxFactorLevels = 20, useSubDir = TRUE, tempDirFolderName = "")
```

**Arguments**

|                                       |   |
|---------------------------------------|---|
| outputPath                            | A string containing the location of the input csv file. Results are also stored in this location. Set to "NoPath" to use tempdir() or leave blank |
| fcsv                                  | A string containing the name of a csv file  |
| exclusionVars                         | A string consisting of a list of variable names with double quotes around each variable   |
| factorToNumericList                   | A list of variable names as strings   |
| treeGenerationMinBucket               | Desired minimum number of data points per leaf (default 20)   |
| treeSummaryMinBucket                  | Minimum number of data points in each leaf for the summary (default 50)   |
| treeSummaryResidualThreshold          | Minimum residual in the summary (default 0 for positive residuals)  |
| treeSummaryResidualMagnitudeThreshold | Minimum residual magnitude in the summary (default 0 i.e. no restriction)   |
| doAllFactors                          | Flag to indicate whether to analyse the levels of all factor variables (default TRUE)   |
| maxFactorLevels                       | Maximum number of levels per factor before it is converted to numeric (default 20)  |
| useSubDir                             | Flag to specify whether the partition trees should be saved in the current directory or a subdirectory  |
| tempDirFolderName                     | specify a subfolder name if writing multiple scans to the temporary directory   |

**Value**

outputPath returns the location of the output for reference in addFeatures and for any other purpose. Saves residual CART trees and associated highlighted residuals for each to the path provided.

**Examples**

```
require(featurefinder)
data(futuresdata)
data=futuresdata
data$SMIFactor=paste("smi",as.matrix(data$SMIFactor),sep="")
n=length(data$DAX)
nn=floor(length(data$DAX)/2)

# Can we predict the relative movement of DAX and SMI?
data$y=data$DAX*0 # initialise the target to 0
data$y[1:(n-1)]=((data$DAX[2:n])-(data$DAX[1:(n-1)]))/
  (data$DAX[1:(n-1)]-(data$SMI[2:n]-(data$SMI[1:(n-1)]))/(data$SMI[1:(n-1)])

# Fit a simple model
thismodel=lm(formula=y ~ .,data=data)
```



```

expected=predict(thismodel,data)
actual=data$y
residual=actual-expected
data=cbind(data,expected, actual, residual)

CSVPath=tempdir()
fcsv=paste(CSVPath,"/futuresdata.csv",sep="")
write.csv(data[(nn+1):(length(data$y)),],file=fcsv,row.names=FALSE)
exclusionVars="\residual\","expected\","actual\","y\"
factorToNumericList=c()

# Now the dataset is prepared, try to find new features
findFeatures(outputPath="NoPath", fcsv, exclusionVars,factorToNumericList,
             treeGenerationMinBucket=50,
             treeSummaryMinBucket=20,
             useSubDir=FALSE)

newfeat1=((data$SMIfactor==0) & (data$CAC < 2253) & (data$CAC< 1998) & (data$CAC>=1882)) * 1.0
newfeat2=((data$SMIfactor==1) & (data$SMI < 7837) & (data$SMI >= 7499)) * 1.0
newfeatures=cbind(newfeat1, newfeat2) # create columns for the newly found features
datanew=cbind(data,newfeatures)
thismodel=lm(formula=y ~ .,data=datanew)
expectednew=predict(thismodel,datanew)

requireNamespace("Metrics")
OriginalRMSE = Metrics::rmse(data$y,expected)
NewRMSE = Metrics::rmse(data$y,expectednew)

print(paste("OriginalRMSE = ",OriginalRMSE))
print(paste("NewRMSE = ",NewRMSE))

```

---

futuresdata

*futuresdata*


---

## Description

Sample futures data based on dataset EuStockMarkets in the datasets package.

## Format

A data frame with 1860 rows and 4 variables

## Author(s)

Richard Davis <richard.davis@cba.com.au>

## Source

[stat.ethz.ch/R-manual/R-devel/library/datasets/html/00Index.html](http://stat.ethz.ch/R-manual/R-devel/library/datasets/html/00Index.html)

**Examples**

```
data(futuresdata)
head(futuresdata)
```

---

```
generateResidualCutoffCode
      generateResidualCutoffCode
```

---

**Description**

For each tree print a summary of the significant residuals as specified by the user

**Usage**

```
generateResidualCutoffCode(data, filename, trees, names, runname, ...)
```

**Arguments**

|                       |  |
|-----------------------|--|
| <code>data</code>     | A dataframe  |
| <code>filename</code> | A string   |
| <code>trees</code>    | A list of trees generated by <code>saveTree</code>                       |
| <code>names</code>    | A list of level names  |
| <code>runname</code>  | A string corresponding to the name of the factor variable being analysed |
| <code>...</code>      | and parameters to be passed through                                      |

**Value**

A list of residuals for each tree provided.

**Examples**

```
require(featurefinder)
data(examples)
generateResidualCutoffCode(data=dat0, "treesAll.txt", treesAll, mainfaclevels, runname,
  treeGenerationMinBucket=treeGenerationMinBucket,
  treeSummaryMinBucket=treeSummaryMinBucket,
  treeSummaryResidualThreshold=treeSummaryResidualThreshold,
  treeSummaryResidualMagnitudeThreshold=treeSummaryResidualMagnitudeThreshold,
  doAllFactors=doAllFactors,
  maxFactorLevels=maxFactorLevels)
```

---

|               |                      |
|---------------|----------------------|
| generateTrees | <i>generateTrees</i> |
|---------------|----------------------|

---

## Description

Generate a residual tree for each level of factor mainfac

## Usage

```
generateTrees(data, vars, expr, outputPath, runname, ...)
```

## Arguments

|            |   |
|------------|---|
| data       | A dataframe   |
| vars       | A list of candidate predictors                                    |
| expr       | A expression to be modelled by the RPART tree                     |
| outputPath | The output directory  |
| runname    | A string corresponding to the name of the variable being modelled |
| ...        | and parameters to be passed through                               |

## Value

A list of residual trees for each level of the mainfac factor provided

## Examples

```
require(featurefinder)
data(examples)
treesThisvar=generateTrees(data=dat0, vars, expr, outputPath=tempdir(), runname,
  treeGenerationMinBucket=treeGenerationMinBucket,
  treeSummaryMinBucket=treeSummaryMinBucket,
  treeSummaryResidualThreshold=treeSummaryResidualThreshold,
  treeSummaryResidualMagnitudeThreshold=treeSummaryResidualMagnitudeThreshold,
  doAllFactors=doAllFactors,
  maxFactorLevels=maxFactorLevels)
```

---

`getVarAv`*getVarAv*

---

**Description**

This function generates a residual tree on a subset of the data

**Usage**

```
getVarAv(dd, varAv, varString)
```

**Arguments**

|                        |  |
|------------------------|--|
| <code>dd</code>        | A dataframe  |
| <code>varAv</code>     | A string corresponding to the numeric field to be averaged within each leaf node |
| <code>varString</code> | A string   |

**Value**

An average of the numeric variable `varString` in the segment

**Examples**

```
require(featurefinder)
data(examples)
av=getVarAv(dat,"expected",pathterms)
```

---

`i`*i*

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

An index variable used in examples.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(i)
head(i)
```

---

|               |                      |
|---------------|----------------------|
| mainfaclevels | <i>mainfaclevels</i> |
|---------------|----------------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

Levels of the main or current factor being scanned.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(mainfaclevels)
head(mainfaclevels)
```

---

|                 |                        |
|-----------------|------------------------|
| maxFactorLevels | <i>maxFactorLevels</i> |
|-----------------|------------------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

Maximum allowable number of factor levels before the variable is converted to numeric.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(maxFactorLevels)
head(maxFactorLevels)
```

---

|         |                |
|---------|----------------|
| mpgdata | <i>mpgdata</i> |
|---------|----------------|

---

**Description**

Sample car data based on dataset mpg in the ggplot2 package

**Format**

A data frame with 234 rows and 11 variables

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(mpgdata)
head(mpgdata)
```

---

|       |              |
|-------|--------------|
| names | <i>names</i> |
|-------|--------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

A list of variable names used in examples.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(names)  
head(names)
```

---

|                          |                    |
|--------------------------|--------------------|
| <code>parseSplits</code> | <i>parseSplits</i> |
|--------------------------|--------------------|

---

**Description**

Extract information relating to the paths and volume of data in the leaves of the tree

**Usage**

```
parseSplits(thistree)
```

**Arguments**

`thistree`      A tree

**Value**

A list of parsed splits.

**Examples**

```
require(featurefinder)  
data(examples)  
parseSplits(treesAll[[1]][[2]])
```

---

|                        |                  |
|------------------------|------------------|
| <code>pathterms</code> | <i>pathterms</i> |
|------------------------|------------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

A string defining a leaf node formula.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(pathterms)
head(pathterms)
```

---

```
printResiduals      printResiduals
```

---

**Description**

This function generates a residual tree on a subset of the data

**Usage**

```
printResiduals(fileConn, all, dat, runname, levelname,
  treeSummaryResidualThreshold, treeSummaryMinBucket,
  treeSummaryResidualMagnitudeThreshold, ...)
```

**Arguments**

|  |   |
|--|---|
| <code>fileConn</code>                              | A file connection   |
| <code>all</code>                                   | A dataframe   |
| <code>dat</code>                                   | The dataset   |
| <code>runname</code>                               | A string corresponding to the name of the factor being analysed |
| <code>levelname</code>                             | A string corresponding to the factor level being analysed       |
| <code>treeSummaryResidualThreshold</code>          | The minimum residual threshold                                  |
| <code>treeSummaryMinBucket</code>                  | The minimum volume per leaf                                     |
| <code>treeSummaryResidualMagnitudeThreshold</code> | Minimum residual magnitude                                      |
| <code>...</code>                                   | and parameters to be passed through                             |

**Value**

Residuals are printed and also saved in a simplified format.

**Examples**

```
require(featurefinder)
data(examples)
printResiduals(fileConn,splitlist[t][[1]],dat, runname, names[t],
  treeSummaryResidualThreshold,treeSummaryMinBucket,
  treeSummaryResidualMagnitudeThreshold)
```



---

|         |                |
|---------|----------------|
| runname | <i>runname</i> |
|---------|----------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

A string corresponding to the name of the variable being modelled

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(runname)
head(runname)
```

---

|          |                 |
|----------|-----------------|
| saveTree | <i>saveTree</i> |
|----------|-----------------|

---

**Description**

Generate a residual tree on a subset of the data specified by the factor level mainfaclev (main factor level)

**Usage**

```
saveTree(data, vars, expr, i, outputPath, varname, mainfaclev,
          treeGenerationMinBucket, ...)
```

**Arguments**

|            |  |
|------------|--|
| data       | A dataframe containing the residual and some predictors                  |
| vars       | A list of candidate predictors   |
| expr       | A expression to be modelled by the RPART tree                            |
| i          | An integer corresponding to the factor level                             |
| outputPath | The output directory   |
| varname    | A string corresponding to the name of the factor variable being analysed |

mainfaclev      A level of the mainfac factor  
treeGenerationMinBucket  
                 Minimum size for tree generation  
...              and parameters to be passed through

**Value**

A tree object

**Examples**

```
require(featurefinder)
data(examples)
fit1=saveTree(data, vars, expr, i, outputPath=tempdir(), runname, mainfaclevels[1],
              treeGenerationMinBucket)
```

---

splitlist

*splitlist*

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

Sample list of node split formulae.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(splitlist)
head(splitlist)
```

---

|   |          |
|---|----------|
| t | <i>t</i> |
|---|----------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

A sample tree.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(t)
head(t)
```

---

|      |             |
|------|-------------|
| tree | <i>tree</i> |
|------|-------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

A sample tree object.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(tree)
head(tree)
```

---

treeGenerationMinBucket

*treeGenerationMinBucket*

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

Minimum number of data points per leaf allowed in tree generation.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(treeGenerationMinBucket)
head(treeGenerationMinBucket)
```

---

trees

*trees*

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

Sample tree set.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(trees)
head(trees)
```

---

|          |                 |
|----------|-----------------|
| treesAll | <i>treesAll</i> |
|----------|-----------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

Full dataset tree example.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(treesAll)
head(treesAll)
```

---

|                      |                             |
|----------------------|-----------------------------|
| treeSummaryMinBucket | <i>treeSummaryMinBucket</i> |
|----------------------|-----------------------------|

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

Minimum number of data points per leaf allowed in tree summary.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(treeSummaryMinBucket)
head(treeSummaryMinBucket)
```

treeSummaryResidualMagnitudeThreshold  
*treeSummaryResidualMagnitudeThreshold*

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

Minimum allowed residual magnitude in leaf summary generation.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(treeSummaryResidualMagnitudeThreshold)
head(treeSummaryResidualMagnitudeThreshold)
```

---

treeSummaryResidualThreshold  
*treeSummaryResidualThreshold*

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

Minimum allowed residual value in leaf summary generation.

**Author(s)**

Richard Davis <richard.davis@cba.com.au>

**Source**

[ggplot2.org](http://ggplot2.org)

**Examples**

```
data(treeSummaryResidualThreshold)
head(treeSummaryResidualThreshold)
```

---

vars

*vars*

---

**Description**

Sample data based on dataset mpg in the ggplot2 package

**Format**

List of predictor variables.

**Author(s)**

Richard Davis <[richard.davis@cba.com.au](mailto:richard.davis@cba.com.au)>

**Source**

[ggplot2.org](https://ggplot2.org)

**Examples**

```
data(vars)
head(vars)
```

# Index

- \* **addFeatures**
    - addFeatures, 2
  - \* **dat0**
    - dat0, 4
  - \* **data**
    - data, 5
  - \* **dat**
    - dat, 4
  - \* **doAllFactors**
    - doAllFactors, 5
  - \* **expr**
    - expr, 6
  - \* **fileConn**
    - fileConn, 6
  - \* **filename**
    - filename, 7
  - \* **findFeatures**
    - findFeatures, 7
  - \* **futuresdata**
    - futuresdata, 9
  - \* **generateTrees**
    - generateTrees, 11
  - \* **i**
    - i, 12
  - \* **mainfaclevels**
    - mainfaclevels, 13
  - \* **maxFactorLevels**
    - maxFactorLevels, 13
  - \* **mpgdata**
    - mpgdata, 14
  - \* **names**
    - names, 14
  - \* **pathterms**
    - pathterms, 15
  - \* **runname**
    - runname, 17
  - \* **saveTree**
    - generateResidualCutoffCode, 10
    - getVarAv, 12
    - parseSplits, 15
    - printResiduals, 16
    - saveTree, 17
  - \* **splitlist**
    - splitlist, 18
  - \* **treeGenerationMinBucket**
    - treeGenerationMinBucket, 20
  - \* **treeSummaryMinBucket**
    - treeSummaryMinBucket, 21
  - \* **treeSummaryResidualMagnitudeThreshold**
    - treeSummaryResidualMagnitudeThreshold, 22
  - \* **treeSummaryResidualThreshold**
    - treeSummaryResidualThreshold, 22
  - \* **treesAll**
    - treesAll, 21
  - \* **trees**
    - trees, 20
  - \* **tree**
    - tree, 19
  - \* **t**
    - t, 19
  - \* **vars**
    - vars, 23
- addFeatures, 2
- dat, 4
- dat0, 4
- data, 5
- doAllFactors, 5
- expr, 6
- fileConn, 6
- filename, 7
- findFeatures, 7
- futuresdata, 9
- generateResidualCutoffCode, 10



generateTrees, 11  
getVarAv, 12  
  
i, 12  
  
mainfaclevels, 13  
maxFactorLevels, 13  
mpgdata, 14  
  
names, 14  
  
parseSplits, 15  
pathterms, 15  
printResiduals, 16  
  
runname, 17  
  
saveTree, 17  
splitlist, 18  
  
t, 19  
tree, 19  
treeGenerationMinBucket, 20  
trees, 20  
treesAll, 21  
treeSummaryMinBucket, 21  
treeSummaryResidualMagnitudeThreshold,  
22  
treeSummaryResidualThreshold, 22  
  
vars, 23