Package ‘otvPlots’

June 26, 2018

Title Over Time Variable Plots
Version 0.2.1
Description Enables automated visualization of variable
distribution and changes over time for predictive model building.
It efficiently computes summary statistics aggregated by time for
large datasets, and create plots for variable level monitoring.
Depends R (>= 3.2.0)
Imports data.table (>= 1.9.6), ggplot2 (>= 2.1.0), grid (>= 3.2.0),
gridExtra (>= 2.2.1), Hmisc (>= 3.17-4), moments, quantreg (>=
5.33), scales (>= 0.4.0), stringi (>= 1.1.1)
License Apache License 2.0 | file LICENSE
LazyData true
Suggests bit64, knitr, proto, testthat
URL https://github.com/capitalone/otvPlots
BugReports https://github.com/capitalone/otvPlots/issues
RoxygenNote 6.0.1
NeedsCompilation no
Author Rebecca Payne [aut],
Zoey Zhu [aut],
Yingbo Li [aut, cre],
Capital One [cph]
Maintainer Yingbo Li <yingbo.li@capitalone.com>
Repository CRAN
Date/Publication 2018-06-26 19:59:09 UTC

R topics documented:

bankData ............................................. 2
bankLabels .......................................... 3
CalcR2 .............................................. 4
**Description**

The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. Records are ordered by date (from May 2008 to November 2010), similar to data analyzed in Moro et al. [2014].

**Usage**

bankData

**Format**

A data frame with 45,211 rows and 19 variables:

- **age** Age of the client, numeric.
- **job** Type of job, a categorical variable with the levels: 'admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', and 'unknown'.
- **marital** Marital status, a categorical variable with levels: 'divorced', 'married', 'single', and 'unknown'. Note that 'divorced' means either divorced or widowed.
- **education** A categorical variable with levels: 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'illiterate', 'professional.course', 'university.degree', and 'unknown'.
- **default** Whether credit is in default, a categorical variable with levels: 'no', 'yes', and 'unknown'.
- **balance** Account balance, numeric.
**bankLabels**

**housing** Whether the client has a housing loan, a categorical variable with levels: 'no', 'yes', and 'unknown'.

**loan** Whether the client has personal loan, a categorical variable with levels: 'no', 'yes', and 'unknown'.

**contact** Type of contact communication, a categorical variable with levels: 'cellular' and 'telephone'.

**duration** Last contact duration in seconds, a numeric variable.

**campaign** Number of contacts performed during this campaign for this client, including the last contact; a numeric variable.

**pdays** Number of days that passed by after the client was last contacted from a previous campaign; a numeric variable, with 999 means that client was not previously contacted.

**previous** Number of contacts performed before this campaign for this client, a numeric variable.

**poutcome** Outcome of the previous marketing campaign, a categorical variable with levels: 'failure', 'nonexistent', and 'success'.

**y** Whether the client has subscribed a term deposit, a categorical variable with levels: 'yes' and 'no'.

**date** Last contact date.

**Source**

https://archive.ics.uci.edu/ml/datasets/Bank+Marketing


---

**bankLabels**

*Labels for bankData*

**Description**

A dataset containing the attribute labels also found in bankData. This data set is used to illustrate the PrepLabels function and other label functionality in the otvPlots package in R.

**Usage**

bankLabels

**Format**

A data frame with 16 rows and 3 variables:

- **V1** Name of each variable in bankData.
- **V2** Label of each variable in bankData.
- **V3** A numeric variable, corresponding to the row number.
CalcR2

Calculates \( R^2 \) of a numerical variable using date as the predictor

Description

Calculates weighted \( R^2 \) of a univariate weighted linear model with dateNm as x and myVar as y using the workhorse \( \text{lm.fit} \) and \( \text{lm.wfit} \) functions.

Usage

\[
\text{CalcR2}(\text{myVar}, \text{dataF1}, \text{dateNm}, \text{weightNm} = \text{NULL}, \text{imputeValue} = \text{NULL})
\]

Arguments

- **myVar**: Name of variable to model.
- **dataF1**: A data.table, containing myVar, dateNm, and weightNm.
- **dateNm**: Name of column containing the date variable (to be modeled as numeric); this date column must not have NA's.
- **weightNm**: Name of column containing row weights. If weights equal one, then the \( \text{lm.fit} \) function will be called, otherwise the \( \text{lm.wfit} \) will be called. The weights column must not have NA's.
- **imputeValue**: Either NULL or numeric. If NULL, model will be fit on only non-NA components of myVar. If numeric, missing cases of myVar will be imputed to imputeValue.

Value

A numeric value of \( R^2 \).

License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

See Also

Functions depend on this function: \text{OrderByR2}.

This function depends on: \text{PrepData}.
OrderByR2

Create numerical variable ranking using R2 between date to and variable

Description

Calculates R2 of a linear model of the formula var ~ dateNm for each var of class nmrcl and returns a vector of variable names ordered by highest R2. The linear model can be calculated over a subset of dates, see details of parameter buildTm. Non-numerical variables are returned in alphabetical order after the sorted numerical variables.

Usage

OrderByR2(dataFl, dateNm, buildTm = NULL, weightNm = NULL, kSample = 50000)

Arguments

dataFl A data table of data; must be the output of the PrepData function.
dateNm Name of column containing the date variable.
buildTm Vector identify time period for ranking/anomaly detection (most likely model build period). Allows for a subset of plotting time period to be used for anomaly detection.
  • Must be a vector of dates and must be inclusive i.e. buildTm[1] <= date <= buildTm[2] will define the time period.
  • Must be either NULL, a vector of length 2, or a vector of length 3.
  • If NULL, the entire dataset will be used for ranking/anomaly detection.
  • If a vector of length 2, the format of the dates must be a character vector in default R date format (e.g. "2017-01-30").
  • If a vector of length 3, the first two columns must contain dates in any strptime format, while the 3rd column contains the strptime format (see strptime).
  • The following are equivalent ways of selecting all of 2014:
    – c("2014-01-01", "2014-12-31")
    – c("01JAN2014", "31DEC2014", "%d%h%Y")
weightNm Name of the variable containing row weights, or NULL for no weights (all rows receiving weight 1).
kSample Either NULL or a positive integer. If an integer, indicates the sample size for both drawing boxplots and ordering numerical graphs by $R^2$. When the data is large, setting kSample to a reasonable value (default is 50K) dramatically improves processing speed. Therefore, for larger datasets (e.g. > 10 percent system memory), this parameter should not be set to NULL, or boxplots may take a very long time to render. This setting has no impact on the accuracy of time series plots on quantiles, mean, SD, and missing and zero rates.
Value

A vector of variable names sorted by R2 of lm of the formula var ~ datenm (highest R2 to lowest)

License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

See Also

Functions depend on this function: v1m.

This function depends on: CalcR2, PrepData.

Examples

data(bankData)
bankData <- PrepData(bankData, datenm = "date", dateGp = "months",
                     dateGpBp = "quarters")
OrderByR2(bankData, datenm = "date")

Description

The otvPlots package uses data.table and ggplot2 packages to efficiently plot time series aggregated from large datasets. Plots of numerical variables are optionally returned ordered by correlation with date – a natural starting point for anomaly detection. Plots are automatically labeled if a variable dictionary is provided.

Details

Output files include:

- A PDF file of plots saved as outfl.pdf, with each individual page on one variable. Variables are plotted in the order indicated in the argument sortVars or sortFn. For each numerical variable, the output plots include
  - side-by-side boxplots grouped by dateGpBp (left),
  - a trace plot of p1, p50, and p99 percentiles, grouped by dateGp (top right),
  - a trace plot of mean and +/-1 SD control limits, grouped by dateGp (middle right), and
  - a trace plot of missing and zero rates, grouped by dateGp (bottom right).

For each categorical variable (including a numerical variable with no more than 2 unique levels not including NA), the output plots include
– a frequency bar plot (left), and
– a grid of trace plots on categories’ proportions over time (right). If the variable contains more than k categories number of categories, trace plots of only the largest k categories will be plotted. If the variable contains only two categories, then only the trace plot of the less prevalent category will be plotted.

• CSV file(s) on summary statistics of variables, both globally and over time aggregated by dateGp. The order of variables in the CSV files is the same as in the PDF file.
  – For numerical variables, number of observations (counts), p1, p25, p50, p75, and p99 quantiles, mean, SD, missing and zeros rates are saved as outF1_numerical_summary.csv.
  – For categorical variables, number of observations (counts) and categories’ proportions are saved as outF1_categorical_summary.csv. Each row is a category of a categorical (or binary) variable. The row whose category \( \text{category} = '\text{NA}' \) corresponds to missing. Categories among the same variable are ordered by global prevalence in a descending order.

License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

See Also

Main function: vlm.
Selected supporting functions: PrepData, PrepLabels, OrderByR2.

### PlotBarplot

**Creates a bar plot for a discrete (or binary) variable**

**Description**

Creates a bar plot for a discrete (or binary) variable

**Usage**

```r
PlotBarplot(dataF1, myVar, weightNm = NULL)
```

**Arguments**

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dataF1</td>
<td>A data.table of data; must be the output of the PrepData function.</td>
</tr>
<tr>
<td>myVar</td>
<td>The name of the variable to be plotted</td>
</tr>
<tr>
<td>weightNm</td>
<td>Name of the variable containing row weights, or NULL for no weights (all rows receiving weight 1).</td>
</tr>
</tbody>
</table>
Value

A ggplot object with a histogram of myVar ordered by category frequency

License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

See Also

Functions depend on this function: PlotCatVar.

This function depends on: PrepData.

Examples

data(bankData)
bankData = PrepData(bankData, dateNm = "date", dateGp = "months",
                    dateGpBp = "quarters", weightNm = NULL)
PlotBarplot(bankData, "job")

## NA will be included as a category if any NA are present
bankData[sample.int(.N)[1:1000], education := NA]
PlotBarplot(bankData, "education")

---

**PlotCatVar**

Create plots and summary statistics for a categorical variable

Description

Output plots include a bar plot with categories ordered by global counts, and trace plots of categories' proportions over time. This function is also applicable to a binary variable, which is treated as categorical in this package. In addition to plots, a data table of summary statistics are generated, on global counts and proportions by category, and proportions by category over time.

Usage

PlotCatVar(myVar, dataFl, weightNm = NULL, dateNm, dateGp, kCategories = 9,
           normBy = "time")
Arguments

myVar The name of the variable to be plotted

dataFl A data.table of data; must be the output of the PrepData function.

weightNm Name of the variable containing row weights, or NULL for no weights (all rows receiving weight 1).

dateNm Name of column containing the date variable.

dateGp Name of the variable that the time series plots should be grouped by. Options are NULL, "weeks", "months", "quarters", "years". See IDate for details. If NULL, then dateNm will be used as dateGp.

kCategories If a categorical variable has more than kCategories, trace plots of only the kCategories most prevalent categories are plotted.

normBy The normalization factor for rate plots, can be "time" or "var". If "time", then for each time period of dateGp, counts are normalized by the total counts over all categories in that time period. This illustrates changes of categories’ proportions over time. If "var", then for each category, its counts are normalized by the total counts over time from only this category. This illustrates changes of categories’ volumes over time.

Value

p A grob (i.e., ggplot grid) object, including a bar plot, and trace plots of categories’ proportions. If the number of categories is larger than kCategories, then trace plots of only the kCategories most prevalent categories are plotted. For a binary variable, only the trace plot of the less prevalent category is plotted.

catVarSummary A data.table, contains categories’ proportions globally, and over-time in each time period in dateGp. Each row is a category of the categorical (or binary) variable myVar. The row whose category == 'NA' corresponds to missing. Categories are ordered by global prevalence in a descending order.

License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

See Also

Functions depend on this function: PlotVar, PrintPlots, vlm.

This function depends on: PlotBarplot, PlotRatesOverTime, PrepData.
Examples

data(bankData)
bankData <- PrepData(bankData, dateNm = "date", dateGp = "months",
    dateGpBp = "quarters", weightNm = NULL)
# Single histogram is plotted for job type since there are 12 categories
plot(PlotCatVar(myVar = "job", dataFl = bankData, weightNm = NULL,
    dateNm = "date", dateGp = "months")$p)

plot(PlotCatVar(myVar = "job", dataFl = bankData, weightNm = NULL,
    dateNm = "date", dateGp = "months", kCategories = 12)$p)

## Binary data is treated as categorical, and only the less frequent
## category is plotted over time.
plot(PlotCatVar(myVar = "default", dataFl = bankData, weightNm = NULL,
    dateNm = "date", dateGp = "months")$p)

---

**PlotDist**

Side-by-side box plots, for a numerical variable, grouped by dateGpBp

Description

For a variable is all positive (no zeros) and has larger than 50 all distinct values, if it is highly skewed, then all box plots can be plotted under the log base 10 transformation. See the argument skewOpt for details.

Usage

`PlotDist(dataFl, myVar, dateGpBp, weightNm = NULL, skewOpt = NULL)`

Arguments

- **dataFl**: A data.table of data; must be the output of the PrepData function.
- **myVar**: The name of the variable to be plotted.
- **dateGpBp**: Name of variable the boxplots should be grouped by. Same options as dateGp. If NULL, then dateGp will be used.
- **weightNm**: Name of the variable containing row weights, or NULL for no weights (all rows receiving weight 1).
- **skewOpt**: Either a numeric constant or NULL. Default is NULL (no transformation). If numeric, say 5, then all box plots of a variable whose skewness exceeds 5 will be on a log10 scale if possible. Negative input of skewOpt will be converted to 3.

Value

A ggplot2 object with a box plot of myVar grouped by dateGpBp
License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Examples

data(bankData)
bankData <- PrepData(bankData, dateNm = "date", dateGp = "months",
                    dateGpBp = "quarters")
PlotDist(datafl = bankData, myVar = "balance", dateGpBp = "quarters")
## The following attempt to log transform will fail due to negative values,
## and the untransformed version will be returned
PlotDist(datafl = bankData, myVar = "balance", dateGpBp = "quarters",
         skewOpt = 3)
## This attempt should succeed, as the skew exceeds 3 and there are no
## negative values
PlotDist(datafl = bankData, myVar = "duration", dateGpBp = "quarters",
         skewOpt = 3)

PlotMean

Plot mean with Mean +- 1SD control limits for a numerical variable

Description

Plot mean with Mean +- 1SD control limits for a numerical variable

Usage

PlotMean(meltdx, myVar, dateGp)

Arguments

meltdx A data.table with Mean and 1SD control limits in long format, produced by
SummaryStats
myVar The name of the variable to be plotted
dateGp Name of the variable that the time series plots should be grouped by. Options
        are NULL, "weeks", "months", "quarters", "years". See IDate for details. If
        NULL, then dateNm will be used as dateGp.

Value

A ggplot2 object with dateGp on the x axis, value on the y axis, and variables Mean, c11, and c12
plotted on the same graph, with mean and control limits differentiated by line type.
License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

PlotNumVar

Create plots and summary statistics for a numerical variable

Description

Output plots include a boxplot on the left, grouped by a courser time scale (dateGpBp), and three trace plots on the right, on p1, p50, and p99 quantiles, mean and +1 SD control limits, missing and zerorates, all grouped by a finer time scale as in dateGp. In addition to plots, a data table of summary statistics are generated, on global and over time summary statistics.

Usage

PlotNumVar(myVar, dataFl, weightNm, dateGp, dateGpBp, skewOpt = NULL, kSample = 50000)

Arguments

myVar The name of the variable to be plotted

dataFl A data table of data; must be the output of the PrepData function.

weightNm Name of the variable containing row weights, or NULL for no weights (all rows receiving weight 1).

dateGp Name of the variable that the time series plots should be grouped by. Options are NULL, "weeks", "months", "quarters", "years". See IDate for details. If NULL, then dateNm will be used as dateGp.

dateGpBp Name of variable the boxplots should be grouped by. Same options as dateGp. If NULL, then dateGp will be used.

skewOpt Either a numeric constant or NULL. Default is NULL (no transformation). If numeric, say 5, then all box plots of a variable whose skewness exceeds 5 will be on a log10 scale if possible. Negative input of skewOpt will be converted to 3.

kSample Either NULL or a positive integer. If an integer, indicates the sample size for both drawing boxplots and ordering numerical graphs by $R^2$. When the data is large, setting kSample to a reasonable value (default is 50K) dramatically improves processing speed. Therefore, for larger datasets (e.g. > 10 percent system memory), this parameter should not be set to NULL, or boxplots may take a very long time to render. This setting has no impact on the accuracy of time series plots on quantiles, mean, SD, and missing and zero rates.
Value

`p` A grob (i.e., `ggplot2` grid) object, including a side-byside boxplot grouped by `dateGpBp`, a time series plot of `p1`, `p50` (median), and `p99` grouped by `dateGp`, a time series plot of mean and ±1 SD control limits grouped by `dateGp`, and a time series plot of missing and zerorates grouped by `dateGp`.

`numVarSummary` A data.table, contains global and over time summary statistics, including `p1`, `p25`, `p50`, `p75`, and `p99` quantiles, mean and SD, missing and zero rates.

License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

See Also

Functions depend on this function: `PlotVar`.

This function depends on: `SummaryStats`, `PlotDist`, `PlotQuantiles`, `PlotMean`, `PlotRates`, `PrepData`.

Examples

data(bankData)
bankData <- PrepData(bankData, dateNm = "date", dateGp = "months",
dateGpBp = "years")
plot(PlotNumVar("balance", bankData, NULL, "months", "years",
               skewOpt = NULL, kSample = NULL)$p)

---

**PlotQuantiles**

Plot 01, 50, and 99 percentile for a numerical variable

Description

Plot 01, 50, and 99 percentile for a numerical variable

Usage

`PlotQuantiles(meltdx, myVar, dateGp)`
**PlotRates**

**Description**

Plot zero and missing rates for a numerical variable

**Usage**

`PlotRates(meltdx, myVar, dateGp)`

**Arguments**

- **meltdx**: A data.table with missing rate and zero rate in long format, produced by `SummaryStats`
- **myVar**: The name of the variable to be plotted
- **dateGp**: Name of the variable that the time series plots should be grouped by. Options are NULL, "weeks", "months", "quarters", "years". See `IDate` for details. If NULL, then `dateNm` will be used as `dateGp`.

**Value**

A ggplot2 object with a missing rate and zero rate grouped by `dateGp`. 
PlotRatesOverTime

License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

PlotRatesOverTime

`PlotRatesOverTime` creates trace plots of categories’ proportions over time for a discrete (or binary) variable.

Description

Creates trace plots of categories’ proportions over time for a discrete (or binary) variable.

Usage

```r
PlotRatesOverTime(dataFl, dateGp, myVar, normBy = "time", weightNm = NULL, newLevels = NULL, kCategories = 9)
```

Arguments

- `dataFl`: A data table of data; must be the output of the `PrepData` function.
- `dateGp`: Name of the variable that the time series plots should be grouped by. Options are `NULL`, "weeks", "months", "quarters", "years". See `IDate` for details. If `NULL`, then `dateNm` will be used as `dateGp`.
- `myVar`: The name of the variable to be plotted.
- `normBy`: The normalization factor for rate plots, can be "time" or "var". If "time", then for each time period of `dateGp`, counts are normalized by the total counts over all categories in that time period. This illustrates changes of categories’ proportions over time. If "var", then for each category, its counts are normalized by the total counts over time from only this category. This illustrates changes of categories’ volumes over time.
- `weightNm`: Name of the variable containing row weights, or `NULL` for no weights (all rows receiving weight 1).
- `newLevels`: categories of `myVar` in order of global frequency.
- `kCategories`: If a categorical variable has more than `kCategories`, trace plots of only the `kCategories` most prevalent categories are plotted.
### Value

A list:

- `p`: ggplot object, trace plots of categories’ proportions `myVar` over time.
- `catVarSummary`: A `data.table`, contains categories’ proportions globally, and over-time in each time period in `dateGp`. Each row is a category of the categorical (or binary) variable `myVar`. The row whose category == ’NA’ corresponds to missing. Categories are ordered by global prevalence in a descending order.

### License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

### See Also

Functions depend on this function: `PlotCatVar`.

This function depends on: `PrepData`.

### Examples

```r
data(bankData)
bankData$weight = rpois(nrow(bankData), 5)
bankData <- PrepData(bankData, dateNm = "date", dateGp = "months",
dateGpBp = "quarters", weightNm = "weight")
PlotRatesOverTime(dataFl = bankData, dateGp = "months", weightNm = "weight",
myVar = "job", newLevels = NULL, normBy = "time")
```

### Description

For a numerical variable, the output includes

- side-by-side boxplots grouped by `dateGpBp` (left),
- a trace plot of p1, p50, and p99 percentiles, grouped by `dateGp` (top right),
- a trace plot of mean and +/-1 SD control limits, grouped by `dateGp` (middle right), and
- a trace plot of missing and zero rates, grouped by `dateGp` (bottom right).
For a categorical variable (including a numerical variable with no more than 2 unique levels not including NA), the output includes

- a frequency bar plot (left), and
- a grid of trace plots on categories’ proportions over time (right). If the variable contains more than kCategories number of categories, trace plots of only the largest kCategories will be plotted.

In addition to plots, a data.table of summary statistics are generated, on global and over time summary statistics.

Usage

PlotVar(dataFl, myVar, weightNm, dateNm, dateGp, dateGpBp = NULL,
        labelFl = NULL, highlightNms = NULL, skewOpt = NULL, kSample = 50000,
        fuzzyLabelFn = NULL, kCategories = 9)

Arguments

dataFl A data.table containing at least the following columns: myVar, weightNm, dateGp, dateGpBp; usually an output of the PrepData function.

myVar Name of the variable to be plotted.

weightNm Name of the variable containing row weights, or NULL for no weights (all rows receiving weight 1).

dateNm Name of column containing the date variable.

dateGp Name of the variable that the time series plots should be grouped by. Options are NULL, "weeks", "months", "quarters", "years". See IDate for details. If NULL, then dateNm will be used as dateGp.

dateGpBp Name of variable the boxplots should be grouped by. Same options as dateGp. If NULL, then dateGp will be used.

labelFl A data.table containing variable labels, or NULL for no labels; usually an output of PrepLabels.

highlightNms Either NULL or a character vector of variables to receive red label. Currently NULL means all variables will get a black legend. Ignored this argument if labelFl == NULL.

skewOpt Either a numeric constant or NULL. Default is NULL (no transformation). If numeric, say 5, then all box plots of a variable whose skewness exceeds 5 will be on a log10 scale if possible. Negative input of skewOpt will be converted to 3.

kSample Either NULL or a positive integer. If an integer, indicates the sample size for both drawing boxplots and ordering numerical graphs by $R^2$. When the data is large, setting kSample to a reasonable value (default is 50K) dramatically improves processing speed. Therefore, for larger datasets (e.g. > 10 percent system memory), this parameter should not be set to NULL, or boxplots may take a very long time to render. This setting has no impact on the accuracy of time series plots on quantiles, mean, SD, and missing and zero rates.
fuzzyLabelFn

Either NULL or a function of 2 parameters: A label file in the format of an output by \texttt{PrepLabels} and a string giving a variable name. The function should return the label corresponding to the variable given by the second parameter. This function should describe how fuzzy matching should be performed to find labels (see example below). If NULL, only exact matches will be returned.

kCategories

If a categorical variable has more than \texttt{kCategories}, trace plots of only the \texttt{kCategories} most prevalent categories are plotted.

Value

\begin{itemize}
\item \texttt{p} \quad \text{A \texttt{ggplot} (i.e., \texttt{ggplot2}) object. See the output \texttt{p} of the function or \texttt{PlotNumVar}}, \texttt{PlotCatVar} for details.
\item \texttt{varSummary} \quad \text{A \texttt{data.frame} of summary statistics. See the output \texttt{numVarSummary} of the function \texttt{PlotNumVar}, or the output \texttt{catVarSummary} of the function \texttt{PlotCatVar} for details.}
\item \texttt{varType} \quad \text{Indicator of the variable’s type, either “nmrcl” or “ctgrl”}
\end{itemize}

License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the “License”); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

See Also

Functions depend on this function: \texttt{PrintPlots}.

This function depends on: \texttt{PlotCatVar, PlotNumVar, PrepData}.

Examples

\begin{verbatim}
data(bankData)
bankData <- PrepData(bankData, dateNm = "date", dateGp = "months", dateGpBp = "quarters")
data(bankLabels)
bankLabels <- PrepLabels(bankLabels)

## PlotVar will treat numerical and categorical data differently.
## Binary data is always treated as categorical.
plot(PlotVar(bankData, myVar = "duration", weightNm = NULL, dateNm = "date",
           dateGp = "months", dateGpBp = "quarters", labelFl = bankLabels))
plot(PlotVar(bankData, myVar = "job", weightNm = NULL, dateNm = "date",
           dateGp = "months", dateGpBp = "quarters", labelFl = bankLabels))
plot(PlotVar(bankData, myVar = "loan", weightNm = NULL, dateNm = "date",
           dateGp = "months", dateGpBp = "quarters", labelFl = bankLabels))
\end{verbatim}
**PrepData**

Prepare an input dataset for plotting

**Description**

This function prepares an input dataset for use by all plotting functions in this package, including the main function `vlm`. The input data `dataF1` must contain, at a minimum, a date column `datenm` and a variable to be plotted. `dataF1` will be converted to a `data.table` class, and all changes are made to it by reference.

**Usage**

```r
PrepData(dataF1, datenm, selectCols = NULL, dropCols = NULL, 
dateFt = "%d%h%M", dateGp = NULL, dateGpBp = NULL, weightNm = NULL, 
varNms = NULL, dropConstants = FALSE, ...) 
```

**Arguments**

- `dataF1`: Either the name of an object that can be converted using `as.data.table` (e.g., a data frame), or a character string containing the name of dataset that can be loaded using `fread` (e.g., a csv file). If the dataset is not in your working directory then `dataF1` must include (relative or absolute) path to file.
- `datenm`: Name of column containing the date variable.
- `selectCols`: Either `NULL`, or a vector of names or indices of variables to read into memory – must include `datenm`, `weightNm` (if not `NULL`) and all variables to be plotted. If both `selectCols` and `dropCols` are `NULL`, then all variables will be read in.
- `dropCols`: Either `NULL`, or a vector of variables names or indices of variables not to read into memory. If both `selectCols` and `dropCols` are `NULL`, then all variables will be read in.
- `dateFt`: `strptime` format of date variable. The default is SAS format "%d%h%M". But input data with R date format "%Y-%m-%d" will also be detected. Both of two formats can be parsed automatically.
- `dateGp`: Name of the variable that the time series plots should be grouped by. Options are `NULL`, "weeks", "months", "quarters", "years". See `IDate` for details. If `NULL`, then `datenm` will be used as `dateGp`.
- `dateGpBp`: Name of variable the boxplots should be grouped by. Same options as `dateGp`. If `NULL`, then `dateGp` will be used.
- `weightNm`: Name of the variable containing row weights, or `NULL` for no weights (all rows receiving weight 1).
- `varNms`: Either `NULL` or a vector of names or indices of variables to be plotted. If `NULL`, will default to all columns which are not `datenm` or `weightNm`. Can also be a vector of indices of the column names, after `dropCols` or `selectCols` have been applied, if applicable, and not including `dateGp`, `dateGpBp` (which will be added to the `dataF1` by the function `PrepData`).
PrepLabels

Description

This function prepares a dataset containing variable labels for use by the main plotting function `vlm`. The input must contain variables’ names in the first column and labels in the second column. All other columns will be dropped. Special characters will create errors and should be stripped outside of R. All labels will be truncated at 145 characters.

Details

If weights (`weightNm`) are provided, then it is normalized to have a sum of weights equal the total sample size, and the weights are used in all summary statistics calculations and plotting.

Value

A data.table object, formatted for use by all plotting functions in this package `otvPlots`, including the main function `vlm`, and the individual variable plotting function `PlotVar`.

License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Examples

```r
## Use the bankData dataset in this package
data(bankData)
bankData <- PrepData(bankData, dateNm = "date", dateGp = "months",
                      dateGpBp = "quarters")
## Columns have been assigned a plotting class (nmrcl/ctgr1)
str(bankData)
```

PrepLabels

Prepare variable labels
Usage
PrepLabels(labelF1, idx = 1:2)

Arguments

labelF1 Either the path of a dataset (a csv file) containing labels, an R object convertible
to data.table (e.g., data frame) or NULL. If NULL, no labels will be used. The
label dataset must contain at least 2 columns: varCol (variable names) and
labelCol (variable labels).

idx A vector of length 2, giving column index of variable names (first position) and
labels (second position).

Value
A data table formated for use by the vlm function.

License
Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance with the License. You may obtain a copy
of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law
or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See
the License for the specific language governing permissions and limitations under the License.

See Also
Functions depend on this function: PrintPlots, vlm.

Examples

```
data(bankLabels)
bankLabels <- PrepLabels(bankLabels)
```

Description
Create a pdf file with plots and compute summary statistics for all
variables

Usage
PrintPlots(outF1, dataF1, sortVars, dateNm, dateGp, dateGpBp, weightNm = NULL,
labelF1 = NULL, genCSV = TRUE, highlightNms = NULL, skewOpt = NULL,
ksample = 50000, fuzzyLabelFn = NULL, kCategories = 9)
Arguments

outfl Name of the output file, with no extension names (e.g., "bank"). A pdf file of plots ("bank.pdf"), and two csv files of summary statistics ("bank_categorical_summary.csv" and "bank_numerical_summary.csv") will be saved to your working directory, unless a path is included in outfl (e.g. ".plots/bank").

datafl A data.table containing at least the following columns: myVar, weightNm, dateGp, dateGbp; usually an output of the PrepData function.

sortVars A character vector of variable names in the order they will be plotted.

datennm Name of column containing the date variable.

dateGp Name of the variable that the time series plots should be grouped by. Options are NULL, "weeks", "months", "quarters", "years". See IDate for details. If NULL, then datennm will be used as dateGp.

dateGbp Name of variable the boxplots should be grouped by. Same options as dateGp. If NULL, then dateGp will be used.

weightNm Name of the variable containing row weights, or NULL for no weights (all rows receiving weight 1).

labelfl A data.table containing variable labels, or NULL for no labels; usually an output of PrepLabels.

genCSV Logical, whether to generate the two csv files of summary statistics for numerical and categorical variables.

highlightNms Either NULL or a character vector of variables to recieve red label. Currently NULL means all variables will get a black legend. Ignored this argument if labelfl == NULL.

skewOpt Either a numeric constant or NULL. Default is NULL (no transformation). If numeric, say 5, then all box plots of a variable whose skewness exceeds 5 will be on a log10 scale if possible. Negative input of skewOpt will be converted to 3.

ksample Either NULL or a positive integer. If an integer, indicates the sample size for both drawing boxplots and ordering numerical graphs by $R^2$. When the data is large, setting kSample to a reasonable value (default is 50K) dramatically improves processing speed. Therefore, for larger datasets (e.g. > 10 percent system memory), this parameter should not be set to NULL, or boxplots may take a very long time to render. This setting has no impact on the accuracy of time series plots on quantiles, mean, SD, and missing and zero rates.

fuzzyLabelFn Either NULL or a function of 2 parameters: A label file in the format of an output by PrepLabels and a string giving a variable name. The function should return the label corresponding to the variable given by the second parameter. This function should describe how fuzzy matching should be performed to find labels (see example below). If NULL, only exact matches will be returned.

kCategories If a categorical variable has more than kCategories, trace plots of only the kCategories most prevalent categories are plotted.

Value

A pdf of plots saved to file outfl.pdf, and if the argument genCSV == TRUE, also two csv files of summary statistics for numerical and categorical variables.
License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

See Also

Functions depend on this function: \texttt{vlm}.
This function depends on: \texttt{PlotVar, PrepData}.

\begin{verbatim}
SummaryStats
\end{verbatim}

Create summary statistics for a numerical variable

Description

Create summary statistics for a numerical variable

Usage

\begin{verbatim}
SummaryStats(myVar, dataFl, dateGp, weightNm = NULL)
\end{verbatim}

Arguments

\begin{verbatim}
myVar         The name of the variable to be plotted
dataFl         A data.table of data; must be the output of the PrepData function.
dateGp         Name of the variable that the time series plots should be grouped by. Options are NULL, "weeks", "months", "quarters", "years". See \texttt{IDate} for details. If NULL, then dateNm will be used as dateGp.
weightNm      Name of the variable containing row weights, or NULL for no weights (all rows receiving weight 1).
\end{verbatim}

Value

\begin{verbatim}
meltdx A data.table for use by the plotting funtions \texttt{PlotMean, PlotQuantiles, and PlotRates}.
numVarSummary A data.table of summary statistics.
\end{verbatim}

License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
Examples

```r
data(bankData)
bankData <- PrepData(bankData, dateNm = "date", dateGp = "quarters",
dateGpBp = "years")
mdx <- SummaryStats(myVar = "age", dataF1 = bankData,
dateGp = "quarters")$meltDx
plot(PlotQuantiles(mdx[variable %in% c("p99", "p50", "p1", "p99_g", "p50_g",
"p1_g"), "age", "quarters")])
plot(PlotMean(mdx[variable %in% c("mean", "c11", "c12")], "age", "quarters")])
plot(PlotRates(mdx, "age", "quarters")])
```

---

**vlm**

Create over time variable plots and summary statistics for variable level monitoring

---

**Description**

Sorts variables according to either user input or correlation with time (among numerical variables only), and create output files including:

- A PDF file of plots saved as outF1.pdf, with each indivual page on one variable. Variables are plotted in the order indicated in the argument `sortVars` or `sortFn`. For each numerical variable, the output plots include
  - side-by-side boxplots grouped by `dateGpBp` (left),
  - a trace plot of p1, p50, and p99 percentiles, grouped by `dateGp` (top right),
  - a trace plot of mean and ±1 SD control limits, grouped by `dateGp` (middle right), and
  - a trace plot of missing and zerorates, grouped by `dateGp` (bottom right).

For each categorical variable (including a numerical variable with no more than 2 unique levels not including NA), the output plots include

- a frequency bar plot (left), and
- a grid of trace plots on categories’ proportions over time (right). If the variable contains more than `k`Categories number of categories, trace plots of only the largest `k`Categories will be plotted. If the variable contains only two categories, then only the trace plot of the less prevalent categoy will be plotted.

- CSV file(s) on summary statistics of variable, both globally and over time aggregated by `dateGp`. The order of variables in the CSV files are the same as in the PDF file.
  - For numerical variables, number of observations (counts), p1, p25, p50, p75, and p99 qunatiles, mean, SD, missing and zerorates are saved as `outF1_numerical_summary.csv`.
  - For categorical variables, number of observations (counts) and categories’ proportions are saved as `outF1_categorical_summary.csv`. Each row is a category of a categorical (or binary) variable. The row whose category == "NA" corresponds to missing. Categories among the same variable are ordered by global prevalence in a descending order.
Usage

vlm(dataFl, dateNm, labelFl = NULL, outFl = "otvplots", genCSV = TRUE, dataNeedPrep = FALSE, dateGp = NULL, dateGpBp = NULL, weightNm = NULL, varNms = NULL, sortVars = NULL, sortFn = NULL, selectCols = NULL, dropCols = NULL, dateFt = "%d-%b-%Y", buildTm = NULL, highlightNms = NULL, skewOpt = NULL, kSample = 50000, fuzzyLabelFn = NULL, dropConstants = FALSE, kCategories = 9, ...)

Arguments

dataFl
Either the name of an object that can be converted using `as.data.table` (e.g., a data frame), or a character string containing the name of dataset that can be loaded using `fread` (e.g., a csv file). If the dataset is not in your working directory then dataFl must include (relative or absolute) path to file.

dateNm
Name of column containing the date variable.

labelFl
Either the path of a dataset (a csv file) containing labels, an R object convertible to `data.table` (e.g., data frame) or NULL. If NULL, no labels will be used. The label dataset must contain at least 2 columns: `varCol` (variable names) and `labelCol` (variable labels).

outFl
Name of the output file, with no extension names (e.g., "bank"). A pdf file of plots ("bank.pdf"), and two csv files of summary statistics ("bank_categorical_summary.csv" and "bank_numerical_summary.csv") will be saved to your working directory, unless a path is included in outFl (e.g. ".plots/bank").

genCSV
Logical, whether to generate the two csv files of summary statistics for numerical and categorical variables.

dataNeedPrep
Logical, indicates if data should be run through the PrepData function. This should be set to TRUE unless the PrepData function has been applied to the input data dataFl.

dateGp
Name of the variable that the time series plots should be grouped by. Options are NULL, "weeks", "months", "quarters", "years". See IDate for details. If NULL, then dateNm will be used as dateGp.

dateGpBp
Name of variable the boxplots should be grouped by. Same options as dateGp. If NULL, then dateGp will be used.

weightNm
Name of the variable containing row weights, or NULL for no weights (all rows receiving weight 1).

varNms
Either NULL or a vector of names or indices of variables to be plotted. If NULL, will default to all columns which are not dateNm or weightNm. Can also be a vector of indices of the column names, after dropCols or selectCols have been applied, if applicable, and not including dateGp, dateGpBp (which will be added to the dataFl by the function PrepData).

sortVars
Determines which variables to be plotted and their order. Either a character vector of variable names to plot variables in the same order as in the sortVars argument), or NULL to keep the original ordering, with numerical variables will being plotted before categorical and binary ones. sortVars should be NULL when the sortFn argument is used.
sortFn
A sorting function which returns sortVars as an output. The function may take the following variables as input: dataFl, dateNm, buildTm, weightNm, kSample. Currently, the only build-in sorting function is OrderByR, which sorts numerical variables in the order of strength of linear association with date, and adds categorical (and binary) variables sorted in alphabetical order after the numerical ones.

selectCols
Either NULL, or a vector of names or indices of variables to read into memory – must include dateNm, weightNm (if not NULL) and all variables to be plotted. If both selectCols and dropCols are NULL, then all variables will be read in.

dropCols
Either NULL, or a vector of variables names or indices of variables not to read into memory. If both selectCols and dropCols are NULL, then all variables will be read in.

dateFt
strptime format of date variable. The default is SAS format "%d%h%Y". But input data with R date format "%Y-%m-%d" will also be detected. Both of two formats can be parsed automatically.

buildTm
Vector identify time period for ranking/anomaly detection (most likely model build period). Allows for a subset of plotting time period to be used for anomaly detection.
- Must be a vector of dates and must be inclusive i.e. buildTm[1] <= date <= buildTm[2] will define the time period.
- Must be either NULL, a vector of length 2, or a vector of length 3.
- If NULL, the entire dataset will be used for ranking/anomaly detection.
- If a vector of length 2, the format of the dates must be a character vector in default R date format (e.g. "2017-01-30").
- If a vector of length 3, the first two columns must contain dates in any strptime format, while the 3rd column contains the strptime format (see strptime).
- The following are equivalent ways of selecting all of 2014:
  - c("2014-01-01", "2014-12-31")
  - c("01JAN2014", "31DEC2014", "%d%h%Y")

highlightNms
Either NULL or a character vector of variables to recieve red label. Currently NULL means all variables will get a black legend. Ignored this argument if labelFl == NULL.

skewOpt
Either a numeric constant or NULL. Default is NULL (no transformation). If numeric, say 5, then all box plots of a variable whose skewness exceeds 5 will be on a log10 scale if possible. Negative input of skewOpt will be converted to 3.

kSample
Either NULL or a positive integer. If an integer, indicates the sample size for both drawing boxplots and ordering numerical graphs by $R^2$. When the data is large, setting kSample to a reasonable value (default is 50K) dramatically improves processing speed. Therefore, for larger datasets (e.g. > 10 percent system memory), this parameter should not be set to NULL, or boxplots may take a very long time to render. This setting has no impact on the accuracy of time series plots on quantiles, mean, SD, and missing and zero rates.

fuzzyLabelFn
Either NULL or a function of 2 parameters: A label file in the format of an output by PrepLabels and a string giving a variable name. The function should return
the label corresponding to the variable given by the second parameter. This
function should describe how fuzzy matching should be performed to find labels
(see example below). If NULL, only exact matches will be returned.

*dropConstants* Logical, indicates whether or not constant (all duplicated or NA) variables should
be dropped from dataF1 prior to plotting.

*kCategories* If a categorical variable has more than kCategories, trace plots of only the
kCategories most prevalent categories are plotted.

... Additional parameters to be passed to *fread*.

Details

If the argument *dataNeedPrep* is set to FALSE, then

- *dataF1* must be a *data.table* containing variables *weightNm*, *dateNm*, *dateGp*, and *dateGpBp*,
  and names of these variables must be the same as the corresponding arguments of the *vlm*
  function.
- the arguments *selectCols*, *dropCols*, *dateFt*, *dropConstants* will be ignored by the *vlm*
  function.
- When analyzing a dataset for the first time, it is recommended to first run the *PrepData* func-
  tion on it, and then apply the *vlm* function with the argument *dataNeedPrep = FALSE*. Please
  see the examples for details.

License

Copyright 2017 Capital One Services, LLC Licensed under the Apache License, Version 2.0 (the
"License"); you may not use this file except in compliance with the License. You may obtain a copy
of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law
or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See
the License for the specific language governing permissions and limitations under the License.

See Also

This function depends on: *PrintPlots, OrderByR2, PrepData, PrepLabels*.

Examples

```R
## Load the data and its label
data(bankData)
data(bankLabels)

## The PrepData function should only need to be run once on a dataset,
## after that vlm can be run with the argument dataNeedPrep = FALSE
bankData <- PrepData(bankData, dateNm = "date", dateGp = "months",
                     dateGpBp = "quarters")
bankLabels <- PrepLabels(bankLabels)

## Not run:
vlm(dataF1 = bankData, dateNm = "date", labelF1 = bankLabels,
```
sortFn = "OrderByR2", dateGp = "months", dateGpBp = "quarters",
outFl = "bank")

## If csv files of summary statistics are not need, set genCSV = FALSE
vlm(dataFl = bankData, dateNm = "date", labelFl = bankLabels, genCSV = FALSE,
sortFn = "OrderByR2", dateGp = "months", dateGpBp = "quarters",
outFl = "bank")

## If weights are provided, they will be used in all statistical calculations
bankData[, weight := rnorm(Nn, 1, .1)]
vlm(dataFl = bankData, dateNm = "date", labelFl = bankLabels,
    dateGp = "months", dateGpBp = "quarters", weightNm = "weight",
    outFl = "bank")

## Customize plotting order by passing a vector of variable names to
## sortVars, but the "date" column must be excluded from sortVars
sortVars <- sort(bankLabels[,varCol="date",varCol])
vlm(dataFl = bankData, dateNm = "date", labelFl = bankLabels,
    dateGp = "months", dateGpBp = "quarters", outFl = "bank",
    sortVars = sortVars)

## Create plots for a specific variable using the varNms parameter
vlm(dataFl = bankData, dateNm = "date", labelFl = bankLabels,
    dateGp = "months", dateGpBp = "quarters", outFl = "bank",
    varNms = "age", sortVars = NULL)

## End(Not run)
Index

*Topic datasets
  bankData, 2
  bankLabels, 3

as.data.table, 19, 25

bankData, 2, 3
bankLabels, 3

CalcR2, 4, 6, 20

fread, 19, 20, 25, 27

IDate, 9, 11, 12, 14, 15, 17, 19, 22, 23, 25

lm.fit, 4
lm.wfit, 4

OrderByR2, 4, 5, 7, 20, 26, 27
otvPlots, 3, 6, 20
otvPlots-package (otvPlots), 6

PlotBarplot, 7, 9, 20
PlotCatVar, 8, 8, 16, 18, 20
PlotDist, 10, 13, 20
PlotMean, 11, 13, 20, 23
PlotNumVar, 12, 18, 20
PlotQuantiles, 13, 13, 20, 23
PlotRates, 13, 14, 20, 23
PlotRatesOverTime, 9, 15, 20
PlotVar, 9, 13, 16, 20, 23
PrepData, 4–10, 12, 13, 15–19, 19, 22, 23, 25, 27
PrepLabels, 3, 7, 17, 18, 20, 22, 26, 27
PrintPlots, 9, 18, 20, 21, 21, 27

strptime, 5, 19, 26

SummaryStats, 11, 13, 14, 20, 23

vlm, 6, 7, 9, 19–21, 23, 24, 27