Package ‘pltesim’

June 26, 2017

Type Package
Title Simulate Probabilistic Long-Term Effects in Models with Temporal Dependence
Description Calculates and depicts probabilistic long-term effects in binary models with temporal dependence variables. The package performs two tasks. First, it calculates the change in the probability of the event occurring given a change in a theoretical variable. Second, it calculates the rolling difference in the future probability of the event for two scenarios: one where the event occurred at a given time and one where the event does not occur. The package is consistent with the recent movement to depict meaningful and easy-to-interpret quantities of interest with the requisite measures of uncertainty. It is the first to make it easy for researchers to interpret short- and long-term effects of explanatory variables in binary autoregressive models, which can have important implications for the correct interpretation of these models.

Version 1.0
Date 2017-06-26

BugReports https://github.com/christophergandrud/pltesim/issues

URL https://CRAN.R-project.org/package=pltesim
License GPL (>= 3)
LazyData TRUE
RoxygenNote 6.0.1

Imports coreSim, ggplot2
Suggests testthat, covr

NeedsCompilation no

Author Christopher Gandrud [aut, cre], Laron K. Williams [aut]

Maintainer Christopher Gandrud <christopher.gandrud@gmail.com>

Repository CRAN

Date/Publication 2017-06-26 12:48:16 UTC
R topics documented:

btscs ................................................................. 2
logistic_prob_FUN .................................................. 3
negative ............................................................. 3
negative_year ....................................................... 4
plte_builder .......................................................... 4
plte_plot ............................................................. 6

Index

btscs ................................................................. 8

---

btscs

Generate spells for binary variables

Description

Generate spells for binary variables

Usage

btscs(df, event, t_var, cs_unit, pad_ts = FALSE)

Arguments

df a data frame
event character string naming the binary variable identifying the event. Note 1 must
signify events and 0 non-events.
t_var character string with the name of the time variable.
cs_unit character string with the name of the cross-sectional unit.
pad_ts logical indicating whether or not to fill in the time-series if panels are unbal-
nanced.

Value

The original (df) data frame with an additional spell_time value identifying the number of ob-
served periods in the spell, i.e. time points since the last period.

Source

This function is a port of Dave Armstrong’s btscs function from:
It was ported largely to reduce the dependencies needed for the examples. There are also internal
improvements, largely to handle single period spells and to start the spell time counter from 1.
David’s package implemented the Stata function from:
Examples

```r
data('negative')

neg_set <- btscs(df = negative, event = 'y', t_var = 'tim', cs_unit = 'group')
```

---

**logistic_prob_FUN**  
*Function for finding predicted probability from the systematic linear component of a logistic regression.*

---

**Description**

Function for finding predicted probability from the systematic linear component of a logistic regression.

**Usage**

```r
logistic_prob_FUN(x)
```

**Arguments**

- `x`  
a numeric vector of the systematic linear component from a logistic regression model.

---

**negative**  
*Simulated data set from Williams (2016) to illustrate negative duration dependence*

---

**Description**

Simulated data set from Williams (2016) to illustrate negative duration dependence

**Format**

A data set with 1000 observations and 6 variables

**Source**

negative_year

Simulated data set based on Williams (2016) to illustrate negative duration dependence in examples

Description
Simulated data set based on Williams (2016) to illustrate negative duration dependence in examples

Format
A data set with 1000 observations and 4 variables

plte_builder

Create simulations for long-term effects in models with temporal dependence

Description
Create simulations for long-term effects in models with temporal dependence

Usage
plte_builder(obj, obj_tvar, cf, cf_duration = "permanent", t_points, FUN = logistic_prob_FUN, ci = 0.95, nsim = 1000)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>obj</td>
<td>a fitted model object.</td>
</tr>
<tr>
<td>obj_tvar</td>
<td>character string specifying the name of the base time variable in obj.</td>
</tr>
<tr>
<td>cf</td>
<td>a data frame with the first row containing the counterfactual. An optional</td>
</tr>
<tr>
<td></td>
<td>second row could be supplied with values for the baseline scenario. If not</td>
</tr>
<tr>
<td></td>
<td>supplied then all values are set to zero for the baseline. Columns should</td>
</tr>
<tr>
<td></td>
<td>have names that match variables in obj and contain fitted values to find</td>
</tr>
<tr>
<td></td>
<td>quantities of interest for. Note, this should not include your time variable</td>
</tr>
<tr>
<td></td>
<td>as this is handled using t_points.</td>
</tr>
<tr>
<td>cf_duration</td>
<td>a character string or numeric specifying the counterfactual’s duration. If</td>
</tr>
<tr>
<td></td>
<td>'permanent' then the counterfactual lasts for the full time span in t_points.</td>
</tr>
<tr>
<td></td>
<td>If 'one-time' then the counterfactual only lasts for one period. If cf_duration</td>
</tr>
<tr>
<td></td>
<td>is numeric then the number specifies the number of time increments at which</td>
</tr>
<tr>
<td></td>
<td>the counterfactual resets.</td>
</tr>
<tr>
<td>t_points</td>
<td>a numeric vector with a minimum length of 2 and a maximum length of 3. The</td>
</tr>
<tr>
<td></td>
<td>first and last values should be the time starting and ending points for the</td>
</tr>
<tr>
<td></td>
<td>simulations. The (optional) middle value can specify a point between the first</td>
</tr>
<tr>
<td></td>
<td>and last time points where a subsequent event occurs.</td>
</tr>
</tbody>
</table>
**Fun**
a function for finding a quantity of interest from the linear systematic component. See **qi_builder**. The default is a function for finding the predicted probability from a logistic regression model.

**ci**
the proportion of the central interval of the simulations to return. Must be in (0, 1] or equivalently (0, 100].

**nsim**
number of simulations to draw.

**Value**
A data frame with the medians and central intervals of the simulated scenarios. Note that the column scenario_name encodes scenarios where \( y = 0 \) as baseline and \( y = 1 \) as counterfactual.

**Source**

**Examples**
```r
data('negative')

# BTSCS set the data
neg_set <- btscs(df = negative, event = 'y', t_var = 'tim',
                 cs_unit = 'group', pad_ts = FALSE)

# Create temporal dependence variable
neg_set$t <- neg_set$spell + 1

m1 <- glm(y ~ x + t + I(t^2) + I(t^3),
           family = binomial(link = 'logit'), data = neg_set)

# Create fitted counterfactual
counterfactual <- data.frame(x = 0.5)

# Permanent counterfactual, one event
sim1 <- plte_builder(obj = m1, obj_tvar = 't',
                     cf = counterfactual, t_points = c(13, 25))

# Multiple events
sim2 <- plte_builder(obj = m1, obj_tvar = 't',
                     cf = counterfactual, t_points = c(13, 18, 25))

# One-time counterfactual
sim3 <- plte_builder(obj = m1, obj_tvar = 't',
                     cf = counterfactual, t_points = c(13, 25),
                     cf_duration = 'one-time')

# Temporary (4 period counterfactual)
sim4 <- plte_builder(obj = m1, obj_tvar = 't',
                     cf = counterfactual, t_points = c(13, 25),
                     cf_duration = 4)
```
# Custom baseline scenario
# Note: the second row is the custom baseline
counterfactual_baseline <- data.frame(x = c(1, 0.5))

sim5 <- plte_builder(obj = m1, obj_tvar = 't', cf_duration = 4,
                     cf = counterfactual_baseline, t_points = c(13, 25))

# Time splines
library(splines)
m2 <- glm(y ~ x + bs(t, degree = 3), family = binomial(link = 'logit'),
         data = neg_set)
sim6 <- plte_builder(obj = m2, obj_tvar = 't', cf_duration = 4,
                     cf = counterfactual, t_points = c(13, 25))

---

**.tplt.plot**  
*Plot objects created by *plte_builder*

**Description**

Plot objects created by plte_builder

**Usage**

```r
plte_plot(obj, t_labels = TRUE)
```

**Arguments**

- **obj**  
a plte class object created by `plte_builder`.
- **t_labels**  
logical whether or not to include time labels for each point.

**Value**

A ggplot2 object that can be modified using the + in combination with other ggplot2 functions.

**Examples**

data('negative')

# BTSCS set the data
neg_set <- btscs(df = negative, event = 'y', t_var = 'tim',
                 cs_unit = 'group', pad_ts = FALSE)

# Create temporal dependence variables
def_set$t <- def_set$spell + 1

m1 <- glm(y ~ x + t + I(t^2) + I(t^3),
          family = binomial(link = 'logit'),
          data = neg_set)

m2 <- glm(y ~ x + bs(t, degree = 3), family = binomial(link = 'logit'),
          data = neg_set)

sim5 <- plte_builder(obj = m1, obj_tvar = 't', cf_duration = 4,
                      cf = counterfactual, t_points = c(13, 25))

sim6 <- plte_builder(obj = m2, obj_tvar = 't', cf_duration = 4,
                      cf = counterfactual, t_points = c(13, 25))

---

**plte.plot**  
*Plot objects created by *plte_builder*

**Description**

Plot objects created by plte_builder

**Usage**

```r
plte_plot(obj, t_labels = TRUE)
```

**Arguments**

- **obj**  
a plte class object created by `plte_builder`.
- **t_labels**  
logical whether or not to include time labels for each point.

**Value**

A ggplot2 object that can be modified using the + in combination with other ggplot2 functions.

**Examples**

data('negative')

# BTSCS set the data
neg_set <- btscs(df = negative, event = 'y', t_var = 'tim',
                 cs_unit = 'group', pad_ts = FALSE)

# Create temporal dependence variables
def_set$t <- def_set$spell + 1

m1 <- glm(y ~ x + t + I(t^2) + I(t^3),
          family = binomial(link = 'logit'),
          data = neg_set)

m2 <- glm(y ~ x + bs(t, degree = 3), family = binomial(link = 'logit'),
          data = neg_set)

sim5 <- plte_builder(obj = m1, obj_tvar = 't', cf_duration = 4,
                      cf = counterfactual, t_points = c(13, 25))

sim6 <- plte_builder(obj = m2, obj_tvar = 't', cf_duration = 4,
                      cf = counterfactual, t_points = c(13, 25))
data = neg_set)

counterfactual <- data.frame(x = 0.5)

sim1 <- plte_builder(obj = m1, obj_tvar = 't',
                      cf = counterfactual, t_points = c(13, 25),
                      cf_duration = 4, ci = 99)

# With time point labels
plte_plot(sim1)

# Without time point labels
plte_plot(sim1, t_labels = FALSE)
Index

*Topic datasets
  negative, 3
  negative_year, 4

btscs, 2

logistic_prob_FUN, 3

negative, 3
negative_year, 4

plte_builder, 4, 6
plte_plot, 6
qi_builder, 5