Package 'dynsim'

June 19, 2025

Title Dynamic Simulations of Autoregressive Relationships

Version 1.2.4

Date 2025-06-09

URL https://cran.r-project.org/package=dynsim

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

Description Dynamic simulations and graphical depictions of autoregressive relationships.

License GPL-3

Depends R (>= 3.0.0)

Imports ggplot2 (>= 1.0.1.9003), grid, gridExtra (>= 2.0.0), MASS

Encoding UTF-8

BuildVignettes true

LazyData true

RoxygenNote 7.1.1

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2025-06-19 12:40:06 UTC

Contents

dynsim *Dynamic simulations of autoregressive relationships*

Description

dynsim dynamic simulations of autoregressive relationships

Usage

```
dynsim(obj, ldv, scen, n = 10, sig = 0.95, num = 1000, shocks = NULL, ...)
```

Arguments

obj	the output object the estimation model.
ldv	character. Names the lagged dependent variable
scen	data frame or list of data frames. Specifies the values of the variables used to generate the predicted values when $t = 0$. If only one scenario is desired then scen should be a data frame. If more than one scenario is desired then the $t = 0$ values should be in data frames contained in a list.
n	numeric. Specifies the number of iterations (or time period) over which the program will generate the predicted value of the dependent variable. The default is 10.
sig	numeric. Specifies the level of statistical significance of the confidence intervals. Any value allowed be greater than 0 and cannot be greater than 1.
num	numeric. Specifies the number of simulations to compute for each value of n. The default is 1000.
shocks	data frame. Allows the user to choose independent variables, their values, and times to introduce these values. The first column of the data frame must be called times this will contain the times in n to use the shock values. The following columns' names must match the names of the variables whose values you wish to alter. You do not need to specify values for variables that you want to remain the same as in scen. In times n where shock values are not specified, non-ldv variable values will revert to those in scen. If * is used to create interactions, interaction terms will be fitted appropriately.
...	arguments to pass to methods.

Details

A post-estimation technique for producing dynamic simulations of autoregressive models.

Value

The command returns a data.frame and dynsim class object. This can contain up to columns elements:

- scenNumber: The scenario number.

- `time`: The time points.
- `shock.`: Columns containing the values of the shock variables at each point in `time`.
- `ldvMean`: Mean of the simulation distribution.
- `ldvLower`: Lower bound of the simulation distribution's central interval set with `sig`.
- `ldvUpper`: Upper bound of the simulation distribution's central interval set with `sig`.
- `ldvLower50`: Lower bound of the simulation distribution's central 50 percent interval.
- `ldvUpper50`: Upper bound of the simulation distribution's central 50 percent interval.

The output object is a data frame class object. Do with it as you like.

References

Williams, L. K., & Whitten, G. D. (2011). Dynamic Simulations of Autoregressive Relationships. The Stata Journal, 11(4), 577-588.

Williams, L. K., & Whitten, G. D. (2012). But Wait, There's More! Maximizing Substantive Inferences from TSCS Models. Journal of Politics, 74(03), 685-693.

Examples

```
# Load Grunfeld data
data(grunfeld, package = "dynsim")

# Create lag invest variable
grunfeld <- grunfeld[order(grunfeld$company, grunfeld$year), ]
grunfeld$InvestLag <- ave(grunfeld$invest, grunfeld$company,
                          FUN = function(x) c(NA, x[-length(x)]))

# Convert company to factor for fixed-effects specification
grunfeld$company <- as.factor(grunfeld$company)

# Estimate basic model
M1 <- lm(invest ~ InvestLag + mvalue + kstock + company, data = grunfeld)

# Estimate model with interaction between mvalue and kstock
M2 <- lm(invest ~ InvestLag + mvalue*kstock + company, data = grunfeld)

# Set up scenarios for company 4
## List version ##
attach(grunfeld)
Scen1 <- data.frame(InvestLag = mean(InvestLag, na.rm = TRUE),
                    mvalue = quantile(mvalue, 0.05),
                    kstock = quantile(kstock, 0.05),
                    company4 = 1)
Scen2 <- data.frame(InvestLag = mean(InvestLag, na.rm = TRUE),
                    mvalue = mean(mvalue),
                    kstock = mean(kstock),
                    company4 = 1)
Scen3 <- data.frame(InvestLag = mean(InvestLag, na.rm = TRUE),
                    mvalue = quantile(mvalue, 0.95),
                    kstock = quantile(kstock, 0.95),
```

```
                    company4 = 1)
detach(grunfeld)

## Not run:
## Alternative data frame version of the scenario builder ##
attach(grunfeld)
ScenComb <- data.frame(InvestLag = rep(mean(InvestLag, na.rm = TRUE), 3),
                       mvalue = c(quantile(mvalue, 0.95), mean(mvalue),
                                  quantile(mvalue, 0.05)),
                       kstock = c(quantile(kstock, 0.95), mean(kstock),
                                  quantile(kstock, 0.05)),
                       company4 = rep(1, 3)
)
detach(grunfeld)

## End(Not run)

# Combine into a single list
ScenComb <- list(Scen1, Scen2, Scen3)

## Run dynamic simulations without shocks and no interactions
Sim1 <- dynsim(obj = M1, ldv = "InvestLag", scen = ScenComb, n = 20)

## Run dynamic simulations without shocks and interactions
Sim2 <- dynsim(obj = M2, ldv = "InvestLag", scen = ScenComb, n = 20)

## Run dynamic simulations with shocks

# Create data frame of shock values
mShocks <- data.frame(times = c(5, 10), kstock = c(100, 1000),
                      mvalue = c(58, 5000))

# Run simulations without interactions
Sim3 <- dynsim(obj = M1, ldv = "InvestLag", scen = ScenComb, n = 20,
               shocks = mShocks)

# Run simulations with interactions
Sim4 <- dynsim(obj = M2, ldv = "InvestLag", scen = ScenComb, n = 20,
               shocks = mShocks)
```

---

dynsimGG *Plot dynamic simulation results from* dynsim

Description

dynsimGG uses ggplot2 to plot dynamic simulation results created by dynsim.

Usage

```
dynsimGG(
  obj,
  lsize = 1,
  color,
  alpha = 0.5,
  xlab = "\nTime",
  ylab = "Predicted Value\n",
  title = "",
  leg.name = "Scenario",
  leg.labels,
  legend = "legend",
  shockplot.var,
  shockplot.ylab,
  shockplot.heights = c(12, 4),
  shockplot.heights.units = c("cm", "cm")
)
```

Arguments

`obj`	a `dynsim` class object.
`lsize`	size of the smoothing line. Default is 1. See ggplot2.
`color`	character string. Specifies the color of the lines and ribbons. If only one scenario is to be plotted then it can either be a single color value using any color value allowed by ggplot2. The default is the hexadecimal color `"#2B8CBE"`. If more than one scenario is to be plotted then a color brewer palette is set. The default is`"Set1"`. See `scale_colour_brewer`.
`alpha`	numeric. Alpha (e.g. transparency) for the ribbons. Default is `alpha = 0.1`. See ggplot2.
`xlab`	a label for the plot's x-axis.
`ylab`	a label of the plot's y-axis.
`title`	the plot's main title.
`leg.name`	name of the legend (if applicable).
`leg.labels`	character vector specifying the labels for each scenario in the legend.
`legend`	specifies what type of legend to include (if applicable). The default is `legend = "legend"`. To hide the legend use `legend = FALSE`. See `discrete_scale` for more details.
`shockplot.var`	character string naming the one shock variable to plot fitted values of over time specified underneath the main plot.
`shockplot.ylab`	character string for the shockplot's y-axis label.
`shockplot.heights`	numeric vector with of length 2 with units of the main and shockplot height plots.
`shockplot.heights.units`	a character vector of length 2 with the unit types for the values in `shockplot.heights`. See `unit` for details.

Details

Plots dynamic simulations of autoregressive relationships from dynsim. The central line is the mean of the simulation distributions. The outer ribbon is the furthest extent of the simulation distributions' central intervals found in dynsim with the sig argument. The middle ribbons plot the limits of the simulation distributions' central 50

Examples

```
# Load Grunfeld data
data(grunfeld, package = "dynsim")

# Create lag invest variable
grunfeld <- grunfeld[order(grunfeld$company, grunfeld$year), ]
grunfeld$InvestLag <- ave(grunfeld$invest, grunfeld$company,
                          FUN = function(x) c(NA, x[-length(x)]))

# Convert company to factor for fixed-effects specification
grunfeld$company <- as.factor(grunfeld$company)

# Estimate basic model
M1 <- lm(invest ~ InvestLag + mvalue + kstock + company, data = grunfeld)

# Set up scenarios for company 4
attach(grunfeld)
Scen1 <- data.frame(InvestLag = mean(InvestLag, na.rm = TRUE),
                    mvalue = quantile(mvalue, 0.05),
                    kstock = quantile(kstock, 0.05),
                    company4 = 1)
Scen2 <- data.frame(InvestLag = mean(InvestLag, na.rm = TRUE),
                    mvalue = mean(mvalue),
                    kstock = mean(kstock),
                    company4 = 1)
Scen3 <- data.frame(InvestLag = mean(InvestLag, na.rm = TRUE),
                    mvalue = quantile(mvalue, 0.95),
                    kstock = quantile(kstock, 0.95),
                    company4 = 1)
detach(grunfeld)

# Combine into a single list
ScenComb <- list(Scen1, Scen2, Scen3)

## Run dynamic simulations without shocks
Sim1 <- dynsim(obj = M1, ldv = "InvestLag", scen = ScenComb, n = 20)

# Create plot legend label
Labels <- c("5th Percentile", "Mean", "95th Percentile")

# Plot
dynsimGG(Sim1, leg.labels = Labels)

## Run dynamic simulations with shocks
```

```
# Create data frame of shock values
mShocks <- data.frame(times = c(5, 10), kstock = c(100, 1000))

# Run simulations
Sim2 <- dynsim(obj = M1, ldv = "InvestLag", scen = ScenComb, n = 20,
               shocks = mShocks)

# Plot
dynsimGG(Sim2, leg.labels = Labels)

# Plot with accompanying shock plot
dynsimGG(Sim2, leg.labels = Labels, shockplot.var = "kstock")
```

---

grunfeld *A data set from Grunfeld (1958)*

Description

A data set from Grunfeld (1958)

Format

A data set with 200 observations and 6 variables

Source

Grunfeld, Yehuda. 1958. The Determinants of Corporate Investment. PhD thesis. University of Chicago.

Index