

# Package ‘DynNom’

April 19, 2017

**Type** Package

**Title** Dynamic Nomograms for Linear, Generalized Linear and Proportional Hazard Models

**Version** 4.1.1

**Author** Amirhossein Jalali, Davood Roshan, Alberto Alvarez-Iglesias, John Newell

**Maintainer** Amirhossein Jalali <a.jalali2@nuigalway.ie>

**Description** Demonstrate the results of a statistical model object as a dynamic nomogram in an RStudio panel or web browser. Also, the generic DNbuilder() function in this package provides a simple and straightforward way to build and publish a dynamic nomogram on the web to use the app independent of R. 'DynNom' supports a variety of model objects; lm(), glm(), coxph() models and also ols(), Glm(), lrm(), cph() models in the 'rms' package.

**License** GPL-2

**LazyData** TRUE

**Depends** survival (>= 2.38-3), rms, plotly

**Imports** shiny, ggplot2 (> 2.1.0), stargazer, compare, BBmisc

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-04-19 08:59:10 UTC

## R topics documented:

DNbuilder . . . . .	2
DNbuilder.coxph . . . . .	3
DNbuilder.glm . . . . .	5
DNbuilder.lm . . . . .	6
DynNom . . . . .	8
DynNom.coxph . . . . .	10
DynNom.cph . . . . .	12
DynNom.Glm . . . . .	13
DynNom.glm . . . . .	15
DynNom.lm . . . . .	16
DynNom.lrm . . . . .	18
DynNom.ols . . . . .	19

---

DNbuilder	<i>Shiny code for Dynamic Nomograms</i>
-----------	---

---

### Description

DNbuilder is a generic function to build dynamic nomograms and provide the required scripts for deploying them on a server on the web such as the <http://shinyapps.io>. DNbuilder supports `lm`, `glm`, `coxph` model objects.

### Usage

```
DNbuilder(model, data, clevel = 0.95, m.summary = c("raw", "formatted"),
          covariate = c("slider", "numeric"), ptype = c("st", "1-st"))
```

### Arguments

<code>model</code>	an <code>lm</code> , <code>glm</code> or <code>coxph</code> model object
<code>data</code>	dataframe containing the accompanying data
<code>clevel</code>	confidence level required
<code>m.summary</code>	The option to choose the format of the model output in the 'Summary Model' tab. If "raw" (the default) is chosen the result of <code>summary(model)</code> will be display while if "formatted" is chosen the model summary using the <code>stargazer</code> package will be displayed.
<code>covariate</code>	The option to choose the type of covariate(s) input control widget for numeric values. If "slider" (the default) is chosen a shiny application with slider control widgets are used while if "numeric" is chosen numeric values input controls will be displayed.
<code>ptype</code>	This plot type option relates to <code>coxph</code> objects only. If "st" (the default) is chosen, a plot of the estimated survivor function, $S(t)$ , is displayed. If "1-st" is chosen a plot of $1 - S(t)$ is displayed.

### Value

A new folder in the current working directory called `DynNomapp` which contains all the required scripts to deploy this dynamic nomogram on a server on the web such as the <http://shinyapps.io>. This folder includes `ui.R`, `server.R` and `global.R` script files needed to build the application and `dataset.rds` which is the accompanying dataset and a user guide text file called `README.txt` which explains how to deploy the app using all these objects.

### Author(s)

Amirhossein Jalali, Alberto Alvarez-Iglesias, John Newell

## References

Banks, J. 2006. Nomograms. Encyclopedia of Statistical Sciences. 8.  
Easy web applications in R. <http://shiny.rstudio.com>

## See Also

[lm](#), [glm](#), [coxph](#), [DynNom](#), [DynNom.lm](#), [DynNom.glm](#), [DynNom.coxph](#)

## Examples

```
## Not run:
# simple linear regression models
model1 <- lm(uptake ~ Plant + conc + Plant * conc, data = CO2)
DNbuilder(model1, CO2)

# Generalized regression models
data1 =as.data.frame(Titanic)
model2 <- glm(Survived ~ Age + Class + Sex, data = data1, weights = Freq,
              family = binomial("probit"))
DNbuilder(model2, data1, clevel = 0.9)

# a proportional hazard model
data.kidney <- kidney
# always make sure that the categorical variables are in a factor class
data.kidney$sex <- as.factor(data.kidney$sex)
levels(data.kidney$sex) <- c("male", "female")

model3 <- coxph(Surv(time, status) ~ age + sex + disease, data.kidney)
DNbuilder(model3, data.kidney)
DNbuilder(model3, data.kidney, ptype = "1-st")

## End(Not run)

if (interactive()) {
# a poisson regression model
model4 <- glm(event ~ mag + station + dist + accel, data = attenu, family = poisson)
DynNom(model4, attenu, covariate = "numeric")
}
```

## Description

DNbuilder.coxph provides required scripts to deploy an lm model object as a dynamic nomogram on a server on the web such as the <http://shinyapps.io>.

**Usage**

```
DNbuilder.coxph(model, data, clevel = 0.95, m.summary = c("raw", "formatted"),
  covariate = c("slider", "numeric"), ptype = c("st", "1-st"))
```

**Arguments**

model	a coxph model object
data	dataframe containing the accompanying data
clevel	confidence level required
m.summary	The option to choose the format of the model output in the 'Summary Model' tab. If "raw" (the default) is chosen the result of summary(model) will be display while if "formatted" is chosen the model summary using the stargazer package will be displayed.
covariate	The option to choose the type of covariate(s) input control widget for numeric values. If "slider" (the default) is chosen a shiny application with slider control widgets are used while if "numeric" is chosen numeric values input controls will be displayed.
ptype	If "st" (the default) is chosen, a plot of the estimated survivor function, $S(t)$ , is displayed. If "1-st" is chosen a plot of $1 - S(t)$ is displayed.

**Value**

A new folder in the current working directory called DynNomapp which contains all the required scripts to deploy this dynamic nomogram on a server on the web such as the <http://shinyapps.io>. This folder includes ui.R, server.R and global.R script files needed to build the application and dataset.rds which is the accompanying dataset and a user guide text file called README.txt which explains how to deploy the app using all these objects.

**Author(s)**

Amirhossein Jalali, Alberto Alvarez-Iglesias, John Newell

**See Also**

[coxph](#), [DynNom](#), [DynNom.coxph](#)

**Examples**

```
## Not run:
data.kidney <- kidney
# always make sure that the categorical variables are in a factor class
data.kidney$sex <- as.factor(data.kidney$sex)
levels(data.kidney$sex) <- c("male", "female")

model1 <- coxph(Surv(time, status) ~ age + sex + disease, data.kidney)
DNbuilder(model1, data.kidney)
DNbuilder(model1, data.kidney, ptype = "1-st")
```

```

# a cox model including a strata term
data(lung)
model2 <- coxph(Surv(time, status) ~ age + strata(sex) + ph.ecog , data = lung)
DNbuilder(model2, lung)

## End(Not run)

if (interactive()) {
  data.ovary <- ovarian
  data.ovary$resid.ds <- as.factor(data.ovary$resid.ds)
  levels(data.ovary$resid.ds) <- c("no", "yes")
  data.ovary$rx <- as.factor(data.ovary$rx)
  data.ovary$ecog.ps <- as.factor(data.ovary$ecog.ps)

  model3 <- coxph(Surv(futime, fustat) ~ age + resid.ds * rx + ecog.ps, data = data.ovary)
  DNbuilder.coxph(model3, data.ovary)
}

```

---

DNbuilder.glm

*Shiny code for Dynamic Nomograms*


---

## Description

DNbuilder.glm provides required scripts to deploy an lm model object as a dynamic nomogram on a server on the web such as the <http://shinyapps.io>.

## Usage

```
DNbuilder.glm(model, data, clevel = 0.95, m.summary = c("raw", "formatted"),
              covariate = c("slider", "numeric"))
```

## Arguments

model	a glm model object
data	dataframe containing the accompanying data
clevel	confidence level required
m.summary	The option to choose the format of the model output in the 'Summary Model' tab. If "raw" (the default) is chosen the result of summary(model) will be display while if "formatted" is chosen the model summary using the stargazer package will be displayed.
covariate	The option to choose the type of covariate(s) input control widget for numeric values. If "slider" (the default) is chosen a shiny application with slider control widgets are used while if "numeric" is chosen numeric values input controls will be displayed.

**Value**

A new folder in the current working directory called DynNomapp which contains all the required scripts to deploy this dynamic nomogram on a server on the web such as the <http://shinyapps.io>. This folder includes ui.R, server.R and global.R script files needed to build the application and dataset.rds which is the accompanying dataset and a user guide text file called README.txt which explains how to deploy the app using all these objects.

**Author(s)**

Amirhossein Jalali, Alberto Alvarez-Iglesias, John Newell

**See Also**

[glm](#), [DynNom](#), [DynNom.glm](#)

**Examples**

```
## Not run:
# a generalised linear model
model1 <- glm(Fertility ~ Agriculture + Education + Catholic, data = swiss)
DNbuilder.glm(model1, swiss, clevel = 0.9)

# a logistic regression model
data1 = as.data.frame(Titanic)
model2 <- glm(Survived ~ Age + Class + Sex, data = data1, weights = Freq,
              family = binomial("probit"))
DNbuilder(model2, as.data.frame(Titanic), clevel = 0.9)

## End(Not run)

if (interactive()) {
# a poisson regression model
model3 <- glm(event ~ mag + dist + accel, data = attenu, family = poisson)
DNbuilder(model3, attenu, covariate = "numeric")
}
```

---

DNbuilder.lm

*Shiny code for Dynamic Nomograms*


---

**Description**

DNbuilder.lm provides required scripts to deploy an lm model object as a dynamic nomogram on a server on the web such as the <http://shinyapps.io>.

**Usage**

```
DNbuilder.lm(model, data, clevel = 0.95, m.summary = c("raw", "formatted"),
             covariate = c("slider", "numeric"))
```

## Arguments

model	an lm model object
data	dataframe containing the accompanying data
clevel	confidence level required
m.summary	The option to choose the format of the model output in the 'Summary Model' tab. If "raw" (the default) is chosen the result of summary(model) will be display while if "formatted" is chosen the model summary using the stargazer package will be displayed.
covariate	The option to choose the type of covariate(s) input control widget for numeric values. If "slider" (the default) is chosen a shiny application with slider control widgets are used while if "numeric" is chosen numeric values input controls will be displayed.

## Value

A new folder in the current working directory called DynNomapp which contains all the required scripts to deploy this dynamic nomogram on a server on the web such as the <http://shinyapps.io>. This folder includes ui.R, server.R and global.R script files needed to build the application and dataset.rds which is the accompanying dataset and a user guide text file called README.txt which explains how to deploy the app using all these objects.

## Author(s)

Amirhossein Jalali, Alberto Alvarez-Iglesias, John Newell

## See Also

[lm](#), [DynNom](#), [DynNom.lm](#)

## Examples

```
## Not run:
# a linear regression model
model1 <- lm(Fertility ~ Agriculture + Education + Catholic , data = swiss)
DNbuilder.lm(model1, swiss)

model2 <- lm(uptake ~ Plant + conc + Plant * conc, data = CO2)
DNbuilder(model2, CO2)

## End(Not run)

if (interactive()) {
  data1 <- data.frame(state.x77)
  fit1 <- lm(formula = Life.Exp ~ ., data = data1)
  DNbuilder(fit1, data1)
}
```

---

DynNom	<i>Dynamic Nomograms for Linear, Generalized Linear and Proportional Hazards Models</i>
--------	---

---

### Description

DynNom is a generic function for displaying the results of an statistical model object as a dynamic nomogram in an 'RStudio' panel or web browser. DynNom supports a variety of model objects; `lm`, `glm`, `coxph` and also `ols`, `Glm`, `lrm`, `cph` models in the `rms` package. It is a translational tool aiming to provide easy, informative individual predictions.

### Usage

```
DynNom(model, data, clevel = 0.95, m.summary = c("raw", "formatted"),
        covariate = c("slider", "numeric"), ptype = c("st", "1-st"))
```

### Arguments

<code>model</code>	an <code>lm</code> , <code>glm</code> , <code>coxph</code> , <code>ols</code> , <code>Glm</code> , <code>lrm</code> or <code>cph</code> model object
<code>data</code>	dataframe containing the accompanying data
<code>clevel</code>	confidence level required
<code>m.summary</code>	The option to choose the format of the model output in the 'Summary Model' tab. If "raw" (the default) is chosen the result of <code>summary(model)</code> will be display while if "formatted" is chosen the model summary using the <code>stargazer</code> package will be displayed.
<code>covariate</code>	The option to choose the type of covariate(s) input control widget for numeric values. If "slider" (the default) is chosen a shiny application with slider control widgets are used while if "numeric" is chosen numeric values input controls will be displayed.
<code>ptype</code>	This plot type option relates to <code>coxph</code> objects only. If "st" (the default) is chosen, a plot of the estimated survivor function, $S(t)$ , is displayed. If "1-st" is chosen a plot of $1 - S(t)$ is displayed.

### Value

A dynamic nomogram in a shiny application which recognises all the predictors in the model and uses them to build a sidebar panel. It sets up drop down menus for factors and sliders set at the mean and bounded by the range for covariates.

The individual predictions with a relative confidence interval are calculated using the `predict` function, displaying either graphically as an interactive plot in the `Graphical Summary` tab or a table in the `Numerical Summary` tab. A table of model output is also available in the `Model Summary` tab. In the case of the Cox proportional hazards model, estimated survivor/death function will be additionally plotted in an extra tab.



**Please cite as:**

Jalali, A., Roshan, D., Alvarez-Iglesias, A., Newell, J. (2017). Dynamic Nomograms for Linear, Generalized Linear and Proportional Hazard Models. R package version 4.1.

**Author(s)**

Amirhossein Jalali, Davood Roshan, Alberto Alvarez-Iglesias, John Newell

Maintainer: Amirhossein Jalali <a.jalali2@nuigalway.ie>

**References**

Banks, J. 2006. Nomograms. Encyclopedia of Statistical Sciences. 8.

Easy web applications in R. <http://shiny.rstudio.com>

Frank E Harrell Jr (2017). rms: Regression Modeling Strategies. R package version 4.5-0. <https://CRAN.R-project.org/package=rms>

**See Also**

[DynNom.lm](#), [DynNom.glm](#), [DynNom.coxph](#), [DynNom.ols](#), [DynNom.lrm](#), [DynNom.Glm](#), [DynNom.cph](#)

**Examples**

```
## Not run:
# simple linear regression models
model1 <- lm(uptake ~ Plant + conc + Plant * conc, data = CO2)
DynNom(model1, CO2)

data1 <- data.frame(state.x77)
model2 <- ols(Life.Exp ~ Population + Income + Illiteracy + Murder + HS.Grad +
Frost + Area,data=data1)
DynNom(model2, data1)

# Generalized regression models
data2 =as.data.frame(Titanic)
model3 <- glm(Survived ~ Age + Class + Sex, data = data2, weights = Freq,
family = binomial("probit"))
DynNom(model3, data2, clevel = 0.9)

model4 <- lrm(formula= vs ~ wt + disp, data = mtcars)
DynNom(model4, mtcars, clevel = 0.9, m.summary = "formatted")

counts <- c(18, 17, 15, 20, 10, 20, 25, 13, 12)
outcome <- gl(3, 1, 9)
treatment <- gl(3, 3)
data2 = data.frame(counts, outcome, treatment)
model5 <- Glm((2 * counts) ~ outcome + treatment, family = poisson(), data = data2)
DynNom.Glm(model5, data2)

# a proportional hazard model
data.kidney <- kidney
# always make sure that the categorical variables are in a factor class
```

```

data.kidney$sex <- as.factor(data.kidney$sex)
levels(data.kidney$sex) <- c("male", "female")

model6 <- coxph(Surv(time, status) ~ age + sex + disease, data.kidney)
DynNom(model6, data.kidney)
DynNom(model6, data.kidney, ptype = "1-st")

model7 <-cph((Surv(log(time), status)) ~ rcs(age, 4) * strat(trt) +
             diagtime * strat(prior) + lsp(karno, 60), data = veteran)
DynNom(model7, veteran)

## End(Not run)

if (interactive()) {
# a poisson regression model
model8 <- glm(event ~ mag + station + dist + accel, data = attenu, family = poisson)
DynNom(model8, attenu, covariate = "numeric")
}

```

---

DynNom.coxph

*Dynamic Nomograms for Proportional Hazards Models*


---

## Description

DynNom.coxph displays the results of a coxph model object as a dynamic nomogram in an 'RStudio' panel or web browser.

## Usage

```

DynNom.coxph(model, data, clevel = 0.95, m.summary = c("raw", "formatted"),
             covariate = c("slider", "numeric"), ptype = c("st", "1-st"))

```

## Arguments

model	a coxph model object
data	dataframe containing the accompanying data
clevel	confidence level required
m.summary	The option to choose the format of the model output in the 'Summary Model' tab. If "raw" (the default) is chosen the result of summary(model) will be display while if "formatted" is chosen the model summary using the stargazer package will be displayed.
covariate	The option to choose the type of covariate(s) input control widget for numeric values. If "slider" (the default) is chosen a shiny application with slider control widgets are used while if "numeric" is chosen numeric values input controls will be displayed.
ptype	If "st" (the default) is chosen, a plot of the estimated survivor function, $S(t)$ , is displayed. If "1-st" is chosen a plot of $1 - S(t)$ is displayed.

**Value**

A dynamic nomogram in a shiny application which recognises all the predictors in the model and uses them to build a sidebar panel. It sets up drop down menus for factors and sliders set at the mean and bounded by the range for covariates.

The individual predictions with a relative confidence interval are calculated using the predict function, displaying graphically as either the Kaplan-Meier in the Estimated  $S(t)$  tab or the Predicted Survival tab. Table of individual predictions and model output are available in the Numerical Summary and Model Summary tabs, respectively.

**Author(s)**

Amirhossein Jalali, Davood Roshan, Alberto Alvarez-Iglesias, John Newell

**See Also**

[coxph](#), [predict.coxph](#)

**Examples**

```
## Not run:
data.kidney <- kidney
# always make sure that the categorical variables are in a factor class
data.kidney$sex <- as.factor(data.kidney$sex)
levels(data.kidney$sex) <- c("male", "female")

model1 <- coxph(Surv(time, status) ~ age + sex + disease, data.kidney)
DynNom(model1, data.kidney)
DynNom(model1, data.kidney, ptype = "1-st")

# a cox model including a strata term
data(lung)
model2 <- coxph(Surv(time, status) ~ age + strata(sex) + ph.ecog , data = lung)
DynNom(model2, lung)

## End(Not run)

if (interactive()) {
data.ovary <- ovarian
data.ovary$resid.ds <- as.factor(data.ovary$resid.ds)
levels(data.ovary$resid.ds) <- c("no", "yes")
data.ovary$rx <- as.factor(data.ovary$rx)
data.ovary$ecog.ps <- as.factor(data.ovary$ecog.ps)

model3 <- coxph(Surv(futime, fustat) ~ age + resid.ds * rx + ecog.ps, data = data.ovary)
DynNom(model3, data.ovary)
}
```

---

DynNom.cph	<i>Dynamic Nomograms for Cox Proportional Hazards Models from the rms package</i>
------------	---

---

### Description

DynNom.cph displays the results of a cph model object from rms package as a dynamic nomogram in an 'RStudio' panel or web browser.

### Usage

```
DynNom.cph(model, data, clevel = 0.95, m.summary = c("raw", "formatted"),
            covariate = c("slider", "numeric"), ptype = c("st", "1-st"))
```

### Arguments

model	a cph model object which accepts a variety of transformation functions such as asis, pol, lsp, rcs, catg, scored, strat and matrx as defined in the rms package.
data	dataframe containing the accompanying data
clevel	confidence level required
m.summary	The option to choose the format of the model output in the 'Summary Model' tab. If "raw" (the default) is chosen the result of summary(model) will be display while if "formatted" is chosen the model summary using the stargazer package will be displayed.
covariate	The option to choose the type of covariate(s) input control widget for numeric values. If "slider" (the default) is chosen a shiny application with slider control widgets are used while if "numeric" is chosen numeric values input controls will be displayed.
ptype	If "st" (the default) is chosen, a plot of the estimated survivor function, $S(t)$ , is displayed. If "1-st" is chosen a plot of $1 - S(t)$ is displayed.

### Value

A dynamic nomogram in a shiny application which recognises all the predictors in the model and uses them to build a sidebar panel. It sets up drop down menus for factors and sliders set at the mean and bounded by the range for covariates.

The individual predictions with a relative confidence interval are calculated using the predict function, displaying graphically as either the Kaplan-Meier in the Estimated  $S(t)$  tab or the Predicted Survival tab. Table of individual predictions and model output are available in the Numerical Summary and Model Summary tabs, respectively.

### Author(s)

Davood Roshan, Amirhossein Jalali, Alberto Alvarez-Iglesias, John Newell

**See Also**

[cph](#), [predict.cph](#), [rms](#)

**Examples**

```
## Not run:
# example 1
data = veteran
model1 <- cph((Surv(log(time), status)) ~ rcs(age, 4) * strat(trt) +
              diagtime * strat(prior) + lsp(karno, 60), data = veteran)
model1 <- update(model1, x = T, y = T, surv = T)
DynNom.cph(model1, data)
DynNom(model1, data, ptype = "1-st")

# example 2
data(lung)
sfit = Surv(lung$time, lung$status)
model2 <- cph(sfit ~ age + strat(sex) + ph.ecog , data = lung)
DynNom.cph(model2, lung)

## End(Not run)

if (interactive()) {
  data.ovary <- ovarian
  data.ovary$resid.ds <- as.factor(data.ovary$resid.ds)
  levels(data.ovary$resid.ds) <- c("no", "yes")
  data.ovary$rx <- as.factor(data.ovary$rx)
  data.ovary$ecog.ps <- as.factor(data.ovary$ecog.ps)

  model3 <- cph(Surv(futime, fustat) ~ age + resid.ds * rx + ecog.ps, data = data.ovary)
  DynNom(model3, data.ovary)
}
```

---

DynNom.Glm

*Dynamic Nomograms for Generalized Linear Models from the rms package*

---

**Description**

DynNom.Glm displays the results of a Glm model object from the rms package as a dynamic nomogram in an 'RStudio' panel or web browser.

**Usage**

```
DynNom.Glm(model, data, clevel = 0.95, m.summary = c("raw", "formatted"),
           covariate = c("slider", "numeric"))
```

**Arguments**

model	a Glm model object which accepts a variety of transformation functions such as <code>asis</code> , <code>pol</code> , <code>lsp</code> , <code>rcs</code> , <code>catg</code> , <code>scored</code> , <code>strat</code> and <code>matrx</code> defined in <code>rms</code> package.
data	dataframe containing the accompanying data
clevel	confidence level required
m.summary	The option to choose the format of the model output in the 'Summary Model' tab. If "raw" (the default) is chosen the result of <code>summary(model)</code> will be display while if "formatted" is chosen the model summary using the <code>stargazer</code> package will be displayed.
covariate	The option to choose the type of covariate(s) input control widget for numeric values. If "slider" (the default) is chosen a shiny application with slider control widgets are used while if "numeric" is chosen numeric values input controls will be displayed.

**Value**

A dynamic nomogram in a shiny application which recognises all the predictors in the model and uses them to build a sidebar panel. It sets up drop down menus for factors and sliders set at the mean and bounded by the range for covariates.

The individual predictions with a relative confidence interval are calculated using the `predict` function, displaying either graphically as an interactive plot in the `Graphical Summary` tab or a table in the `Numerical Summary` tab. A table of model output is also available in the `Model Summary` tab.

**Author(s)**

Davoud Roshan, Amirhossein Jalali, Alberto Alvarez-Iglesias, John Newell

**See Also**

[Glm](#), [predict.Glm](#), [rms](#)

**Examples**

```
## Not run:
# example 1 - a generalized linear model
set.seed(1)
x1 <- runif(200)
x2 <- sample(0:3, 200, TRUE)
x3 <- sample(0:2, 200, TRUE)

distance <- (x1 + x2 / 3 + rnorm(200)) ^ 2
d <- datadist(x1, x2)
options(datadist = "d")
data1 = data.frame(distance, x1, x2, x3)
model1 <- Glm(distance ~ x3 + rcs(x1, 4) * scored(x2), data = data1)
DynNom.Glm(model1, data1)

# example 2 - a poisson regression model
counts <- c(18, 17, 15, 20, 10, 20, 25, 13, 12)
```

```

outcome <- gl(3, 1, 9)
treatment <- gl(3, 3)
data2 = data.frame(counts, outcome, treatment)
model2 <- Glm((2 * counts) ~ outcome + treatment, family = poisson(), data = data2)
DynNom.Glm(model2, data2)

## End(Not run)

if (interactive()) {
# a Gamma regression model
clotting <- data.frame(
  u = c(5, 10, 15, 20, 30, 40, 60, 80, 100),
  lot1 = c(118, 58, 42, 35, 27, 25, 21, 19, 18),
  lot2 = c(69, 35, 26, 21, 18, 16, 13, 12, 12),
  cat = c(rep("A",5), rep("B",4)))
model3 <- Glm(lot1 ~ log(u) + cat, data = clotting, family = Gamma)
DynNom.Glm(model3, clotting)
}

```

---

DynNom.glm

*Dynamic Nomograms for Generalized Linear Models*


---

## Description

DynNom.glm displays the results of a glm model object as a dynamic nomogram in an 'RStudio' panel or web browser.

## Usage

```

DynNom.glm(model, data, clevel = 0.95, m.summary = c("raw", "formatted"),
           covariate = c("slider", "numeric"))

```

## Arguments

model	a glm model object
data	dataframe containing the accompanying data
clevel	confidence level required
m.summary	The option to choose the format of the model output in the 'Summary Model' tab. If "raw" (the default) is chosen the result of summary(model) will be display while if "formatted" is chosen the model summary using the stargazer package will be displayed.
covariate	The option to choose the type of covariate(s) input control widget for numeric values. If "slider" (the default) is chosen a shiny application with slider control widgets are used while if "numeric" is chosen numeric values input controls will be displayed.

**Value**

A dynamic nomogram in a shiny application which recognises all the predictors in the model and uses them to build a sidebar panel. It sets up drop down menus for factors and sliders set at the mean and bounded by the range for covariates.

The individual predictions with a relative confidence interval are calculated using the `predict` function, displaying either graphically as an interactive plot in the `Graphical Summary` tab or a table in the `Numerical Summary` tab. A table of model output is also available in the `Model Summary` tab.

**Author(s)**

Amirhossein Jalali, Alberto Alvarez-Iglesias, John Newell

**See Also**

[glm](#), [predict.glm](#)

**Examples**

```
## Not run:
# a generalized linear model
model1 <- glm(Fertility ~ Agriculture + Education + Catholic, data = swiss)
DynNom(model1, swiss, clevel = 0.9)

# a logistic regression model
data1 =as.data.frame(Titanic)
model2 <- glm(Survived ~ Age + Class + Sex, data = data1, weights = Freq,
              family = binomial("probit"))
DynNom(model2, as.data.frame(Titanic), clevel = 0.9, m.summary = "formatted")

## End(Not run)

if (interactive()) {
# a poisson regression model
model3 <- glm(event ~ mag + dist + accel, data = attenu, family = poisson)
DynNom(model3, attenu, covariate = "numeric")
}
```

---

DynNom.lm

*Dynamic Nomograms for Linear Models*

---

**Description**

DynNom.lm displays the results of an lm model object as a dynamic nomogram in an 'RStudio' panel or web browser.

**Usage**

```
DynNom.lm(model, data, clevel = 0.95, m.summary = c("raw", "formatted"),
           covariate = c("slider", "numeric"))
```



**Arguments**

model	an lm model object
data	dataframe containing the accompanying data
clevel	confidence level required
m.summary	The option to choose the format of the model output in the 'Summary Model' tab. If "raw" (the default) is chosen the result of summary(model) will be display while if "formatted" is chosen the model summary using the stargazer package will be displayed.
covariate	The option to choose the type of covariate(s) input control widget for numeric values. If "slider" (the default) is chosen a shiny application with slider control widgets are used while if "numeric" is chosen numeric values input controls will be displayed.

**Value**

A dynamic nomogram in a shiny application which recognises all the predictors in the model and uses them to build a sidebar panel. It sets up drop down menus for factors and sliders set at the mean and bounded by the range for covariates.

The individual predictions with a relative confidence interval are calculated using the predict function, displaying either graphically as an interactive plot in the Graphical Summary tab or a table in the Numerical Summary tab. A table of model output is also available in the Model Summary tab.

**Author(s)**

Amirhossein Jalali, Alberto Alvarez-Iglesias, John Newell

**See Also**

[lm](#), [predict.lm](#)

**Examples**

```
## Not run:
# a linear regression model
model1 <- lm(Fertility ~ Agriculture + Education + Catholic , data = swiss)
DynNom(model1, swiss)

model2 <- lm(uptake ~ Plant + conc + Plant * conc, data = CO2)
DynNom(model2, CO2)

## End(Not run)

if (interactive()) {
  data1 <- data.frame(state.x77)
  fit1 <- lm(formula = Life.Exp ~ ., data = data1)
  DynNom(fit1, data1)
}
```

---

DynNom.lrm	<i>Dynamic Nomograms for Logistic Regression Models from the rms package</i>
------------	--

---

## Description

DynNom.lrm displays the results of a lrm model object from the rms package as a dynamic nomogram in an 'RStudio' panel or web browser.

## Usage

```
DynNom.lrm(model, data, clevel = 0.95, m.summary = c("raw", "formatted"),
           covariate = c("slider", "numeric"))
```

## Arguments

model	a lrm model object which accepts a variety of transformation functions such as asis, pol, lsp, rcs, catg, scored, strat and matrix defined in rms package.
data	dataframe containing the accompanying data
clevel	confidence level required
m.summary	The option to choose the format of the model output in the 'Summary Model' tab. If "raw" (the default) is chosen the result of summary(model) will be display while if "formatted" is chosen the model summary using the stargazer package will be displayed.
covariate	The option to choose the type of covariate(s) input control widget for numeric values. If "slider" (the default) is chosen a shiny application with slider control widgets are used while if "numeric" is chosen numeric values input controls will be displayed.

## Value

A dynamic nomogram in a shiny application which recognises all the predictors in the model and uses them to build a sidebar panel. It sets up drop down menus for factors and sliders set at the mean and bounded by the range for covariates.

The individual predictions with a relative confidence interval are calculated using the predict function, displaying either graphically as an interactive plot in the Graphical Summary tab or a table in the Numerical Summary tab. A table of model output is also available in the Model Summary tab.

## Author(s)

Davoud Roshan, Amirhossein Jalali, Alberto Alvarez-Iglesias, John Newell

## See Also

[lrm](#), [predict.lrm](#), [rms](#)

**Examples**

```

## Not run:
# example
n <- 1000
set.seed(17)
age <- rnorm(n, 50, 10)
blood.pressure <- rnorm(n, 120, 15)
cholesterol <- rnorm(n, 200, 25)
sex <- factor(sample(c('female', 'male'), n, TRUE))
label(age) <- 'Age' # label is in Hmisc
label(cholesterol) <- 'Total Cholesterol'
label(blood.pressure) <- 'Systolic Blood Pressure'
label(sex) <- 'Sex'
units(cholesterol) <- 'mg/dl'
units(blood.pressure) <- 'mmHg'

ch <- cut2(cholesterol, g = 40, levels.mean = TRUE)

d <- data.frame(age = seq(0, 90, by = 10))

L <- .4 * (sex == 'male') + .045 * (age - 50) +
  (log(cholesterol - 10) - 5.2) * (-2 * (sex == 'female') + 2 * (sex == 'male'))
y <- ifelse(runif(n) < plogis(L), 1, 0)
cholesterol[1:3] <- NA

ddist <- datadist(age, blood.pressure, cholesterol, sex)
options(datadist = 'ddist')

data = data.frame(y = y, blood.pressure = blood.pressure, sex = sex, age = age,
  cholesterol = cholesterol)
model <- lrm(y ~ blood.pressure + sex * (age + rcs(cholesterol, 4)),
  x = TRUE, y = TRUE, m.summary = "formatted")

DynNom.lrm(model, data, m.summary = "formatted")

## End(Not run)

if (interactive()) {
fit <- lrm(formula = vs ~ wt + disp, data = mtcars)
DynNom.lrm(fit, mtcars, clevel = 0.9)
}

```

**Description**

DynNom.ols displays the results of an ols model object from the rms package as a dynamic nomogram in an 'RStudio' panel or web browser.

**Usage**

```
DynNom.ols(model, data, clevel = 0.95, m.summary = c("raw", "formatted"),
           covariate = c("slider", "numeric"))
```

**Arguments**

model	an ols model object which accepts a variety of transformation functions such as <code>asis</code> , <code>pol</code> , <code>lsp</code> , <code>rcs</code> , <code>catg</code> , <code>scored</code> , <code>strat</code> and <code>matrx</code> defined in <code>rms</code> package.
data	dataframe containing the accompanying data
clevel	confidence level required
m.summary	The option to choose the format of the model output in the 'Summary Model' tab. If "raw" (the default) is chosen the result of <code>summary(model)</code> will be display while if "formatted" is chosen the model summary using the <code>stargazer</code> package will be displayed.
covariate	The option to choose the type of covariate(s) input control widget for numeric values. If "slider" (the default) is chosen a shiny application with slider control widgets are used while if "numeric" is chosen numeric values input controls will be displayed.

**Value**

A dynamic nomogram in a shiny application which recognises all the predictors in the model and uses them to build a sidebar panel. It sets up drop down menus for factors and sliders set at the mean and bounded by the range for covariates.

The individual predictions with a relative confidence interval are calculated using the `predict` function, displaying either graphically as an interactive plot in the `Graphical Summary` tab or a table in the `Numerical Summary` tab. A table of model output is also available in the `Model Summary` tab.

**Author(s)**

Davood Roshan, Amirhossein Jalali, Alberto Alvarez-Iglesias, John Newell

**See Also**

[ols](#), [predict.ols](#), [rms](#)

**Examples**

```
## Not run:
# example 1
x1 <- runif(200)
x2 <- runif(200)
x3 <- runif(200)
x4 <- runif(200)
y <- x1 + x2 + rnorm(200)
data = data.frame(x1, x2, x3, x4, y)
f <- ols(y ~ rcs(x1, 4) + x2 + x3 + x4)
DynNom.ols(f, data)
```

```
# example 2
data1 = as.data.frame(Titanic)
year <- sample(c(1:5), 32, replace = TRUE)
data <- data.frame(year, data1)
model <- ols(year ~ Age + Class + Sex, data = data, weights = Freq)
DynNom.ols(model, data)

## End(Not run)

if (interactive()) {
data1 <- data.frame(state.x77)
fit1 <- ols(Life.Exp ~ Population + Income + Murder + Frost , data = data1)
DynNom(fit1, data1)
}
```

# Index

## \*Topic **Cox Proportional Hazards Model**

DNbuilder.coxph, 3

DynNom.coxph, 10

## \*Topic **cox proportional hazards model**

DynNom.cph, 12

## \*Topic **dynamic nomograms**

DNbuilder, 2

DNbuilder.coxph, 3

DNbuilder.glm, 5

DNbuilder.lm, 6

DynNom, 8

DynNom.coxph, 10

DynNom.cph, 12

DynNom.Glm, 13

DynNom.glm, 15

DynNom.lm, 16

DynNom.lrm, 18

DynNom.ols, 19

## \*Topic **generalized linear models**

DNbuilder.glm, 5

DynNom.Glm, 13

DynNom.glm, 15

## \*Topic **individual prediction**

DNbuilder, 2

DNbuilder.coxph, 3

DNbuilder.glm, 5

DNbuilder.lm, 6

DynNom, 8

DynNom.coxph, 10

DynNom.cph, 12

DynNom.Glm, 13

DynNom.glm, 15

DynNom.lm, 16

DynNom.lrm, 18

DynNom.ols, 19

## \*Topic **linear models**

DNbuilder.lm, 6

DynNom.lm, 16

DynNom.ols, 19

## \*Topic **logistic regression models**

DynNom.lrm, 18

## \*Topic **shiny**

DNbuilder, 2

DNbuilder.coxph, 3

DNbuilder.glm, 5

DNbuilder.lm, 6

DynNom, 8

DynNom.coxph, 10

DynNom.cph, 12

DynNom.Glm, 13

DynNom.glm, 15

DynNom.lm, 16

DynNom.lrm, 18

DynNom.ols, 19

coxph, 3, 4, 11

cph, 13

DNbuilder, 2

DNbuilder.coxph, 3

DNbuilder.glm, 5

DNbuilder.lm, 6

DynNom, 3, 4, 6, 7, 8

DynNom.coxph, 3, 4, 9, 10

DynNom.cph, 9, 12

DynNom.Glm, 9, 13

DynNom.glm, 3, 6, 9, 15

DynNom.lm, 3, 7, 9, 16

DynNom.lrm, 9, 18

DynNom.ols, 9, 19

Glm, 14

glm, 3, 6, 16

lm, 3, 7, 17

lrm, 18

ols, 20

`predict.coxph`, [11](#)  
`predict.cph`, [13](#)  
`predict.Glm`, [14](#)  
`predict.glm`, [16](#)  
`predict.lm`, [17](#)  
`predict.lrm`, [18](#)  
`predict.ols`, [20](#)  
  
`rms`, [13](#), [14](#), [18](#), [20](#)