# Package 'DynNom'

January 20, 2025

**Type** Package

**Title** Visualising Statistical Models using Dynamic Nomograms

**Version** 5.1

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

**Maintainer** Amirhossein Jalali <amir.jalali@ul.ie>

**Description** Demonstrate the results of a statistical model object as a dynamic nomogram in an RStudio panel or web browser. The package provides two generics functions: DynNom, which display statistical model objects as a dynamic nomogram; DNbuilder, which builds required scripts to publish a dynamic nomogram on a web server such as the <https://www.shinyapps.io/>. Current version of 'DynNom' supports stats::lm, stats::glm, survival::coxph, rms::ols, rms::Glm, rms::lrm, rms::cph, and mgcv::gam model objects.

**License** GPL-2

**Imports** magrittr, shiny, ggplot2 (> 2.1.0), plotly, stargazer, dplyr, compare, BBmisc, broom, rms, survival

**Suggests** mgcv

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2024-06-07 12:20:21 UTC

## Contents

DNbuilder *Publishing a dynamic nomogram*

## Description

DNbuilder is a generic function which builds required scripts to publish a dynamic nomogram on a web server such as the https://www.shinyapps.io/. This application can then access through a URL and be used independent of R. DNbuilder supports a large number of model objects from a variety of packages.

## Usage

```
DNbuilder(model, data = NULL, clevel = 0.95, m.summary = c("raw", "formatted"),
  covariate = c("slider", "numeric"), ptype = c("st", "1-st"),
  DNtitle = NULL, DNxlab = NULL, DNylab = NULL, DNlimits = NULL,
  KMtitle = NULL, KMxlab = NULL, KMylab = NULL)

DNbuilder.core(model, data, clevel, m.summary, covariate,
  DNtitle, DNxlab, DNylab, DNlimits)

DNbuilder.surv(model, data, clevel, m.summary, covariate,
  ptype, DNtitle, DNxlab, DNylab, KMtitle, KMxlab, KMylab)
```

## Arguments

| | |
|---|---|
| model | an lm, glm, coxph, ols, Glm, lrm, cph or mgcv::gam model object. |
| data | a dataframe of the accompanying dataset for the model (if required). |
| clevel | a confidence level for constructing the confidence interval. If not specified, a 95% level will be used. |
| m.summary | an option to choose the type of the model output represented in the 'Summary Model' tab. "raw" (the default) returns an unformatted summary of the model; "formatted" returns a formatted table of the model summary using stargazer package. |
| covariate | an option to choose the type of input control widgets used for numeric values. "slider" (the default) picks out sliderInput; "numeric" picks out numericInput. |
| ptype | an option for coxph or cph model objects to choose the type of plot which displays in "Survival plot" tab. "st" (the default) returns plot of estimated survivor probability (S(t)). "1-st" returns plot of estimated failure probability (1-S(t)). |
| DNtitle | a character vector used as the app's title. If not specified, "Dynamic Nomogram" will be used. |
| DNxlab | a character vector used as the title for the x-axis in "Graphical Summary" tab. If not specified, "Probability" will be used for logistic model and Cox proportional model objects; or "Response variable" for other model objects. |
| DNylab | a character vector used as the title for the y-axis in "Graphical Summary" tab (default is NULL). |

| | |
|---|---|
| DNlimits | a vector of 2 numeric values used to set x-axis limits in "Graphical Summary" tab. Note: This also removes the 'Set x-axis ranges' widget in the sidebar panel. |
| KMtitle | a character vector used as KM plot's title in "Survival plot" tab. If not specified, "Estimated Survival Probability" for ptype = "st" and "Estimated Probability" for ptype = "1-st" will be used. |
| KMxlab | a character vector used as the title for the x-axis in "Survival plot" tab. If not specified, "Follow Up Time" will be used. |
| KMylab | a character vector used as the title for the y-axis in "Survival plot" tab. If not specified, "S(t)" for ptype = "st" and "F(t)" for ptype = "1-st" will be used. |

## Value

A new folder called 'DynNomapp' will be created in the current working directory which contains all the required scripts to deploy this dynamic nomogram on a host server such as the https://www.shinyapps.io/. This folder includes ui.R, server.R, global.R and data.RData which needs to publish the app. A user guide text file (README.txt) will be also added to explain how to deploy the app using these files.

## Please cite as:

Jalali A, Alvarez-Iglesias A, Roshan D, Newell J (2019) Visualising statistical models using dynamic nomograms. PLOS ONE 14(11): e0225253. https://doi.org/10.1371/journal.pone.0225253

## 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. https://www.rstudio.com/products/shiny/
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, getpred.DN

## Examples

```
## Not run:
# Simple linear regression models
fit1 <- lm(uptake ~ Plant + conc + Plant * conc, data = CO2)
DNbuilder(fit1)

t.data <- datadist(swiss)
options(datadist = 't.data')
ols(Fertility ~ Agriculture + Education + rcs(Catholic, 4), data = swiss) %>%
```

```
  DNbuilder(clevel = 0.9, m.summary="formatted")

# Generalized regression models
fit2 <- glm(Survived ~ Age + Class + Sex,
  data = as.data.frame(Titanic), weights = Freq, binomial("probit"))
DNbuilder(fit2, DNtitle="Titanic", DNxlab = "Probability of survival")

counts <- c(18, 17, 15, 20, 10, 20, 25, 13, 12)
outcome <- gl(3, 1, 9)
treatment <- gl(3, 3)
d <- datadist(treatment, outcome)
options(datadist = "d")
Glm((2 * counts) ~ outcome + treatment, family = poisson(),
  data = data.frame(counts, outcome, treatment)) %>%
  DNbuilder()

# Proportional hazard models
coxph(Surv(time, status) ~ age + strata(sex) + ph.ecog, data = lung) %>%
  DNbuilder()

data.kidney <- kidney
data.kidney$sex <- as.factor(data.kidney$sex)
levels(data.kidney$sex) <- c("male", "female")
coxph(Surv(time, status) ~ age + strata(sex) + disease, data.kidney) %>%
  DNbuilder(ptype = "1-st")

d <- datadist(veteran)
options(datadist = "d")
fit3 <- cph((Surv(time/30, status)) ~ rcs(age, 4) * strat(trt) + diagtime +
  strat(prior) + lsp(karno, 60), veteran)
DNbuilder(fit3, DNxlab = "Survival probability",
  KMtitle="Kaplan-Meier plot", KMxlab = "Time (Days)", KMylab = "Survival probability")

# Generalized additive models
mgcv::gam(Fertility ~ s(Agriculture) + Education + s(Catholic), data=swiss) %>%
  DNbuilder(DNlimits = c(0, 110), m.summary="formatted")

## End(Not run)
if (interactive()) {
  data(rock)
  lm(area~I(log(peri)), data = rock) %>%
    DNbuilder()
}
```

---

DynNom *Dynamic nomogram to visualise statistical models*

---

## Description

DynNom is a generic function to display the results of statistical model objects as a dynamic nomogram in an 'RStudio' panel or web browser. DynNom supports a large number of model objects from

a variety of packages.

## Usage

```
DynNom(model, data = NULL, clevel = 0.95, m.summary = c("raw", "formatted"),
  covariate = c("slider", "numeric"), ptype = c("st", "1-st"),
  DNtitle = NULL, DNxlab = NULL, DNylab = NULL, DNlimits = NULL,
  KMtitle = NULL, KMxlab = NULL, KMylab = NULL)

DynNom.core(model, data, clevel, m.summary, covariate, DNtitle, DNxlab, DNylab, DNlimits)

DynNom.surv(model, data, clevel, m.summary, covariate,
  ptype, DNtitle, DNxlab, DNylab, KMtitle, KMxlab, KMylab)
```

## Arguments

| | |
|---|---|
| model | an lm, glm, coxph, ols, Glm, lrm, cph or mgcv::gam model object. |
| data | a dataframe of the accompanying dataset for the model (if required). |
| clevel | a confidence level for constructing the confidence interval. If not specified, a 95% level will be used. |
| m.summary | an option to choose the type of the model output represented in the 'Summary Model' tab. "raw" (the default) returns an unformatted summary of the model; "formatted" returns a formatted table of the model summary using stargazer package. |
| covariate | an option to choose the type of input control widgets used for numeric values. "slider" (the default) picks out sliderInput; "numeric" picks out numericInput. |
| ptype | an option for coxph or cph model objects to choose the type of plot which displays in "Survival plot" tab. "st" (the default) returns plot of estimated survivor probability (S(t)). "1-st" returns plot of estimated failure probability (1-S(t)). |
| DNtitle | a character vector used as the app's title. If not specified, "Dynamic Nomogram" will be used. |
| DNxlab | a character vector used as the title for the x-axis in "Graphical Summary" tab. If not specified, "Probability" will be used for logistic model and Cox proportional model objects; or "Response variable" for other model objects. |
| DNylab | a character vector used as the title for the y-axis in "Graphical Summary" tab (default is NULL). |
| DNlimits | a vector of 2 numeric values used to set x-axis limits in "Graphical Summary" tab. Note: This also removes the 'Set x-axis ranges' widget in the sidebar panel. |
| KMtitle | a character vector used as KM plot's title in "Survival plot" tab. If not specified, "Estimated Survival Probability" for ptype = "st" and "Estimated Probability" for ptype = "1-st" will be used. |
| KMxlab | a character vector used as the title for the x-axis in "Survival plot" tab. If not specified, "Follow Up Time" will be used. |
| KMylab | a character vector used as the title for the y-axis in "Survival plot" tab. If not specified, "S(t)" for ptype = "st" and "F(t)" for ptype = "1-st" will be used. |

## Value

A dynamic nomogram in a shiny application providing individual predictions which can be used as a model visualisation or decision-making tools.

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, an estimated survivor/failure function will be additionally displayed in a new tab.

## Please cite as:

Jalali A, Alvarez-Iglesias A, Roshan D, Newell J (2019) Visualising statistical models using dynamic nomograms. PLOS ONE 14(11): e0225253. https://doi.org/10.1371/journal.pone.0225253

## 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. https://www.rstudio.com/products/shiny/
Frank E Harrell Jr (2017). rms: Regression Modeling Strategies. R package version 4.5-0. https://CRAN.R-project.org/package=rms/

## See Also

`DNbuilder`, `getpred.DN`

## Examples

```
## Not run:
# Simple linear regression models
fit1 <- lm(uptake ~ Plant + conc + Plant * conc, data = CO2)
DynNom(fit1)

t.data <- datadist(swiss)
options(datadist = 't.data')
ols(Fertility ~ Agriculture + Education + rcs(Catholic, 4), data = swiss) %>%
  DynNom(clevel = 0.9, m.summary="formatted")

# Generalized regression models
fit2 <- glm(Survived ~ Age + Class + Sex,
  data = as.data.frame(Titanic), weights = Freq, family = binomial("probit"))
DynNom(fit2, DNtitle="Titanic", DNxlab = "Probability of survival")

counts <- c(18, 17, 15, 20, 10, 20, 25, 13, 12)
outcome <- gl(3, 1, 9)
treatment <- gl(3, 3)
```

```
d <- datadist(treatment, outcome)
options(datadist = "d")
Glm((2 * counts) ~ outcome + treatment, family = poisson(),
  data = data.frame(counts, outcome, treatment)) %>%
  DynNom()

# Proportional hazard models
coxph(Surv(time, status) ~ age + strata(sex) + ph.ecog, data = lung) %>%
  DynNom()

data.kidney <- kidney
data.kidney$sex <- as.factor(data.kidney$sex)
levels(data.kidney$sex) <- c("male", "female")
coxph(Surv(time, status) ~ age + strata(sex) + disease, data.kidney) %>%
  DynNom(ptype = "1-st")

d <- datadist(veteran)
options(datadist = "d")
fit3 <- cph((Surv(time/30, status)) ~ rcs(age, 4) * strat(trt) + diagtime +
  strat(prior) + lsp(karno, 60), veteran)
DynNom(fit3, DNxlab = "Survival probability",
     KMtitle="Kaplan-Meier plot", KMxlab = "Time (Days)", KMylab = "Survival probability")

# Generalized additive models
mgcv::gam(Fertility ~ s(Agriculture) + Education + s(Catholic), data=swiss) %>%
  DynNom(DNlimits = c(0, 110), m.summary="formatted")

## End(Not run)
if (interactive()) {
  data(rock)
  lm(area~I(log(peri)), data = rock) %>%
    DynNom()
}
```

---

`getclass.DN` *Extract class and family of a model object*

---

## Description

`getclass.DN` extracts class and family of a model object (supported in `DynNom`).

## Usage

```
getclass.DN(model)
```

## Arguments

| | |
|---|---|
| `model` | an `lm`, `glm`, `coxph`, `ols`, `Glm`, `lrm`, cph or `mgcv::gam` model objects. |

## Value

A list including the model class and the family name of the model (if relevant).

## See Also

DynNom, DNbuilder

## Examples

```
fit1 <- glm(Survived ~ Age + Class + Sex, data = as.data.frame(Titanic),
  weights = Freq, family = binomial("probit"))
getclass.DN(fit1)

library(survival)
fit2 <- coxph(Surv(time, status) ~ age + strata(sex) + ph.ecog, data = lung)
getclass.DN(fit2)
```

---

# getdata.DN *Extract dataset from a model object*

---

## Description

getdata.DN extracts dataset that was used to produce the model object (supported in DynNom).

## Usage

```
getdata.DN(model)
```

## Arguments

| | |
|---|---|
| model | an lm, glm, coxph, ols, Glm, lrm, cph or mgcv::gam model objects. |

## Value

A data.frame containing the dataset used in the fitted model object.

## See Also

DynNom, DNbuilder

## Examples

```
fit1 <- glm(Survived ~ Age + Class + Sex, data = as.data.frame(Titanic),
  weights = Freq, family = binomial("probit"))
getdata.DN(fit1)

library(survival)
fit2 <- coxph(Surv(time, status) ~ age + strata(sex) + ph.ecog, data = lung)
getdata.DN(fit2)
```

getpred.DN *Extract predictions from a Model Object*

## Description

getpred.DN extracts class, family and inverse of link function from a model object (supported in DynNom).

## Usage

```
getpred.DN(model, newd, set.rms=F)
```

## Arguments

| | |
|---|---|
| model | an lm, glm, coxph, ols, Glm, lrm, cph or mgcv::gam model objects. |
| newd | a data frame of predictors for prediction |
| set.rms | a logical value indicating if data should be updated in the model object (required for rms model objects in DNbuilder). |

## Value

A list including the prediction (pred) and the standard error of prediction (SEpred).

## See Also

DynNom, DNbuilder

## Examples

```
fit1 <- glm(Survived ~ Age + Class + Sex, data = as.data.frame(Titanic),
  weights = Freq, family = binomial("probit"))
getpred.DN(fit1, newd = data.frame(Class="1st", Sex="Male", Age="Child"))
```

# Index