Package ‘rmda’

July 23, 2025

Type Package

Title Risk Model Decision Analysis

Version 1.6

Date 2018-07-17

Author Marshall Brown

Maintainer Marshall Brown <mdbrown@fredhutch.org>

Description Provides tools to evaluate the value of using a risk prediction instrument to decide treatment or intervention (versus no treatment or intervention). Given one or more risk prediction instruments (risk models) that estimate the probability of a binary outcome, rmda provides functions to estimate and display decision curves and other figures that help assess the population impact of using a risk model for clinical decision making. Here, ``population'' refers to the relevant patient population. Decision curves display estimates of the (standardized) net benefit over a range of probability thresholds used to categorize observations as 'high risk'. The curves help evaluate a treatment policy that recommends treatment for patients who are estimated to be 'high risk' by comparing the population impact of a risk-based policy to ``treat all'' and ``treat none'' intervention policies. Curves can be estimated using data from a prospective cohort. In addition, rmda can estimate decision curves using data from a case-control study if an estimate of the population outcome prevalence is available. Version 1.4 of the package provides an alternative framing of the decision problem for situations where treatment is the standard-of-care and a risk model might be used to recommend that low-risk patients (i.e., patients below some risk threshold) opt out of treatment. Confidence intervals calculated using the bootstrap can be computed and displayed. A wrapper function to calculate cross-validated curves using k-fold cross-validation is also provided.

License GPL-2

URL http://mdbrown.github.io/rmda/, https://github.com/mdbrown/rmda

LazyData TRUE

Imports reshape, pander, MASS, caret

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2018-07-17 17:30:02 UTC

Contents

`rmda-package` *rmda: Risk Model Decision Analysis*

Description

The package 'rmda' (risk model decision analysis) provides tools to evaluate the value of using a risk prediction instrument to decide treatment or intervention (versus no treatment or intervention). Given one or more risk prediction instruments (risk models) that estimate the probability of a binary outcome, rmda provides functions to estimate and display decision curves and other figures that help assess the population impact of using a risk model for clinical decision making. Here, "population" refers to the relevant patient population.

Details

Decision curves display estimates of the (standardized) net benefit over a range of probability thresholds used to categorize observations as 'high risk'. The curves help evaluate a treatment policy that recommends treatment for patients who are estimated to be 'high risk' by comparing the population impact of a risk-based policy to "treat all" and "treat none" intervention policies. Curves can be estimated using data from a prospective cohort. In addition, rmda can estimate decision curves using data from a case-control study if an estimate of the population outcome prevalence is available. Version 1.4 of the package provides an alternative framing of the decision problem for situations where treatment is the standard-of-care and a risk model might be used to recommend that low-risk patients (i.e., patients below some risk threshold) opt out of treatment.

Confidence intervals calculated using the bootstrap can be computed and displayed. A wrapper function to calculate cross-validated curves using k-fold cross-validation is also provided.

Functions in this package include:

`decision_curve`,

`summary.decision_curve`,

`plot_decision_curve`,

`plot_clinical_impact`,

`plot_roc_components`,

`cv_decision_curve` and

`Add_CostBenefit_Axis`.

See Also

Useful links:

- `http://mdbrown.github.io/rmda/`
- `https://github.com/mdbrown/rmda`

`Add_CostBenefit_Axis` *Add cost benefit ratio axis to a decision curve plot.*

Description

Add cost benefit ratio axis to a decision curve plot.

Usage

```
Add_CostBenefit_Axis(xlim, cost.benefits, n.cost.benefits = 6, line = 4,
  policy, ...)
```

Arguments

`xlim`	range of x-axis.
`cost.benefits`	Character vector of the form c('c1:b1', 'c2:b2', ..., 'cn:bn') with integers ci, bi corresponding to specific cost:benefit ratios to print.
`n.cost.benefits`	number of cost:benefit ratios to print if cost.benefit.axis = TRUE (default n.cost.benefit = 6).
`line`	x-axis line to print the axis (default is 4).
`policy`	Either 'opt-in' (default) or 'opt-out', describing the type of policy for which to report the net benefit. A policy is 'opt-in' when the standard-of-care for a population is to assign a particular 'treatment' to no one. Clinicians then use a risk model to categorize patients as 'high-risk', with the recommendation to treat high-risk patients with some intervention. Alternatively, an 'opt-out' policy is applicable to contexts where the standard-of-care is to recommend a treatment to an entire patient population. The potential use of a risk model in this setting is to identify patients who are 'low-risk' and recommend that those patients 'opt-out' of treatment.
`...`	other options sent to 'axis'.

Value

List with components threshold, value and name.

See Also

`summary.decision_curve`, `decision_curve`

`cv_decision_curve` *Calculate cross-validated decision curves*

Description

This is a wrapper for 'decision_curve' that computes k-fold cross-validated estimates of sensitivity, specificity, and net benefit so that cross-validated net benefit curves can be plotted.

Usage

```
cv_decision_curve(formula, data, family = binomial(link = "logit"),
  thresholds = seq(0, 1, by = 0.01), folds = 5, study.design = c("cohort",
  "case-control"), population.prevalence, policy = c("opt-in", "opt-out"))
```

Arguments

`formula`	an object of class 'formula' of the form outcome ~ predictors, giving the prediction model to be fitted using glm. The outcome must be a binary variable that equals '1' for cases and '0' for controls.
`data`	data.frame containing outcome and predictors. Missing data on any of the predictors will cause the entire observation to be removed.
`family`	a description of the error distribution and link function to pass to 'glm" used for model fitting. Defaults to binomial(link = "logit") for logistic regression.
`thresholds`	Numeric vector of high risk thresholds to use when plotting and calculating net benefit values.
`folds`	Number of folds for k-fold cross-validation.
`study.design`	Either 'cohort' (default) or 'case-control' describing the study design used to obtain data. See details for more information.
`population.prevalence`	Outcome prevalence rate in the population used to calculate decision curves when study.design = 'case-control'.
`policy`	Either 'opt-in' (default) or 'opt-out', describing the type of policy for which to report the net benefit. A policy is 'opt-in' when the standard-of-care for a population is to assign a particular 'treatment' to no one. Clinicians then use a risk model to categorize patients as 'high-risk', with the recommendation to treat high-risk patients with some intervention. Alternatively, an 'opt-out' policy is applicable to contexts where the standard-of-care is to recommend a treatment to an entire patient population. The potential use of a risk model in this setting is to identify patients who are 'low-risk' and recommend that those patients 'opt-out' of treatment.

Value

List with components

- derived.data: derived.data: A data frame in long form showing the following for each predictor and each 'threshold', 'FPR':false positive rate, 'TPR': true positive rate, 'NB': net benefit, 'sNB': standardized net benefit, 'rho': outcome prevalence, 'prob.high.risk': percent of the population considered high risk. 'DP': detection probability = TPR*rho, 'model': name of prediction model or 'all' or 'none', and cost.benefit.ratio's.
- folds: number of folds used for cross-validation.
- call: matched function call.

See Also

`summary.decision_curve`, `decision_curve`, `Add_CostBenefit_Axis`

Examples

```
full.model_cv <- cv_decision_curve(Cancer~Age + Female + Smokes + Marker1 + Marker2,
                                  data = dcaData,
                                  folds = 5,
                                  thresholds = seq(0, .4, by = .01))

full.model_apparent <- decision_curve(Cancer~Age + Female + Smokes + Marker1 + Marker2,
                                     data = dcaData,
                                     thresholds = seq(0, .4, by = .01),
                                     confidence.intervals = 'none')

plot_decision_curve( list(full.model_apparent, full.model_cv),
                    curve.names = c('Apparent curve', 'Cross-validated curve'),
                    col = c('red', 'blue'),
                    lty = c(2,1),
                    lwd = c(3,2, 2, 1),
                    legend.position = 'bottomright')
```

`dcaData` *Simulated dataset for package 'DecisionCurve'*

Description

Simulated cohort data containing demographic variables, marker values and cancer outcome.

Usage

```
dcaData
```

Format

A data frame with 500 rows and 6 variables:

- Age: Age in years.
- Female: Indicator for female gender.
- Smokes: Indicator for smoking status.
- Marker1: simulated biomarker.
- Marker2: simulated biomarker.
- Cancer: Indicator for cancer.

`dcaData_cc` *Simulated dataset for package 'DecisionCurve'*

Description

Simulated data from a case-control study containing demographic variables, marker values and cancer outcome. The prevalence of cancer in the population that this data is sampled from is approximately 0.11.

Usage

```
dcaData_cc
```

Format

A data frame with 500 rows and 6 variables:

- Age: Age in years.
- Female: Indicator for female gender.
- Smokes: Indicator for smoking status.
- Marker1: simulated biomarker.
- Marker2: simulated biomarker.
- Cancer: Indicator for cancer.

`decision_curve` *Calculate net benefit/decision curves*

Description

This function calculates decision curves, which are estimates of the standardized net benefit by the probability threshold used to categorize observations as 'high risk.' Curves can be estimated using data from an observational cohort (default), or from case-control studies when an estimate of the population outcome prevalence is available. Confidence intervals calculated using the bootstrap are calculated as well. Once this function is called, use `plot_decision_curve` or `summary` to plot or view the curves, respectively.

Usage

```
decision_curve(formula, data, family = binomial(link = "logit"),
  policy = c("opt-in", "opt-out"), fitted.risk = FALSE,
  thresholds = seq(0, 1, by = 0.01), confidence.intervals = 0.95,
  bootstraps = 500, study.design = c("cohort", "case-control"),
  population.prevalence)
```

Arguments

`formula`	an object of class 'formula' of the form outcome ~ predictors, giving the prediction model to be fitted using glm. The outcome must be a binary variable that equals '1' for cases and '0' for controls.
`data`	data.frame containing outcome and predictors. Missing data on any of the predictors will cause the entire observation to be removed.
`family`	a description of the error distribution and link function to pass to 'glm' used for model fitting. Defaults to binomial(link = 'logit') for logistic regression.
`policy`	Either 'opt-in' (default) or 'opt-out', describing the type of policy for which to report the net benefit. A policy is 'opt-in' when the standard-of-care for a population is to assign a particular 'treatment' to no one. Clinicians then use a risk model to categorize patients as 'high-risk', with the recommendation to treat high-risk patients with some intervention. Alternatively, an 'opt-out' policy is applicable to contexts where the standard-of-care is to recommend a treatment to an entire patient population. The potential use of a risk model in this setting is to identify patients who are 'low-risk' and recommend that those patients 'opt-out' of treatment.
`fitted.risk`	logical (default FALSE) indicating whether the predictor provided are estimated risks from an already established model. If set to TRUE, no model fitting will be done and all estimates will be conditional on the risks provided. Risks must fall between 0 and 1.
`thresholds`	Numeric vector of high risk thresholds to use when plotting and calculating net benefit values.

`confidence.intervals`	Numeric (default 0.95 for 95% confidence bands) level of bootstrap confidence intervals to plot. Set as NA or 'none' to remove confidence intervals. See details for more information.
`bootstraps`	Number of bootstrap replicates to use to calculate confidence intervals (default 500).
`study.design`	Either 'cohort' (default) or 'case-control' describing the study design used to obtain data. See details for more information.
`population.prevalence`	Outcome prevalence rate in the population used to calculate decision curves when study.design = 'case-control'.

Details

Confidence intervals for (standardized) net benefit are calculated pointwise at each risk threshold. For when data come from an observational cohort, bootstrap sampling is done without stratifying on outcome, so disease prevalence varies within bootstrap samples. For case-control data, bootstrap sampling is done stratified on outcome.

Value

List with components

- derived.data: A data frame in long form showing the following for each predictor and each 'threshold', 'FPR':false positive rate, 'TPR': true positive rate, 'NB': net benefit, 'sNB': standardized net benefit, 'rho': outcome prevalence, 'prob.high.risk': percent of the population considered high risk. DP': detection probability = TPR*rho, 'model': name of prediction model or 'all' or 'none', cost.benefit.ratio, and 'xx_lower', 'xx_upper': the lower and upper confidence bands for all measures (if calculated).
- confidence.intervals: Level of confidence intervals returned.
- call: matched function call.

See Also

`summary.decision_curve`, `cv_decision_curve`, `Add_CostBenefit_Axis`

Examples

```
#helper function
expit <- function(xx) exp(xx)/ (1+exp(xx))

#load simulated cohort data
data(dcaData)
baseline.model <- decision_curve(Cancer~Age + Female + Smokes,
                                data = dcaData,
                                thresholds = seq(0, .4, by = .01),
                                study.design = 'cohort',
                                bootstraps = 10) #number of bootstraps should be higher

full.model <- decision_curve(Cancer~Age + Female + Smokes + Marker1 + Marker2,
```

```
                              data = dcaData,
                              thresholds = seq(0, .4, by = .01),
                              bootstraps = 10)

#simulated case-control data with same variables as above
data(dcaData_cc)

table(dcaData_cc$Cancer)

#estimated from the population where the
#case-control sample comes from.
population.rho = 0.11

full.model_cc <- decision_curve(Cancer~Age + Female + Smokes + Marker1 + Marker2,
                                 data = dcaData,
                                 thresholds = seq(0, .4, by = .01),
                                 bootstraps = 10,
                                 study.design = 'case-control',
                                 population.prevalence = population.rho)

#estimate the net benefit for an 'opt-out' policy.
nb.opt.out  <- decision_curve(Cancer~Age + Female + Smokes + Marker1 + Marker2,
                              data = dcaData,
                              policy = 'opt-out',
                              thresholds = seq(0, .4, by = .01),
                              bootstraps = 10)
```

`plot_clinical_impact` *Plot the clinical impact curve from a DecisionCurve object.*

Description

For a given population size, plot the number of subjects classified as high risk, and the number of subjects classified high risk with the outcome of interest at each high risk threshold.

Usage

```
plot_clinical_impact(x, population.size = 1000, cost.benefit.axis = TRUE,
  n.cost.benefits = 6, cost.benefits, confidence.intervals, col = "black",
  lty = 1, lwd = 2, xlim, ylim, xlab, ylab,
  cost.benefit.xlab = "Cost:Benefit Ratio", legend.position = c("topright",
  "right", "bottomright", "bottom", "bottomleft", "left", "topleft", "top",
  "none"), ...)
```

Arguments

x	decision_curve object to plot. Assumes output from function ’decision_curve’

`population.size`	Hypothetical population size (default 1000).
`cost.benefit.axis`	logical (default TRUE) indicating whether to print an additional x-axis showing relative cost:benefit ratios in addition to risk thresholds.
`n.cost.benefits`	number of cost:benefit ratios to print if cost.benefit.axis = TRUE (default n.cost.benefit = 6).
`cost.benefits`	Character vector of the form c("c1:b1", "c2:b2", ..., "cn:bn") with integers ci, bi corresponding to specific cost:benefit ratios to print. Default allows the function to calculate these automatically.
`confidence.intervals`	logical indicating whether to plot confidence intervals.
`col`	vector of length two indicating the color for the number high risk and the second to the number high risk with outcome, respectively.
`lty`	vector of linetypes. The first element corresponds to the number high risk and the second to the number high risk with outcome.
`lwd`	vector of linewidths. The first element corresponds to the number high risk and the second to the number high risk with outcome.
`xlim`	vector giving c(min, max) of x-axis. Defaults to c(min(thresholds), max(thresholds)).
`ylim`	vector giving c(min, max) of y-axis.
`xlab`	label of main x-axis.
`ylab`	label of y-axis.
`cost.benefit.xlab`	label of cost:benefit ratio axis.
`legend.position`	character vector giving position of legend. Options are "topright" (default), "right", "bottomright", "bottom", "bottomleft", "left", "topleft", "top", or "none".
`...`	other options directly send to plot()

Examples

```
#'data(dcaData)
set.seed(123)
baseline.model <- decision_curve(Cancer~Age + Female + Smokes,
                                data = dcaData,
                                thresholds = seq(0, .4, by = .001),
                       bootstraps = 25) #should use more bootstrap replicates in practice!

#plot the clinical impact
plot_clinical_impact(baseline.model, xlim = c(0, .4),
                    col = c(”black”, ”blue”))
```

`plot_decision_curve` *Plot the net benefit curves from a decision_curve object or many decision_curve objects*

Description

Plot the net benefit curves from a decision_curve object or many decision_curve objects

Usage

```
plot_decision_curve(x, curve.names, cost.benefit.axis = TRUE,
  n.cost.benefits = 6, cost.benefits, standardize = TRUE,
  confidence.intervals, col, lty, lwd = 2, xlim, ylim, xlab, ylab,
  cost.benefit.xlab, legend.position = c("topright", "right", "bottomright",
  "bottom", "bottomleft", "left", "topleft", "top", "none"), ...)
```

Arguments

`x`	'decision_curve' object to plot or a list of 'decision_curve' objects. Assumes output from function 'decision_curve'
`curve.names`	vector of names to use when plotting legends.
`cost.benefit.axis`	logical (default TRUE) indicating whether to print an additional x-axis showing relative cost:benefit ratios in addition to risk thresholds.
`n.cost.benefits`	number of cost:benefit ratios to print if cost.benefit.axis = TRUE (default n.cost.benefit = 6).
`cost.benefits`	Character vector of the form c("c1:b1", "c2:b2", ..., "cn:bn") with integers ci, bi corresponding to specific cost:benefit ratios to print. Default allows the function to calculate these automatically.
`standardize`	logical (default TRUE) indicating whether to use the standardized net benefit (NB/disease prevalence) or not.
`confidence.intervals`	logical indicating whether to plot confidence intervals.
`col`	vector of color names to be used in plotting corresponding to the 'predictors' given. Default colors will be chosen from rainbow(..., v = .8). See details for more information on plot parameters.
`lty`	vector of linetypes.
`lwd`	vector of linewidths.
`xlim`	vector giving c(min, max) of x-axis. Defaults to c(min(thresholds), max(thresholds)).
`ylim`	vector giving c(min, max) of y-axis.
`xlab`	label of main x-axis.
`ylab`	label of y-axis.

`cost.benefit.xlab`	label of cost:benefit ratio axis.
`legend.position`	character vector giving position of legend. Options are "topright" (default), "right", "bottomright", "bottom", "bottomleft", "left", "topleft", "top", or "none".
`...`	other options directly send to plot()

Details

When k decision_curve objects are input, the first k elements of col, lty, lwd ... correspond to the curves provided. The next two elements (..., k+1, k+2) correspond to the attributes of the ’all’ and ’none’ curves. See below for an example.

Examples

```
data(dcaData)
set.seed(123)
baseline.model <- decision_curve(Cancer~Age + Female + Smokes,
                                data = dcaData,
                                thresholds = seq(0, .4, by = .005),
                                bootstraps = 10)

#plot using the defaults
plot_decision_curve(baseline.model,  curve.names = "baseline model")

set.seed(123)
full.model <- decision_curve(Cancer~Age + Female + Smokes + Marker1 + Marker2,
                            data = dcaData,
                            thresholds = seq(0, .4, by = .005),
                            bootstraps = 10)

# for lwd, the first two positions correspond to the decision curves, then 'all' and 'none'
plot_decision_curve( list(baseline.model, full.model),
                    curve.names = c("Baseline model", "Full model"),
                    col = c("blue", "red"),
                    lty = c(1,2),
                    lwd = c(3,2, 2, 1),
                    legend.position = "bottomright")


plot_decision_curve( list(baseline.model, full.model),
                    curve.names = c("Baseline model", "Full model"),
                    col = c("blue", "red"),
                    confidence.intervals = FALSE,  #remove confidence intervals
                    cost.benefit.axis = FALSE, #remove cost benefit axis
                    legend.position = "none") #remove the legend

#Set specific cost:benefit ratios.

plot_decision_curve( list(baseline.model, full.model),
                    curve.names = c("Baseline model", "Full model"),
                    col = c("blue", "red"),
```

```
                    cost.benefits = c("1:1000", "1:4", "1:9", "2:3", "1:3"),
                    legend.position = "bottomright")

#Plot net benefit instead of standardize net benefit.

plot_decision_curve( list(baseline.model, full.model),
                    curve.names = c("Baseline model", "Full model"),
                    col = c("blue", "red"),
                    ylim = c(-0.05, 0.15), #set ylim
                    lty = c(2,1),
                standardize = FALSE, #plot Net benefit instead of standardized net benefit
                   legend.position = "topright")
```

`plot_roc_components` *Plot the components of a ROC curve by the high risk thresholds.*

Description

Plot the components of the ROC curve –the true positive rates and false positive rates– by high risk thresholds.

Usage

```
plot_roc_components(x, cost.benefit.axis = TRUE, n.cost.benefits = 6,
  cost.benefits, confidence.intervals, col = "black", lty.fpr = 2,
  lty.tpr = 1, lwd = 2, xlim, ylim, xlab = "Risk Threshold", ylab,
  cost.benefit.xlab = "Cost:Benefit Ratio", legend.position = c("topright",
  "right", "bottomright", "bottom", "bottomleft", "left", "topleft", "top",
  "none"), ...)
```

Arguments

`x`	decision_curve object to plot. Assumes output from function ’decision_curve’
`cost.benefit.axis`	logical (default TRUE) indicating whether to print an additional x-axis showing relative cost:benefit ratios in addition to risk thresholds.
`n.cost.benefits`	number of cost:benefit ratios to print if cost.benefit.axis = TRUE (default n.cost.benefit = 6).
`cost.benefits`	Character vector of the form c("c1:b1", "c2:b2", ..., "cn:bn") with integers ci, bi corresponding to specific cost:benefit ratios to print. Default allows the function to calculate these automatically.
`confidence.intervals`	logical indicating whether to plot confidence intervals.
`col`	vector of length two indicating the color for the true positive rates and false positive rates, respectively.

`lty.fpr`	linetype for the false positive rate curve.
`lty.tpr`	linetype for the true positive rate curve.
`lwd`	vector of linewidths. The first element corresponds to the tpr and the second to the fpr.
`xlim`	vector giving c(min, max) of x-axis. Defaults to c(min(thresholds), max(thresholds)).
`ylim`	vector giving c(min, max) of y-axis.
`xlab`	label of main x-axis.
`ylab`	label of y-axis.
`cost.benefit.xlab`	label of cost:benefit ratio axis.
`legend.position`	character vector giving position of legend. Options are "topright" (default), "right", "bottomright", "bottom", "bottomleft", "left", "topleft", "top", or "none".
`...`	other options directly send to plot()

Examples

```
data(dcaData)
set.seed(123)
baseline.model <- decision_curve(Cancer~Age + Female + Smokes,
                                 data = dcaData,
                                 thresholds = seq(0, .4, by = .001),
                         bootstraps = 25) #should use more bootstrap replicates in practice!

#plot using the defaults
plot_roc_components(baseline.model,  xlim = c(0, 0.4), col = c("black", "red"))
```

`summary.decision_curve` *Displays a useful description of a decision_curve object*

Description

Displays a useful description of a decision_curve object

Usage

```
## S3 method for class 'decision_curve'
summary(object, ..., measure = c("sNB", "NB", "TPR",
  "FPR", "TNR", "FNR"), nround = 3)
```

Arguments

`object`	decision_curve object to summarise
`...`	other arguments ignored (for compatibility with generic)
`measure`	name of summary measure to print out. For standardized net benefit: "sNB" (default), net benefit: "NB", true positive rate: "TPR", false positive rate: "FPR".
`nround`	number of decimal places to round (default 3).

Examples

```
#helper function

#load simulated data
data(dcaData)

full.model <- decision_curve(Cancer~Age + Female + Smokes + Marker1 + Marker2,
data = dcaData,
thresholds = seq(0, .4, by = .05),
bootstraps = 25)

summary(full.model) #outputs standardized net benefit by default

summary(full.model, nround = 2, measure = "TPR")
```

Index