Package 'Qindex'

March 16, 2023

Type Package

Title Quantile-Based Predictors for Survival Outcome

Version 0.1.2

Date 2023-03-16

Author Misung Yi [aut, cph] (<https://orcid.org/0000-0002-4007-5408>),
Tingting Zhan [aut, cre, cph] (<https://orcid.org/0000-0001-9971-4844>),
Inna Chervoneva [aut, cph] (<https://orcid.org/0000-0002-9104-4505>)

Maintainer Tingting Zhan <tingtingzhan@gmail.com>

Description Select optimal quantile-based predictors for survival outcome,
to include the means to handle dichotomizing the quantiles,
mean-signal-intensity or other continuous markers, and to obtain
estimates for dichotomized predictors by Bootstrap-based bias
correction.

LazyData true

LazyDataCompression xz

RoxygenNote 7.2.3

Encoding UTF-8

License GPL-2

Depends R (>= 4.2),

Language en-US

Imports stats, boot, rpart, survival, matrixStats, mgcv, grDevices

Suggests knitr

Collate '0info.R' 'pkg_data_doc.R' 'optQp.R' 'sampleQp.R'
'BBC_dichotom.R' 'FRindex.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2023-03-16 18:00:10 UTC

R topics documented:

`Qindex-package` *Qindex: Continuous and dichotomized index predictors based on distribution quantiles*

Description

Select optimal functional or dichotomized quantile-based predictors for survival/logistic/numeric outcome and perform optimal dichotomization with optimistic bias correction for any continuous predictors

Details

The package provides tools to

1. use sampleQp to calculate user-selected sample quantiles in each independent cluster of observations. This function is simply a wrapper of aggregate for quantile function.
2. use eval_dichotom to estimate the effect size for dichotomized user-selected predictors of interest (e.g. sample quantiles) (absolute value of the corresponding hazard ratio, odds ratio or regression coefficient) using repeated random split train/test sampling. In this function, we first use rpart to identify optimal cutoff in each training set and use this cutoff to dichotomize each predictor of interest in the corresponding independent test set. The effect size for dichotomized predictor is estimated in the test set by fitting coxph, glm or lm to fit a Cox proportional hazard model, logistic regression, or linear regression for Surv, logical, or numeric endpoint.
3. use optQp to select the set of optimal quantiles that has the largest effect size (absolute value of the corresponding hazard ratio, odds ratio or regression coefficient) for a given Surv, logical, or numeric endpoint. optQp is a wrapper of summary.eval_dichotom.
4. use BBC_dichotom to dichotomize predictors of interest and to obtain bootstrap-based optimism corrected effect size from Cox model, logistic regression, or linear regression. Internally, BBC_dichotom calls dichotom_int to dichotomize each predictor in `contX` based on univariate model setting and `model_dichotom` to fit Cox proportional hazard model, logistic

regression, or linear regression for Surv, logical, or numeric endpoint with the dichotomized predictors from dichotom_int.

5. use FRindex to derive a scalar functional regression index as a predictor in the functional regression model with any response supported by **gam** package. Function FRindex calls gam to fit a generalized additive model(GAM) to the training set and makes use of plot.gam to extract functional coefficient tabulated on the same grid as functional predictor(s) in the training and test set (if the test set is provided).

References

Selection of optimal quantile protein biomarkers based on cell-level immunohistochemistry data, Misung Yi, Tingting Zhan , Amy P. Peck, Jeffrey A. Hooke, Albert J. Kovatich, Craig D. Shriver, Hai Hu, Yunguang Sun, Hallgeir Rui and Inna Chervoneva, under review

Quantile index biomarkers based on single-cell expression data, Misung Yi, Tingting Zhan , Amy P. Peck, Jeffrey A. Hooke, Albert J. Kovatich, Craig D. Shriver, Hai Hu, Yunguang Sun, Hallgeir Rui and Inna Chervoneva, under review

---

`BBC_dichotom` *Bootstrap Bias Correction for Dichotomization*

Description

Bootstrap-based optimism correction for dichotomizing selected continuous predictor(s)

Usage

```
BBC_dichotom(formula, data, contX, R = 200L, ...)
```

Arguments

`formula`	formula, with a Surv, logical, or numeric endpoint and covariates to be included as is (i.e., without dichotomization).
`data`	data.frame
`contX`	character scalar, name of the matrix containing the continuous predictor(s) to be dichotomized optimally
`R`	integer scalar, number of bootstrap samples, see boot. Default `200L`.
`...`	additional parameters, currently not in use

Details

The bootstrap optimism correction procedure is performed as described for a general model selection. First, `R` bootstrap samples are drawn with replacement from the main sample. In each bootstrap sample, the recursive partitioning tree model is used to establish an objective data-driven optimal cut-point for selected continuous predictors The cut-point from the current bootstrap sample is used to compute the effect size (log hazards ratio (HR), odds ratio (OR), or coefficient) for each dichotomized predictor in the current bootstrap sample (**bootstrap performance**) and in the main

sample (**test performance**), and the optimism in log OR/HR or coefficient estimation is computed as the difference between log OR/HR or coefficient for "Bootstrap performance" and for "Test performance". The median optimism estimate is computed as the median of optimism estimates over all bootstrap samples. The cutpoint for dichotomizing each selected continuous predictor is also established in the main sample and its "apparent performance" is computed as the log OR/HR or coefficient for dichotomized quantile in the univariate Cox models, logistic regression, or linear regression. Finally, the optimism-corrected performance estimate is computed by subtracting the median optimism estimate from the apparent performance estimate.

Value

BBC_dichotom returns a numeric bootstrap-based bias adjusted coefficients of model, corresponding median optimism, R bootstrap resampling based thresholds, and a single apparent performance threshold

References

Ewout W. Steyerberg (2009) Clinical Prediction Models. doi:10.1007/9780387772448

Frank E. Harrell Jr., Kerry L. Lee, Daniel B. Mark. (1996) Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. doi:10.1002/(SICI)10970258(19960229)15:4<361::AIDSIM168>3.0.CO;24

Examples

```
# see ?FRindex
# see ?eval_dichotom
```

---

celldata *Ki67 Data*

Description

Ki67 cell data containing 622 patients

Usage

```
Ki67
```

Format

PATIENT_ID factor, unique patient identifier

tissueID factor, ID of the TMA core

RECURRENCE integer, recurrence indicator, 1 = Recurred, 0 = not Recurred

RECFREESURV_MO integer, recurrence-free survival time in months

Marker numeric, cell signal intensity of the protein immunofloerscence signal

`inner_x` integer, x-coordinate in the cell centroid in the TMA core

`inner_y` integer, y-coordinate in the cell centroid in the TMA core

`AGE_AT_DX` integer, age at diagnosis

`Tstage` integer, tumor stage

`NodeSt` integer, node stage, -1 = unknown, 0 = Node Negative, 1 = Node Positive

`HRpos` integer, indicator of hormone positive status (ER+ or PR+), 1 = positive, 0 = negative

`HistologicalGrade` integer, histology grade

`Her2_path_qIF` integer, Her2 status, 1 = positive, 0 = negative

`RACE` character, race, White, Black, Asian, Native Hawaiian or Other Pacific Islander, American Indian or Alaska Native, Unknown

`RadjCHEMO` integer, adjuvant chemo treatment, 0 = unknown, 1 = done, 2 = NOT done

`RadjRAD` integer, adjuvant radiation treatment, 0 = unknown, 1 = done, 2 = NOT done

`HORM_4cat` integer, hormone treatment, 0 = unknown, 1 = not indicated, 2 = done, 3 = recommended, but not done

`MSI` numeric, mean signal intensity (mean over all cells in the TMA core)

`dichotom_int` *Dichotomizing*

Description

Generate dichotimized variables by a threshold calculated from rpart using Surv, logical, or numeric endpoint and each quantile

Usage

```
dichotom_int(edp, contX)
```

Arguments

`edp`	Surv, logical, or numeric object, the endpoint
`contX`	see BBC_dichotom

Value

dichotom_int returns a logical matrix of the same dimension and dimension names as argument `contX`, as dichotomized using the first node of rpart as threshold. The thresholds (per column) are returned as an numeric vector in attribute `'thres'`.

`eval_dichotom` *Optimal Quantile Predictor*

Description

From a given set of sample quantiles, this function selects the optimal quantile with the largest effect size for predicting given Surv, logical, or numeric outcome

Usage

```
eval_dichotom(formula, data, seeds, pct_train = 0.8, ...)
```

Arguments

`formula`	formula, supports Surv, logical, or numeric endpoint and one matrix predictor of the descriptive statistics of markers per cluster. See details of parameter `data`.
`data`	data.frame, with at least • two Surv columns including time-to-event and event indicator, one logical column, or one numeric column as the outcome • one matrix column as the descriptive statistics of markers per cluster. Currently only a quantile sequence is supported.
`seeds`	integer vector of random seeds for generating repeated random split samples, see set.seed
`pct_train`	numeric scalar, proportion of the training set, default `.8`
`...`	additional parameters, currently not in use

Details

Optimal quantile selection algorithm For a sample $x_i, 1 \leq i \leq n$ of repeated measures of independent variable X and probability $p, 0 < p < 1$, the empirical quantile function $Q_n(p)$ is defined as the k th order statistic of the sample, where k is such that $(k-1)/n < p < k/n$. For $p = .01, \cdots, .99$ and $k = 100p$, $Q_n(p)$ is also known as kth percentile. The following algorithm is proposed to identify the optimal $Q(p)$ predictor of survival outcome in a screening data set:

1. Select the set of quantiles to be evaluated as predictors and the desired ratio for training/test sets.
2. Split the data into training and test sets
   - In case of Surv outcomes, split the group of subjects with event into a training set and a test set randomly with desired ratio. Similarly, split the group of subjects without event into a training set and a test set randomly with desired ratio. Combine training sets with and without event and test sets with and without event.
   - In case of logical outcomes, split the group of subjects with one level of outcome into a training set and a test set randomly with desired ratio. Similarly, split the group of subjects with the other level of outcome into a training set and a test set randomly with desired ratio. Combine training sets with both levels of logical outcome and test sets with both levels of logical outcome.

- In case of numeric outcomes, split the entire subjects into a training set and a test set randomly with desired ratio.

3. For each training/test set pair and each considered quantile,
   - Determine the optimal cutoff (e.g. using the R package rpart) in the combined training set.
   - Apply the optimal cutoff to the combined test set and estimate the effect size (hazard ratio, odds ratio, or exponentiated coefficient).
4. Repeat steps 2 and 3 for 100 training/test splits, compute the median log effect size (log hazard ratio(HR), log odds ratio(OR), or coefficient) for each quantile.
5. Rank the effect sizes (absolute value of log OR, HR, or coefficient) for all considered quantiles and select the optimal quantile with the highest effect size.
6. Perform bootstrap-based optimism correction for the selected optimal quantile(s). In this work, for each random split, 80% of subjects were assigned to the training set and 20% of subjects were assigned to the test set. We considered every fifth quantile starting from the 5th quantile to the 95th quantile plus 99th quantile as candidate predictors of PFS (a total of 20 quantile predictors). Also, we identified quantiles with the second and third highest effect sizes to compare them to the optimal ones.

Value

eval_dichotom returns a eval_dichotom object, which is a list with elements elements `thresholds` and `coefs`.

`thresholds` matrix, data set with all cut points for all candidate quantiles and Np columns and Nsplit rows

`coefs` matrix, coefficients corresponding to thresholds

`data` data.frame, this is the unmodified input `data`

References

Selection of optimal quantile protein biomarkers based on cell-level immunohistochemistry data, Misung Yi, Tingting Zhan , Amy P. Peck, Jeffrey A. Hooke, Albert J. Kovatich, Craig D. Shriver, Hai Hu, Yunguang Sun, Hallgeir Rui and Inna Chervoneva, under review

Examples

```
if (FALSE) { # masked to save time
library(survival)

Ki67_Qps = sampleQp(data = Ki67, subjID = 'PATIENT_ID',
  exclude = c('tissueID','inner_x','inner_y'), Qpredictor = 'Marker')
Ki67c = eval_dichotom(Surv(RECFREESURV_MO, RECURRENCE) ~ Marker, data = Ki67_Qps,
  seeds = 1:20)
# summary(Ki67c) # works, but not needed
head(Ki67_opt <- optQp(Ki67c, n = 2L))

set.seed(1)
```

```
mod_c = BBC_dichotom(Surv(RECFREESURV_MO, RECURRENCE) ~ NodeSt + Tstage,
  data = optQp(Ki67c, n = 2L), contX = 'Marker', R = 100)
summary(mod_c)
}

if (FALSE) { # mask to save time
Ki67a = eval_dichotom(RECFREESURV_MO ~ Marker, data = Ki67_Qps, seeds = 1:20)
set.seed(1)
mod_a = BBC_dichotom(RECFREESURV_MO ~ NodeSt + Tstage, data = optQp(Ki67a, n = 2L),
  contX = 'Marker', R = 100)
summary(mod_a)

Ki67b = eval_dichotom(RECURRENCE ~ Marker, data = Ki67_Qps, seeds = 1:20)
set.seed(1)
mod_b = BBC_dichotom(RECURRENCE ~ NodeSt + Tstage, data = optQp(Ki67b, n = 2L),
  contX = 'Marker', R = 100)
summary(mod_b)
}
```

FRindex *Functional Regression Index (FRindex)*

Description

Computes a scalar functional index as a predictor in the functional regression model. The training data are used to estimate the functional coefficient by fitting functional regression model with any response supported by **gam** package. The functional regression index is computed for training and (optional) test data using the functional coefficient from the model fitted to the training data.

Usage

```
FRindex(
  trainData,
  testData,
  trainArgs = attr(trainData, which = "xarg", exact = TRUE),
 testArgs = if (!missing(testData)) attr(testData, which = "xarg", exact = TRUE) else
    trainArgs,
  response,
  family,
  predictor = "Marker",
  log = TRUE,
  knot_pct = 0.4,
  knot.value = ceiling(ncol(trainData[[predictor]]) * knot_pct),
  ...
)
```

Arguments

`trainData`	data.frame, training data including a response variable and a matrix of tabulated functional predictor. If the functional predictor is the quantile function, then `trainData` is preferably the returned object of sampleQp.
`testData`	data.frame, test data including a response variable and a matrix of tabulated functional predictor. The number of values in the grid for tabulating functional predictor does not need to be the same as that number for the training data. If the functional predictor is the quantile function, then `testData` is preferably the returned object of sampleQp.
`trainArgs`	numeric vector
`testArgs`	numeric vector
`response`	language of the response variable. All response variable types supported by **gam** package are allowed, including continuous, binary or survival outcome.
`family`	family object specifying the distribution and link to use in gam
`predictor`	character scalar, name of the matrix column variable in `trainData` and `testData`, with each row representing the tabulated, on a common grid, functional predictor values for each subject. If the functional predictor is the quantile function, then the number of columns in the matrix column identified by `predictor` equals to the number of quantiles used.
`log`	logical scalar, whether to perform log transformation on quantiles (default `FALSE`).
`knot_pct`	numeric scalar, percentage of the column dimension of the matrix column `predictor`, to be used as `knot.value`. Default is 40%. If `knot.value` is provided by the end-user, then `knot_pct` is not used.
`knot.value`	integer scalar, number of knots (i.e., parameter `k` in the spline smooth function `s`) used in gam function. Default is the ceiling of `knot_pct` of the column dimension of the matrix column `predictor`.
`...`	additional parameters, currently not in use

Details

The Functional Regression Index (FRindex) is defined as the integral of the functional predictor multiplied by common weight function. The weight function is proportional to the functional coefficient in the functional regression model for a given response variable fitted to training data. The FRindex values computed for the training data and test data, if provided.

Value

FRindex returns a data.frame with the training data and FRindex, and the vectors of weights on the grid of the functional predictor

References

J. O. Ramsay, B. W. Silverman (2005). Functional Data Analysis, ed 2. Springer New York, NY doi:10.1007/b98888

Cui, E., Crainiceanu, C. M., & Leroux, A. (2021). Additive Functional Cox Model. Journal of Computational and Graphical Statistics. 2021;30(3):780-793. doi:10.1080/10618600.2020.1853550

Gellar, J. E., Colantuoni, E., Needham, D. M., & Crainiceanu, C. M. (2015). Cox regression models with functional covariates for survival data. Statistical Modelling, 15(3), 256-278. doi:10.1177/1471082X14565526

Examples

```
pt = unique(Ki67$PATIENT_ID)
length(pt) # 622
train = subset(Ki67, PATIENT_ID %in% pt[1:500])
test = subset(Ki67, PATIENT_ID %in% pt[501:622])
train_Qps = sampleQp(data = train, subjID = 'PATIENT_ID',
  exclude = c('tissueID','inner_x','inner_y'), Qpredictor = 'Marker')
test_Qps = sampleQp(data = test, subjID = 'PATIENT_ID',
  exclude = c('tissueID','inner_x','inner_y'), Qpredictor = 'Marker')

if (FALSE) { # masked to save time
FRQI = FRindex(trainData = train_Qps, testData = test_Qps,
  response = Surv(RECFREESURV_MO, RECURRENCE), predictor = 'Marker', log = TRUE)
set.seed(1)
mod = BBC_dichotom(Surv(RECFREESURV_MO, RECURRENCE) ~ NodeSt + Tstage, data = FRQI$trainData,
  contX = 'FRindex_std', R = 100)
}

if (FALSE) { # masked to save time
FRindex(trainData = train_Qps, testData = test_Qps, response = RECURRENCE)
FRindex(trainData = train_Qps, testData = test_Qps, response = RECFREESURV_MO)

set.seed(1)
mod = BBC_dichotom(Surv(RECFREESURV_MO, RECURRENCE) ~ 1, data = FRQI$trainData,
  contX = 'FRindex_std', R = 200)
summary(mod)
names(attributes(mod))
attr(mod, 'median_optimism')
}
```

`model_dichotom` *Model with Dichotomized Predictors*

Description

fit Cox proportional hazard model, logistic regression, or linear regression with dichotomized marker and/or other covariates built as formula in BBC_dichotom

Usage

```
model_dichotom(
  formula,
  data,
  contX,
```

```
  dQ = dichotom_int(edp = eval(formula[[2L]], envir = data), contX = data[[contX]])
)
```

Arguments

formula	same formula in BBC_dichotom
data	data.frame
contX	see BBC_dichotom
dQ	returned object from dichotom_int

Value

model_dichotom returns a numeric vector of the concatenated coefficients.

---

optQp *Optimal Quantile Predictor*

Description

Wrapper of eval_dichotom.

To create a data.frame with designated number of optimal quantiles returned from eval_dichotom in addition to subjID, Surv endpoint and numeric, character, or logical predictors

Usage

```
optQp(x, n = 3L, ...)
```

Arguments

x	eval_dichotom object
n	integer scalar, number of optimal quantiles to be printed
...	additional parameters, not currently in use

Value

optQp returns a data.frame with designated number of optimal quantiles

Examples

```
# see ?BBC_dichotom
```

print.eval_dichotom *Print eval_dichotom Object*

Description

To print the optimal quantile that has the largest absolute value of coefficients for a given Surv outcome, logical, or numeric outcome

Usage

```
## S3 method for class 'eval_dichotom'
print(x, n = 3L, ...)
```

Arguments

x	eval_dichotom object
n	integer scalar, number of optimal quantiles to be printed
...	additional parameters, currently not in use

Details

print.eval_dichotom simply calls summary.eval_dichotom and print three largest absolute value of coefficients, hazard ratio or odds ratio, and threshold

Value

print.eval_dichotom does not have a returned value.

sampleQp *Cluster-Specific Sample Quantiles*

Description

This function calculates vectors of sample quantiles in each independent cluster of observations (sample or subject) Equidistant probabilities between user provided p_min and p_max are used (including both ends).

Usage

```
sampleQp(
  data,
  subjID = "PATIENT_ID",
  Qpredictor = "Marker",
  include = setdiff(names(data), y = c(Qpredictor)),
  exclude = NULL,
  from = 0.01,
  to = 0.99,
  by = 0.01,
  type = 7,
  ...
)
```

Arguments

data	data.frame
subjID	character scalar, column name of the subject/patient index in data
Qpredictor	character scalar, column name of the predictor variable
include	character vector, predictors to be included in output data set
exclude	character vector, predictors to be excluded from output data set
from, to, by	see seq a sequence of probabilities with starting, ending values, and interval to calculate corresponding quantiles
type	integer scalar, type of the formula for quantiles,see quantile
...	additional parameters, currently not in use

Details

sampleQp calculates N_p sample quantiles in each independent cluster of observations defined by subjID or sampleID. Sample quantiles are stored in N_p columns for $Q(p)$, $p = 1, \cdots, Np$ Additional subject-level predictors may be designated to be kept in the output data set. Observation-level predictors should be excluded from the input data set.

Value

sampleQp returns a data.frame, with aggregated Surv endpoint and numeric, character, or logical predictors, in addition to a matrix of quantiles.

Examples

```
Ki67_Qps = sampleQp(data = Ki67, subjID = 'PATIENT_ID',
  exclude = c('tissueID','inner_x','inner_y'), Qpredictor = 'Marker')
head(Ki67_Qps)
sapply(Ki67_Qps, FUN = class)
head(Ki67_Qps$Marker)
```

summary.eval_dichotom *Summary Information of eval_dichotom Object*

Description

To summarize eval_dichotom object

Usage

```
## S3 method for class 'eval_dichotom'
summary(object, FUN = median, ...)
```

Arguments

object	eval_dichotom object
FUN	summarizing function, either median (default) or mean
...	additional parameters, currently not in use

Details

summary.eval_dichotom present effect sizes (absolute value of log hazard ratio, log odds ratio, or coefficient), corresponding hazard ratio or odds ratio, and threshold to dichotomize quantile. And then sort by abs.coef

Value

summary.eval_dichotom returns a data.frame with three (3) columns

abs.coef effect sizes (absolute value of log hazard ratio, log odds ratio, or coefficient)

HR corresponding hazard ratio or odds ratio

threshold threshold for dichotomizing the quantile

See Also

summary

Index