Package ‘CARRoT’

January 20, 2025

Title Predicting Categorical and Continuous Outcomes Using One in Ten Rule

Version 3.0.2

Description Predicts categorical or continuous outcomes while concentrating on a number of key points. These are Cross-validation, Accuracy, Regression and Rule of Ten or ``one in ten rule'' (CARRoT), and, in addition to it R-squared statistics, prior knowledge on the dataset etc. It performs the cross-validation specified number of times by partitioning the input into training and test set and fitting linear/multinomial/binary regression models to the training set. All regression models satisfying chosen constraints are fitted and the ones with the best predictive power are given as an output. Best predictive power is understood as highest accuracy in case of binary/multinomial outcomes, smallest absolute and relative errors in case of continuous outcomes. For binary case there is also an option of finding a regression model which gives the highest AUROC (Area Under Receiver Operating Curve) value. The option of parallel toolbox is also available. Methods are described in Peduzzi et al. (1996) <doi:10.1016/S0895-4356(96)00236-3> , Rhemtulla et al. (2012) <doi:10.1037/a0029315>, Riley et al. (2018) <doi:10.1002/sim.7993>, Riley et al. (2019) <doi:10.1002/sim.7992>.

Depends R (>= 3.4.0)

License GPL-2

Encoding UTF-8

Imports stats,utils,nnet,doParallel,Rdpack,parallel,foreach

RoxygenNote 7.2.3

RdMacros Rdpack

NeedsCompilation yes

Author Alina Bazarova [aut, cre],
Marko Raseta [aut]

Maintainer Alina Bazarova <al.bazarova@fz-juelich.de>

Repository CRAN

Date/Publication 2023-10-13 20:20:06 UTC

Contents

AUC *Area Under the Curve*

Description

Function enables efficient computation of area under receiver operating curve (AUC). Source: https://stat.ethz.ch/pipermail/r-help/2005-September/079872.html

Usage

```
AUC(probs, class)
```

Arguments

probs	probabilities
class	outcomes

Value

A value for AUC

Examples

```
AUC(runif(100,0,1),rbinom(100,1,0.3))
```

av_out *Averaging out the predictive power*

Description

Function which averages out the predictive power over all cross-validations

Usage

```
av_out(preds,crv,k)
```

Arguments

preds	An M x crvN matrix consisting of crv horizontally concatenated M x N matrices. These M x N matrices are the matrices of predictive powers for all feasible regressions (M is maximum feasible number of variables included in a regression, N is the maximum feasible number of regressions of the fixed size; the row index indicates the number of variables included in a regression)
crv	number of cross-validations
k	size of the test set for which the predictions are made

Value

Returns an M x N matrix of average predictive powers where M is maximum feasible number of variables included in a regression, N is the maximum feasible number of regressions of the fixed size; the row index indicates the number of variables included in a regression

Examples

```
#creating a matrix of predictive powers

preds<-cbind(matrix(runif(40,1,4),ncol=10),matrix(runif(40,1.5,4),ncol=10))
preds<-cbind(preds,matrix(runif(40,1,3.5),ncol=10))

#running the function

av_out(preds,3,5)
```

comb *Combining in a list*

Description

Function for combining outputs in a list

Usage

```
comb(...)
```

Arguments

...	an argument of mapply used by this function

See Also

Function mapply

Examples

```
#array of numbers to be separated in a list

a<-1:4

#running the function

comb(a)
```

compute_max_length *Maximum number of the regressions*

Description

Function which computes the maximum number of regressions with fixed number of variables based on the rule of thumb

Usage

```
compute_max_length(vari_col,k,c,we,minx,maxx,st)
```

Arguments

`vari_col`	number of predictors
`k`	maximum weight of the predictors
`c`	array of all indices of the predictors
`we`	array of weights of the predictors. Continuous or categorical numerical variable with more then 5 categories has weight 1, otherwise it has weight `n-1` where `n` is the number of categories
`minx`	minimum number of predictors, 1 by default
`maxx`	maximum number of predictors, total number of variables by default
`st`	a subset of predictors to be always included into a predictive model

Value

Integer correponding to maximum number of regressions of the same size

References

Peduzzi P, Concato J, Kemper E, Holford TR, Feinstein AR (1996). "A simulation study of the number of events per variable in logistic regression analysis." *Journal of Clinical Epidemiology*, **49**(12), 1373-1379. ISSN 0895-4356, doi:10.1016/S08954356(96)002363, `http://dx.doi.org/10.1016/S0895-4356(96)00236-3`.

Rhemtulla M, Brosseau-Liard PÉ, Savalei V (2012). "When can categorical variables be treated as continuous?: A comparison of robust continuous and categorical SEM estimation methods under suboptimal conditions." *Psychological Methods*, **17**(3), 354-373. doi:10.1037/a0029315.

See Also

Function uses `combn`

Examples

```
compute_max_length(4,40,1:4,c(1,1,2,1))
```

`compute_max_weight` *Maximum feasible weight of the predictors*

Description

Function which computes maximal weight (multiplied by the corresponding EPV rule) of a regression according to the rule of thumb applied to the outcome variable. Weight of a regression equals the sum of weights of its predictors.

Usage

```
compute_max_weight(outi,mode)
```

Arguments

outi	set of outcomes
mode	indicates the mode: 'linear' (linear regression), 'binary' (logistic regression), 'multin' (multinomial regression)

Details

For continuous outcomes it equals sample size divided by 10, for multinomial it equals the size of the smallest category divided by 10

Value

returns an integer value of maximum allowed weight multiplied by 10

References

Peduzzi P, Concato J, Kemper E, Holford TR, Feinstein AR (1996). "A simulation study of the number of events per variable in logistic regression analysis." *Journal of Clinical Epidemiology*, **49**(12), 1373-1379. ISSN 0895-4356, doi:10.1016/S08954356(96)002363, http://dx.doi.org/10.1016/S0895-4356(96)00236-3.

Examples

```
#continuous outcomes

compute_max_weight(runif(100,0,1),'linear')

#binary outcomes

compute_max_weight(rbinom(100,1,0.4),'binary')
```

---

compute_weights *Weights of predictors*

Description

Function which computes the weight of each predictor according to the rules of thumb and outputs it into corresponding array

Usage

```
compute_weights(vari_col, vari)
```

Arguments

vari_col	number of predictors
vari	set of predictors

Details

Continuous or categorical numerical variable with more then 5 categories has weight 1, otherwise it has weight n-1 where n is the number of categories

Value

Returns an array of weights of the size vari_col

References

Peduzzi P, Concato J, Kemper E, Holford TR, Feinstein AR (1996). "A simulation study of the number of events per variable in logistic regression analysis." *Journal of Clinical Epidemiology*, **49**(12), 1373-1379. ISSN 0895-4356, doi:10.1016/S08954356(96)002363, http://dx.doi.org/10.1016/S0895-4356(96)00236-3.

Rhemtulla M, Brosseau-Liard PÉ, Savalei V (2012). "When can categorical variables be treated as continuous?: A comparison of robust continuous and categorical SEM estimation methods under suboptimal conditions." *Psychological Methods*, **17**(3), 354-373. doi:10.1037/a0029315.

Examples

```
#creating data-set with for variables

a<-matrix(NA,nrow=100,ncol=4)

#binary variable

a[,1]=rbinom(100,1,0.3)

#continuous variable

a[,2]=runif(100,0,1)

#categorical numeric with les than 5 categories

a[,3]=t(rmultinom(100,1,c(0.2,0.3,0.5)))%*%c(1,2,3)

#categorical numeric with 5 categories

a[,4]=t(rmultinom(100,1,c(0.2,0.3,0.3,0.1,0.1)))%*%c(1,2,3,4,5)

#running the function

compute_weights(4,a)
```

cross_val *Cross-validation run*

Description

Function running a single cross-validation by partitioning the data into training and test set

Usage

```
cross_val(
  vari,
  outi,
  c,
  rule,
  part,
  l,
  we,
  vari_col,
  preds,
  mode,
  cmode,
  predm,
  cutoff,
  objfun,
  minx = 1,
  maxx = NULL,
  nr = NULL,
  maxw = NULL,
  st = NULL,
  corr = 1,
  Rsq = F,
  marg = 0,
  n_tr,
  preds_tr
)
```

Arguments

`vari`	set of predictors
`outi`	array of outcomes
`c`	set of all indices of the predictors
`rule`	an Events per Variable (EPV) rule, defaults to 10
`part`	indicates partition of the original data-set into training and test set in a proportion `(part-1):1`
`l`	number of observations

`we`	weights of the predictors
`vari_col`	overall number of predictors
`preds`	array to write predictions for the test split into, intially empty
`mode`	`'binary'` (logistic regression), `'multin'` (multinomial regression)
`cmode`	`'det'` or `''`; `'det'` always predicts the more likely outcome as determined by the odds ratio; `''` predicts certain outcome with probability corresponding to its odds ratio (more conservative). Option available for multinomial/logistic regression
`predm`	`'exact'` or `''`; for logistic and multinomial regression; `'exact'` computes how many times the exact outcome category was predicted, `''` computes how many times either the exact outcome category or its nearest neighbour was predicted
`cutoff`	cut-off value for logistic regression
`objfun`	`'roc'` for maximising the predictive power with respect to AUC, `'acc'` for maximising predictive power with respect to accuracy.
`minx`	minimum number of predictors to be included in a regression, defaults to 1
`maxx`	maximum number of predictors to be included in a regression, defaults to maximum feasible number according to one in ten rule
`nr`	a subset of the data-set, such that `1/part` of it lies in the test set and `1-1/part` is in the training set, defaults to empty set
`maxw`	maximum weight of predictors to be included in a regression, defaults to maximum weight according to one in ten rule
`st`	a subset of predictors to be always included into a predictive model,defaults to empty set
`corr`	maximum correlation between a pair of predictors in a model
`Rsq`	whether R-squared statistics constrained is introduced
`marg`	margin of error for R-squared statistics constraint
`n_tr`	size of the training set
`preds_tr`	array to write predictions for the training split into, intially empty

Value

`regr`	An M x N matrix of sums of the absolute errors for each element of the test set for each feasible regression. M is maximum feasible number of variables included in a regression, N is the maximum feasible number of regressions of the fixed size; the row index indicates the number of variables included in a regression. Therefore each row corresponds to results obtained from running regressions with the same number of variables and columns correspond to different subsets of predictors used.
`regrr`	An M x N matrix of sums of the relative errors for each element of the test set (only for `mode = 'linear'`) for each feasible regression. M is maximum feasible number of variables included in a regression, N is the maximum feasible number of regressions of the fixed size; the row index indicates the number of variables included in a regression. Therefore each row corresponds to results obtained from running regressions with the same number of variables and columns correspond to different subsets of predictors used.

nvar	Maximum feasible number of variables in the regression
emp	An accuracy of always predicting the more likely outcome as suggested by the training set (only for mode = 'binary' and objfun = 'acc')

In regr and regrr NA values are possible since for some numbers of variables there are fewer feasible regressions than for the others.

See Also

Uses compute_max_weight, sum_weights_sub, make_numeric_sets, get_predictions_lin, get_predictions, get_probabilities, AUC, combn

Examples

```
#creating variables

vari<-matrix(c(1:100,seq(1,300,3)),ncol=2)

#creating outcomes

out<-rbinom(100,1,0.3)

#creating array for predictions

pr<-array(NA,c(2,2))

pr_tr<-array(NA,c(2,2))

#passing set of the inexes of the predictors

c<-c(1:2)

#passing the weights of the predictors

we<-c(1,1)

#setting the mode

m<-'binary'

#running the function

cross_val(vari,out,c,10,10,100,we,2,pr,m,'det','exact',0.5,'acc',nr=c(1,4),n_tr=90,preds_tr=pr_tr)
```

---

cub *Three-way interactions and squares*

Description

Function transforms a set of predictors into a set of predictors, their squares, pairwise interactions, cubes and three-way interactions

Usage

```
cub(A, n = 1000)
```

Arguments

A	set of predictors
n	first n predictors, whose interactions with the rest should be taken into account, defaults to all of the predictors

Value

Returns the predictors including their squares, pairwise interactions, cubes and three-way interactions

Examples

```
cub(cbind(1:100,rnorm(100),runif(100),rnorm(100,0,2)))
```

---

find_int *Finding the interacting terms based on the index*

Description

Function transforms an index of an array of two- or three-way interactions into two or three indices corresponding to the interacting variables

Usage

```
find_int(ind,N)
```

Arguments

ind	index to transform
N	number of interacting variables

Value

Returns two or three indices corredsponding to a combination of variables written under the given index

Examples

```
find_int(28,9)
```

find_sub *Finds certain subsets of predictors*

Description

Reorders the columns of matrix a according to the ordered elements of array s

Usage

```
find_sub(a,s,j,c,st)
```

Arguments

a	A j x N matrix, containing all possible subsets (N overall) of the size j of predictors' indices.
s	array of numbers of the size N
j	number of rows in a
c	array of all indices of the predictors
st	a subset of predictors to be always included into a predictive model

Value

Returns a submatrix of matrix a which consits of columns determined by the input array s

Examples

```
#all two-element subsets of 1:3

a<-combn(3,2)
s<-c(3,2,3)

find_sub(a,s,2,1:3)
```

get_indices *Best regression*

Description

Function which identifies regressions with the highest predictive power

Usage

```
get_indices(predsp,nvar,c,we,st,minx)
```

Arguments

predsp	An M x N matrix of averaged out predictive power values. M is maximum feasible number of variables included in a regression, N is the maximum feasible number of regressions of the fixed size; the row index indicates the number of variables included in a regression.
nvar	array of maximal number of variables for each cross-validation
c	array of all indices of the prediction variables
we	array of all weights of the prediction variables
st	a subset of predictors to be always included into a predictive model
minx	minimum number of predictors, defaults to 1

Value

A list of arrays which contain indices of the predictors corresponfing to the best regressions

See Also

Uses sum_weights_sub, find_sub, combn

Examples

```
#creating a set of averaged out predictive powers

predsp<-matrix(NA,ncol=3,nrow=3)

predsp[1,]=runif(3,0.7,0.8)
predsp[2,]=runif(3,0.65,0.85)
predsp[3,1]=runif(1,0.4,0.5)

#running the function

get_indices(predsp,c(3,3,3),1:3,c(1,1,1))
```

---

get_predictions *Predictions for multinomial regression*

Description

Function which makes a prediction for multinomial/logistic regression based on the given cut-off value and probabilities.

Usage

```
get_predictions(p,k,cutoff,cmode,mode)
```

Arguments

p	probabilities of the outcomes for the test set given either by an array (logistic regression) or by a matrix (multinomial regression)
k	size of the test set
cutoff	cut-off value of the probability
cmode	'det' or ''; 'det' always predicts the more likely outcome as determined by the odds ratio; '' predicts certain outcome with probability corresponding to its odds ratio (more conservative). Option available for multinomial/logistic regression
mode	'binary' (logistic regression), 'multin' (multinomial regression)

Value

Outputs the array of the predictions of the size of p.

See Also

Uses rbinom, rmultinom

Examples

```
#binary mode

get_predictions(runif(20,0.4,0.6),20,0.5,'det','binary')

#creating a data-set for multinomial mode

p1<-runif(20,0.4,0.6)
p2<-runif(20,0.1,0.2)
p3<-1-p1-p2

#running the function

get_predictions(matrix(c(p1,p2,p3),ncol=3),20,0.5,'det','multin')
```

---

get_predictions_lin *Predictions for linear regression*

Description

Function which runs a linear regression on a training set, computes predictions for the test set

Usage

```
get_predictions_lin(trset,testset,outc,k,n_tr,p,Rsq,Rsq_v,marg)
```

Arguments

trset	values of predictors on the training set
testset	values of predictors on the test set
outc	values of predictors on the training set
k	length of the test set
n_tr	size of the training set
p	weight of the model
Rsq	whether the R-squared statistics constraint is introduced
Rsq_v	value of R-squared statistics on the training spli of the data
marg	margin of error for R-squared statistics constraint

Value

An array of continous variables of the length equal to the size of a `testset`

See Also

Function uses function `lsfit` and `coef`

Examples

```
trset<-matrix(c(rnorm(90,2,4),runif(90,0,0.5),rbinom(90,1,0.5)),ncol=3)

testset<-matrix(c(rnorm(10,2,4),runif(10,0,0.5),rbinom(10,1,0.5)),ncol=3)

get_predictions_lin(trset,testset,runif(90,0,1),10)
```

`get_probabilities` *Probabilities for multinomial regression*

Description

Function which computes probabilities of outcomes on the test set by applying regression parameters inferred by a run on the training set. Works for logistic or multinomial regression

Usage

```
get_probabilities(trset,testset,outc,mode,Rsq,p,n_tr)
```

Arguments

`trset`	values of predictors on the training set
`testset`	values of predictors on the test set
`outc`	values of outcomes on the training set
`mode`	`'binary'` (logistic regression) or `'multin'` (multinomial regression)
`Rsq`	whether R-squared statistics constrained is introduced
`p`	weight of the model
`n_tr`	size of the training set

Details

In binary mode this function computes the probabilities of the event '0'. In multinomial mode computes the probabilities of the events '0','1',...,'N-1'.

Value

Probabilities of the outcomes. In `'binary'` mode returns an array of the size of the number of observations in a testset. In `'multin'` returns an M x N matrix where M is the size of the number of observations in a testset and N is the number of unique outcomes minus 1.

See Also

Function uses `multinom` and `coef`

Examples

```
trset<-matrix(c(rbinom(70,1,0.5),runif(70,0.1)),ncol=2)

testset<-matrix(c(rbinom(10,1,0.5),runif(10,0.1)),ncol=2)

get_probabilities(trset,testset,rbinom(70,1,0.6),'binary')
```

`make_numeric` *Turning a non-numeric variable into a numeric one*

Description

Function which turns a single categorical (non-numeric) variable into a numeric one (or several) by introducing dummy '0'/'1' variables.

Usage

```
make_numeric(vari, outcome, ra,mode)
```

Arguments

- `vari` array of values to be transformed
- `outcome` TRUE/FALSE indicates whether the variable `vari` is an outcome (TRUE) or a predictor (FALSE)
- `ra` indices of the input array `vari` which indicate which values will be transformed
- `mode` `'binary'` (logistic regression), `'multin'` (multinomial regression)

Details

This function is essentially a standard way to turn categorical non-numeric variables into numeric ones in order to run a regression

Value

Returned value is an M x N matrix where M is the length of the input array of indices `ra` and N is `length(vari)-1`.

Examples

```
#creating a non-numeric set

a<-t(rmultinom(100,1,c(0.2,0.3,0.5)))%*%c(1,2,3)

a[a==1]='red'
a[a==2]='green'
a[a==3]='blue'

#running the function

make_numeric(a,FALSE,sample(1:100,50),"linear")

make_numeric(a,TRUE,sample(1:100,50))
```

`make_numeric_sets` *Transforming the set of predictors into a numeric set*

Description

Function which turns a set of predictors containing non-numeric variables into a fully numeric set

Usage

```
make_numeric_sets(a,ai,k,vari,ra,l,mode)
```

Arguments

a	An M x N matrix, containing all possible subsets (N overall) of the size M of predictors' indices; therefore each column of a defines a unique subset of the predictors
ai	array of indices of the array a
k	index of the array ai
vari	set of all predictors
ra	array of sample indices of vari
l	size of the sample
mode	'binary' (logistic regression), 'multin' (multinomial regression)

Details

Function transforms the whole set of predictors into a numeric set by consecutively calling function make_numeric for each predictor

Value

Returns a list containing two objects: tr and test

tr	training set transformed into a numeric one
test	test set transformed into a numeric one

See Also

make_numeric

Examples

```
#creating a categorical numeric variable

a<-t(rmultinom(100,1,c(0.2,0.3,0.5)))%*%c(1,2,3)

#creating an analogous non-numeric variable

c<-array(NA,100)
c[a==1]='red'
c[a==2]='green'
c[a==3]='blue'

#creating a data-set

b<-data.frame(matrix(c(a,rbinom(100,1,0.3),runif(100,0,1)),ncol=3))

#making the first column of the data-set non-numeric

b[,1]=data.frame(c)
```

```
#running the function

make_numeric_sets(combn(3,2),1:3,1,b,sample(1:100,60),100,"binary")
```

`quadr` *Pairwise interactions and squares*

Description

Function transforms a set of predictors into a set of predictors, their squares and pairwise interactions

Usage

```
quadr(A, n = 1000)
```

Arguments

A	set of predictors
n	first n predictors, whose interactions with the rest should be taken into account, defaults to all of the predictors

Value

Returns the predictors including their squares and pairwise interactions

Examples

```
quadr(cbind(1:100,rnorm(100),runif(100),rnorm(100,0,2)))
```

`regr_ind` *Indices of the best regressions*

Description

One of the two main functions of the package. Identifies the predictors included into regressions with the highest average predictive power

Usage

```
regr_ind(
  vari,
  outi,
  crv,
  cutoff = NULL,
  part = 10,
  mode,
  cmode = "det",
  predm = "exact",
  objfun = "acc",
  parallel = FALSE,
  cores,
  minx = 1,
  maxx = NULL,
  nr = NULL,
  maxw = NULL,
  st = NULL,
  rule = 10,
  corr = 1,
  Rsq = F,
  marg = 0
)
```

Arguments

`vari`	set of predictors
`outi`	array of outcomes
`crv`	number of cross-validations
`cutoff`	cut-off value for mode `'binary'`
`part`	for each cross-validation partitions the dataset into training and test set in a proportion `(part-1):part`
`mode`	`'binary'` (logistic regression), `'multin'` (multinomial regression)
`cmode`	`'det'` or `''`; `'det'` always predicts the more likely outcome as determined by the odds ratio; `''` predicts certain outcome with probability corresponding to its odds ratio (more conservative). Option available for multinomial/logistic regression
`predm`	`'exact'` or `''`; for logistic and multinomial regression; `'exact'` computes how many times the exact outcome category was predicted, `''` computes how many times either the exact outcome category or its nearest neighbour was predicted
`objfun`	`'roc'` for maximising the predictive power with respect to AUC, available only for `mode='binary'`; `'acc'` for maximising predictive power with respect to accuracy.
`parallel`	TRUE if using parallel toolbox, FALSE if not. Defaults to FALSE
`cores`	number of cores to use in case of parallel=TRUE

minx	minimum number of predictors to be included in a regression, defaults to 1
maxx	maximum number of predictors to be included in a regression, defaults to maximum feasible number according to one in ten rule
nr	a subset of the data-set, such that `1/part` of it lies in the test set and `1-1/part` is in the training set, defaults to empty set. This is to ensure that elements of this subset are included both in the training and in the test set.
maxw	maximum weight of predictors to be included in a regression, defaults to maximum weight according to one in ten rule
st	a subset of predictors to be always included into a predictive model,defaults to empty set
rule	an Events per Variable (EPV) rule, defaults to 10'
corr	maximum correlation between a pair of predictors in a model
Rsq	whether the R-squared statistics constraint is introduced
marg	margin of error for R-squared statistics constraint

Value

Prints the best predictive power provided by a regression, predictive accuracy of the empirical prediction (value of `emp` computed by `cross_val` for logistic and linear regression). Returns indices of the predictors included into regressions with the highest predictive power written in a list. For `mode='linear'` outputs a list of two lists. First list corresponds to the smallest absolute error, second corresponds to the smallest relative error

See Also

Uses `compute_weights`, `make_numeric`, `compute_max_weight`, `compute_weights`, `compute_max_length`, `cross_val`,`av_out`, `get_indices`

Examples

```
#creating variables for linear regression mode

variables_lin<-matrix(c(rnorm(56,0,1),rnorm(56,1,2)),ncol=2)

#creating outcomes for linear regression mode

outcomes_lin<-rnorm(56,2,1)

#running the function

regr_ind(variables_lin,outcomes_lin,100,mode='linear',parallel=TRUE,cores=2)

#creating variables for binary mode

vari<-matrix(c(1:100,seq(1,300,3)),ncol=2)

#creating outcomes for binary mode
```

```
out<-rbinom(100,1,0.3)

#running the function

regr_ind(vari,out,20,cutoff=0.5,part=10,mode='binary',parallel=TRUE,cores=2,nr=c(1,10,20),maxx=1)
```

---

sum_weights_sub *Cumulative weights of the predictors' subsets*

Description

Function which computes the sum of predictors' weights for each subset containing a fixed number of predictors

Usage

```
sum_weights_sub(a,m,we,st)
```

Arguments

a	an m x N matrix, containing all possible subsets (N overall) of the size m of predictors' indices; therefore each column of a defines a unique subset of the predictors
m	number of elements in each subset of indices
we	array of weights of the predictors
st	a subset of predictors to be always included into a predictive model

Value

Returns an array of weights for predictors defined by each colun of the matrix a

Examples

```
#all two-element subsets of the set 1:3

a<-combn(3,2)

sum_weights_sub(a,2,c(1,2,1))
```

Index