Package 'simIReff'

May 9, 2026

Type Package

Title Stochastic Simulation for Information Retrieval Evaluation: Effectiveness Scores

Version 1.0

Description Provides tools for the stochastic simulation of effectiveness scores to mitigate data-related limitations of Information Retrieval evaluation research, as described in Urbano and Nagler (2018) <doi:10.1145/3209978.3210043>. These tools include: fitting, selection and plotting
distributions to model system effectiveness, transformation towards a prespecified expected value, proxy to fitting of copula models based on these distributions, and simulation of new evaluation data from these distributions and copula models.

BugReports https://github.com/julian-urbano/simIReff/issues

URL https://github.com/julian-urbano/simIReff/

Depends R (>= 3.4)

Imports stats, graphics, MASS, rvinecopulib (>= 0.2.8.1.0), truncnorm, bde, ks, np, extraDistr

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Author Julián Urbano [aut, cre], Thomas Nagler [ctb]

Maintainer Julián Urbano <urbano.julian@gmail.com>

Repository CRAN

Date/Publication 2018-06-15 15:26:32 UTC

Contents

`simIReff-package` `simIReff`*: Stochastic Simulation for Information Retrieval Evaluation: Effectiveness Scores*

Description

Provides tools for the stochastic simulation of effectiveness scores to mitigate data-related limitations of Information Retrieval evaluation research. These tools include:

- Fitting of continuous and discrete distributions to model system effectiveness.
- Plotting of effectiveness distributions.
- Selection of distributions best fitting to given data.
- Transformation of distributions towards a prespecified expected value.
- Proxy to fitting of copula models based on these distributions.
- Simulation of new evaluation data from these distributions and copula models.

Author(s)

Maintainer: Julián Urbano <urbano.julian@gmail.com>

Other contributors:

- Thomas Nagler <thomas.nagler@tum.de> [contributor]

References

J. Urbano and T. Nagler. (2018). Stochastic Simulation of Test Collections: Evaluation Scores. ACM SIGIR.

See Also

Useful links:

- https://github.com/julian-urbano/simIReff/
- Report bugs at https://github.com/julian-urbano/simIReff/issues

Examples

```
## Fit a marginal AP distribution and simulate new data
x <- web2010ap[,10] # sample AP scores of a system
e <- effContFitAndSelect(x, method = "BIC") # fit and select based on log-likelihood
plot(e) # plot pdf, cdf and quantile function
e$mean # expected value
y <- reff(50, e) # simulation of 50 new topics

## Transform the distribution to have a pre-specified expected value
e2 <- effTransform(e, mean = .14) # transform for expected value of .14
plot(e2)
e2$mean # check the result

## Build a copula model of two systems
d <- web2010ap[,2:3] # sample AP scores
e1 <- effCont_norm(d[,1]) # force the first margin to follow a truncated gaussian
e2 <- effCont_bks(d[,2]) # force the second margin to follow a beta kernel-smoothed
cop <- effcopFit(d, list(e1, e2)) # copula
y <- reffcop(1000, cop) # simulation of 1000 new topics
c(e1$mean, e2$mean) # expected means
colMeans(y) # observed means

## Modify the model to both systems have the same distribution
cop2 <- cop # copy the model
cop2$margins[[2]] <- e1 # modify 2nd margin
y <- reffcop(1000, cop2) # simulation of 1000 new topics
colMeans(y) # observed means

## Automatically build a gaussian copula to many systems
d <- web2010p20[,1:20] # sample P@20 data from 20 systems
effs <- effDiscFitAndSelect(d, support("p20")) # fit and select margins
cop <- effcopFit(d, effs, family_set = "gaussian") # fit copula
```

```
y <- reffcop(1000, cop) # simulate new 1000 topics

# compare observed vs. expected mean
E <- sapply(effs, function(e) e$mean)
E.hat <- colMeans(y)
plot(E, E.hat)
abline(0:1)

# compare observed vs. expected variance
Var <- sapply(effs, function(e) e$var)
Var.hat <- apply(y, 2, var)
plot(Var, Var.hat)
abline(0:1)

# compare distributions
o <- order(colMeans(d))
boxplot(d[,o])
points(colMeans(d)[o], col = "red", pch = 4) # plot means
boxplot(y[,o])
points(colMeans(y)[o], col = "red", pch = 4) # plot means
```

`eff` *Effectiveness Distributions*

Description

Density, distribution function, quantile function and random generation for an effectiveness distribution.

Usage

```
deff(x, .eff)

peff(q, .eff)

qeff(p, .eff)

reff(n, .eff)
```

Arguments

`x`, `q`	vector of quantiles.
`.eff`	the `eff` object representing the effectiveness distribution.
`p`	vector of probabilities.
`n`	number of observations.

Value

`deff` gives the density, `peff` gives the distribution function, `qeff` gives the quantile function, and `reff` generates random variates.

See Also

`effCont` for continuous distributions, and `effDisc` for discrete distributions.

Examples

```
# sample distribution from AP scores
e <- effCont_beta(web2010ap[,1])
# pdf integrates to 1
integrate(deff, lower = 0, upper = 1, .eff = e)
# qeff (quantile) is the inverse of peff (cumulative)
qeff(peff(.2, e), e)
# random generation of 100 scores
r <- reff(100, e)
```

---

eff.cont-class *Class* `eff.cont`

Description

This is the base S3 class for all continuous effectiveness distributions, which is itself a subclass of `eff`. Function `effCont_new` is the constructor of the class.

Usage

```
effCont_new(mean, var, df, x = NULL)
```

Arguments

`mean`	the expected value of the distibution.
`var`	the variance of the distribution.
`df`	the effective degrees of freedom of the distribution.
`x`	the sample of effectiveness scores used to fit the distribution. Defaults to `NULL`.

Details

A new distribution family is expected to build new objects through this constructor, and they must implement methods `deff`, `peff`, `qeff` and `reff`.

Value

an object of class `eff.cont`, with the following components:

mean	the expected value.
var	the variance.
df	the degrees of freedom (effective number of parameters) for model selection.
data	the sample data used to fit the distribution, or NULL if none.
model	a list with the family-specific data.

See Also

effCont for a list of currently implemented distribution families, effContFit to fit distributions, and effCont-helper for helper functions.

For discrete distributions, see eff.disc.

---

eff.disc-class *Class* eff.disc

Description

This is the base S3 class for all discrete effectiveness distributions, which is itself a subclass of eff. Function effDisc_new is the constructor of the class.

Usage

```
effDisc_new(p, support, df, x = NULL)
```

Arguments

p	the values of the distribution function at the support points.
support	the support of the distribution.
df	the effective degrees of freedom of the distribution.
x	the sample of effectiveness scores used to fit the distribution. Defaults to NULL.

Details

A new distribution family is expected to build new objects through this constructor. Default implementations are readily available for methods deff, peff, qeff and reff.

Value

an object of class eff.disc, with the following components:

mean	the expected value.
var	the variance.
df	the degrees of freedom (effective number of parameters) for model selection.
support	the support of the distribution.
data	the sample data used to fit the distribution, or NULL if none.

`model` a list with the family-specific data.

See Also

`effDisc` for a list of currently implemented distribution families, `effDiscFit` to fit distributions, and `effDisc-helper` for helper functions.

For continuous distributions, see `eff.cont`.

`effCont` *Continuous Effectiveness Distributions*

Description

Families to model effectiveness distributions with continuous support. Currently implemented families are:

`effCont_norm`	Truncated Normal.
`effCont_beta`	Beta.
`effCont_nks`	Truncated Kernel-smoothed with Gaussian kernel.
`effCont_bks`	Kernel-smoothed with Beta kernel.

See Also

`effContFit` to fit continuous distributions, and `eff.cont` for the S3 class.

For discrete distributions, see `effDisc`.

`effCont-helper` *Helper functions for continuous effectiveness distributions*

Description

These are functions to help in the creation and use of continuous effectiveness distributions.

Usage

```
cap(x, xmin = 1e-06, xmax = 1 - xmin)

effContMean(qfun, abs.tol = 1e-06, subdivisions = 500)

effContVar(qfun, mu, abs.tol = 1e-06, subdivisions = 500)

effContTrunc(dfun, pfun, qfun, ...)
```

Arguments

x	a sample of effectiveness scores.
xmin	lowest value to cap scores.
xmax	highest value to cap scores.
qfun	a quantile function.
abs.tol	absolute accuracy requested, passed to integrate.
subdivisions	the maximum number of subintervals, passed to integrate.
mu	the expected value of the distribution (see effContMean).
dfun	a density function.
pfun	a distribution function.
...	additional arguments passed to other functions, if any.

Details

cap caps (censor) a variable from below and above.

effContMean computes the expected value of a distribution by numerical integration of the given quantile function.

effContVar computes the variance of a distribution by numerical integration of the given quantile function.

effContTrun computes the density, distribution and quantile functions of the distribution resulting from truncating a given distribution between 0 and 1.

Value

cap: the original vector, but censored.

effContMean: the estimate of the expected value.

effContVar: the estimate of the variance.

effContTrunc: a list with components:

td	the truncated density function.
tp	the truncated distribution function.
tq	the truncated quantile function.

See Also

eff.cont.

Examples

```
cap(c(0, .5, 1))

effContMean(function(p) qnorm(p, mean = 4))
effContMean(function(p) qbeta(p, 1, 2))

effContVar(function(p) qnorm(p, mean = 2, sd = 4), 2)
```

```
effContVar(function(p) qbeta(p, 1, 2), 1/3)

tr <- effContTrunc(dnorm, pnorm, qnorm, mean = .8, sd = .3)
x01 <- seq(0, 1, .01)
plot(x01, tr$d(x01), type = "l")
plot(x01, tr$p(x01), type = "l")
plot(x01, tr$q(x01), type = "l")
```

`effCont_beta` *Continuous Effectiveness as Beta Distribution.*

Description

Fits a Beta distribution to the given sample of scores.

Usage

```
effCont_beta(x)
```

Arguments

x	a sample of effectiveness scores between 0 and 1.

Value

an object of class `eff.cont.beta`, which inherits from `eff.cont`.

See Also

`deff`, `peff`, `qeff` and `reff`.

Examples

```
e <- effCont_beta(web2010ap[,1])
c(e$mean, e$var)
plot(e, plot.data = TRUE)
```

effCont_bks *Continuous Effectiveness as Beta Kernel-smoothed Distribution.*

Description

Fits a bounded kernel-smoothed distribution to the given sample of scores. In particular, the beta kernel by Chen (1999) is used, as in `Chen99Kernel`.

Usage

```
effCont_bks(x)
```

Arguments

x	a sample of effectiveness scores between 0 and 1.

Value

an object of class `eff.cont.bks`, which inherits from `eff.cont`.

References

S.X. Chen (1999). Beta kernel estimators for density functions. Computational Statistics & Data Analysis, 31, 131-145.

See Also

`deff`, `peff`, `qeff` and `reff`.

Examples

```
e <- effCont_bks(web2010ap[,1])
c(e$mean, e$var)
plot(e, plot.data = TRUE)
```

effCont_nks *Continuous Effectiveness as Truncated Gaussian Kernel-smoothed Distribution.*

Description

Fits a kernel-smoothed distribution to the given sample of scores, truncated between 0 and 1, and using a gaussian kernel.

Usage

```
effCont_nks(x)
```

Arguments

x	a sample of effectiveness scores between 0 and 1.

Value

an object of class `eff.cont.nks`, which inherits from `eff.cont`.

See Also

`deff`, `peff`, `qeff` and `reff`.

Examples

```
e <- effCont_nks(web2010ap[,1])
c(e$mean, e$var)
plot(e, plot.data = TRUE)
```

---

effCont_norm *Continuous Effectiveness as Truncated Normal Distribution.*

Description

Fits a Normal distribution, truncated between 0 and 1, to the given sample of scores.

Usage

```
effCont_norm(x)
```

Arguments

x	a sample of effectiveness scores between 0 and 1.

Value

an object of class `eff.cont.norm`, which inherits from `eff.cont`.

See Also

`deff`, `peff`, `qeff` and `reff`.

Examples

```
e <- effCont_norm(web2010ap[,1])
c(e$mean, e$var)
plot(e, plot.data = TRUE)
```

effcop *Fit Vine copula models to matrices of effectiveness scores*

Description

Fitting of and simulation from a copula model.

Usage

```
effcopFit(x, eff, ...)

reffcop(n, .effcop)
```

Arguments

x	a matrix or data frame of effectiveness scores to estimate dependence.
eff	a list of effectiveness distributions to use for the margins.
...	other parameters for vinecop, such as family_set, selcrit, trunc_lvl and cores.
n	number of observations to simulate.
.effcop	the effcop object representing the copula model (see effcopFit).

Value

effcopFit: an object of class effcop, with the following components:

data	the matrix of effectiveness scores used to fit the copula.
pobs	the matrix of pseudo-observations computed from data. This is stored because pseudo-observations are calculated
margins	the list of marginal effectiveness distributions.
cop	the underlying copulas fitted with vinecop.

These components may be altered to gain specific simulation capacity, such as systems with the same expected value.

reffcop: a matrix of random scores.

See Also

effCont and effDisc for available distributions for the margins. See package rvinecopulib for details on fitting the copulas.

Examples

```
## Automatically build a gaussian copula to many systems
d <- web2010p20[,1:20] # sample P@20 data from 20 systems
effs <- effDiscFitAndSelect(d, support("p20")) # fit and select margins
cop <- effcopFit(d, effs, family_set = "gaussian") # fit copula
y <- reffcop(1000, cop) # simulate new 1000 topics

# compare observed vs. expected mean
E <- sapply(effs, function(e) e$mean)
E.hat <- colMeans(y)
plot(E, E.hat)
abline(0:1)

# compare observed vs. expected variance
Var <- sapply(effs, function(e) e$var)
Var.hat <- apply(y, 2, var)
plot(Var, Var.hat)
abline(0:1)
```

---

effDisc *Discrete Effectiveness Distributions*

Description

Families to model effectiveness distributions with discrete support. Currently implemented families are:

effDisc_bbinom	Beta-Binomial
effDisc_dks	Kernel-smoothed with Discrete kernel.

See Also

effDiscFit to fit discrete distributions, and eff.disc for the S3 class. For continuous distributions, see effCont.

---

effDisc-helper *Helper functions for discrete effectiveness distributions*

Description

These are functions to help in the creation and use of discrete effectiveness distributions.

Usage

```
matchTol(x, support, tol = 1e-04)

support(measure, runLength = 1000)
```

Arguments

x	a vector of effectiveness scores.
support	the support of the distribution.
tol	tolerance for matching.
measure	the case insensitive name of the effectiveness measure. See Details.
runLength	the maximum number of documents retrieved for a query (defautls to 1000).

Details

matchTol returns a vector of the positions of matches of x in the vector of possible support values, within tolerance (see match). This is helpful when data are loaded from disk and possibly rounded or truncated.

support obtains the discrete support defined by an effectiveness measure given its name. Current measures are Reciprocal Rank ("RR"), and Precision at k ("P@k" or "Pk", where k is the cutoff, eg. "P@10" or "P10").

Value

matchTol: an integer vector giving the position in the support of the match if there is a match, otherwise NA.

support: the support of the distribution of scores defined by the measure.

See Also

eff.disc.

Examples

```
support("rr")
support("rr", runLength = 10)
support("p@10")
support("p20")

(i <- matchTol(c(.1, .4, .41, .40001), support("p10")))
support("p10")[i]
```

---

effDisc_bbinom *Discrete Effectiveness as Beta-Binomial Distribution.*

Description

Fits a discrete kernel-smoothed distribution, to the given sample of scores and support points.

Usage

```
effDisc_bbinom(x, support)
```

Arguments

x	a sample of effectiveness scores between 0 and 1.
support	the support of the distribution.

Value

an object of class `eff.disc.bbinom`, which inherits from `eff.disc`.

See Also

`deff`, `peff`, `qeff` and `reff`.

Examples

```
e <- effDisc_bbinom(web2010p20[,1], seq(0,1,.05))
c(e$mean, e$var)
plot(e, plot.data = TRUE)
```

---

effDisc_dks *Discrete Effectiveness as Discrete Kernel-smoothed Distribution.*

Description

Fits a Beta-Binomial distribution, to the given sample of scores and support points.

Usage

```
effDisc_dks(x, support, mult = 1)
```

Arguments

x	a sample of effectiveness scores between 0 and 1.
support	the support of the distribution.
mult	a constant to multiply the initially selected bandwidth.

Value

an object of class `eff.disc.dks`, which inherits from `eff.disc`.

References

M.C. Wang and J.V. Ryzing (1981). A Class of Smooth Estimators for Discrete Distributions. Biometrika, 68, 301-309.

See Also

`deff`, `peff`, `qeff` and `reff`.

Examples

```
e <- effDisc_dks(web2010p20[,1], seq(0,1,.05))
c(e$mean, e$var)
plot(e, plot.data = TRUE)
e2 <- effDisc_dks(web2010p20[,1], seq(0,1,.05), mult = 2)
c(e2$mean, e2$var)
plot(e2, plot.data = TRUE)
```

---

effFit *Fit Effectiveness Distributions*

Description

Attempts to fit the distribution families listed in effCont or effDisc. In the discrete case, the dks distribution is fitted with multipliers 1, 2, 5 and 10. Failure to fit any distribution family results in an error.

Usage

```
effContFit(x, silent = TRUE)

effDiscFit(x, support, silent = TRUE)
```

Arguments

x	a sample of effectiveness scores between 0 and 1.
silent	logical: should the report of error messages be suppressed?
support	the support of the distribution (see support).

Value

a list of eff.cont objects fitted to the given data.

See Also

effCont and effDisc for the available distribution families.

See effSelect for model selection, and effFitAndSelect to fit and select automatically.

Examples

```
e <- effContFit(web2010ap[,1])
str(e, 1)
sapply(e, plot, plot.data = TRUE)

e <- effDiscFit(web2010p20[,1], seq(0,1,.05))
str(e, 1)
sapply(e, plot, plot.data = TRUE)
```

effFitAndSelect *Automatic Fitting and Selection of Effectiveness Distributions*

Description

Automatic Fitting and Selection of Effectiveness Distributions

Usage

```
effContFitAndSelect(x, method = "AIC", silent = TRUE)

effDiscFitAndSelect(x, support, method = "AIC", silent = TRUE)
```

Arguments

x	a sample of effectiveness scores between 0 and 1, or a matrix or data frame of topic-by-system scores.
method	selection method. See effSelect.
silent	logical: should the report of error messages be suppressed?
support	the support of the distribution (see support).

Value

if x is a vector, the selected disttribution. If x is a matrix or data frame, a list of the selected distributions.

See Also

effFit and effSelect.

Examples

```
e <- effContFitAndSelect(web2010ap[,1], method = "logLik")
c(e$mean, e$var)
e2 <- effContFitAndSelect(web2010ap[,2], method = "logLik")
c(e2$mean, e2$var)

ee <- effContFitAndSelect(web2010ap[,1:2], method = "logLik")
sapply(ee, function(e) c(e$mean, e$var)) # same as above
```

effSelect *Model Selection for Effectiveness Distributions*

Description

Functions to compute the log-likelihood, the Akaike Information Criterion, and the Bayesian Information Criterion for an effectiveness distribution. `effSelect` and `which.effSelect` are helper function for automatic selection from a given list of candidates.

Usage

```
effSelect(effs, method = "AIC", ...)

which.effSelect(effs, method = "AIC", ...)

## S3 method for class 'eff'
logLik(object, ...)
```

Arguments

`effs`	the list of candidate distributions to select from.
`method`	selection method. One of `"AIC"` (default), `"BIC"`, or `"logLik"`.
`...`	other parameters to the selection function.
`object`	an effectiveness distribution.

Value

the selected disttribution (`effSelect`), or its index within `effs` (`which.effSelect`).

See Also

`logLik`, `AIC`, `BIC` for details on model selection.

See `effFitAndSelect` to fit and select automatically.

Examples

```
ee <- effContFit(web2010ap[,5])
e <- effSelect(ee, method = "BIC")
e2 <- ee[[which.effSelect(ee, method = "BIC")]] # same as e

logLik(e)
AIC(e, k=4)
BIC(e)
```

`effTransform` *Transform effectiveness distributions towards a expected value*

Description

Transforms the given effectiveness distribution such that its expected value matches a predefined value. For details, please refer to section 3.4 of (Urbano and Nagler, 2018).

Usage

```
effTransform(eff, mean, abs.tol = 1e-05)

effTransformAll(effs, means, abs.tol = 1e-05, silent = TRUE)
```

Arguments

`eff`	the distribution to transform.
`mean`	the target expected value to transform to. If missing, defaults to the mean in the data used to fit `eff`, if any.
`abs.tol`	the absolute tolerance of the transformation.
`effs`	the list of distributions to transform.
`means`	the vector of target expected values to transform to. If missing, defaults to the means in the data used to fit `effs`, if any.
`silent`	logical: should the report of error messages be suppressed?

Details

`effTransformAll` does the same but for a list of distributions and target means.

Value

an effectiveness distribution of class `eff.cont.trans` or `eff.disc.trans`, depending on the type of distribution.

References

J. Urbano and T. Nagler. (2018). Stochastic Simulation of Test Collections: Evaluation Scores. ACM SIGIR.

Examples

```
e <- effCont_beta(web2010ap[,1])
e2 <- effTransform(e, 0.12)
c(e$mean, e2$mean)
plot(e)
plot(e2)
```

```
# transform a list of distributions to the observed means
ee <- effContFitAndSelect(web2010ap[,1:5])
ee2 <- effTransformAll(ee)
obsmeans <- colMeans(web2010ap[,1:5])
sapply(ee, function(e)e$mean) - obsmeans
sapply(ee2, function(e)e$mean) - obsmeans
```

`plot.eff` *Plotting tools for effectiveness distributions*

Description

Plot the density, distribution and quantile functions of an effectiveness distribution. Function `plot` plots all three functions in the same graphics device.

Usage

```
## S3 method for class 'eff'
plot(x, ..., plot.data = TRUE)

dplot(x, ..., plot.data = TRUE)

pplot(x, ..., plot.data = TRUE)

qplot(x, ..., plot.data = TRUE)
```

Arguments

`x`	the effectiveness distribution to plot.
`...`	other arguments to be passed to graphical functions.
`plot.data`	logical: whether to plot the data used to fit the distribution, if any.

See Also

`plot.eff.cont` and `plot.eff.disc` for more details.

plot.eff.cont *Plotting tools for Continuous effectiveness distributions*

Description

Plot the density, distribution and quantile functions of a continuous effectiveness distribution.

Usage

```
## S3 method for class 'eff.cont'
dplot(x, ..., plot.data = TRUE, subdivisions = 200,
  xlab = "x", ylab = "f(x)", main = "density")

## S3 method for class 'eff.cont'
pplot(x, ..., plot.data = TRUE, subdivisions = 200,
  xlab = "q", ylab = "F(q)", main = "distribution")

## S3 method for class 'eff.cont'
qplot(x, ..., plot.data = TRUE, subdivisions = 200,
  xlab = "p", ylab = expression(F^-1 * (p)), main = "quantile")
```

Arguments

x	the effectiveness distribution to plot.
...	arguments to be passed to lines.
plot.data	logical: whether to plot the data used to fit the distribution, if any.
subdivisions	number of equidistant points at which to evaluate the distribution to plot.
xlab	the title for the x axis.
ylab	the title for the y axis.
main	the overall title for the plot.

See Also

plot.eff.disc for discrete distributions.

plot.eff.disc *Plotting tools for Discrete effectiveness distributions*

Description

Plot the density, distribution and quantile functions of a discrete effectiveness distribution.

Usage

```
## S3 method for class 'eff.disc'
dplot(x, ..., plot.data = TRUE, xlab = "x",
  ylab = "f(x)", main = "mass")

## S3 method for class 'eff.disc'
pplot(x, ..., plot.data = TRUE, xlab = "q",
  ylab = "F(q)", main = "distribution")

## S3 method for class 'eff.disc'
qplot(x, ..., plot.data = TRUE, xlab = "p",
  ylab = expression(F^-1 * (p)), main = "quantile")
```

Arguments

x	the effectiveness distribution to plot.
...	arguments to be passed to lines.
plot.data	logical: whether to plot the data used to fit the distribution, if any.
xlab	the title for the x axis.
ylab	the title for the y axis.
main	the overall title for the plot.

See Also

plot.eff.cont for continuous distributions.

---

web2010ap *TREC 2010 Web Ad hoc track.*

Description

These are the topic-by-system effectiveness matrices for the 88 systems submitted to the TREC 2010 Web Ad hoc track, evaluated over 48 topics. web2010ap contains Average Precision scores, web2010p20 contains Precision at 20 scores, and web2010rr contains Reciprocal Rank scores.

Usage

```
web2010ap

web2010p20

web2010rr
```

Format

A data frame with 88 columns (systems) and 48 rows (queries).

References

C.L.A. Clarke, N. Craswell, I. Soboroff, G.V. Cormack (2010). Overview of the TREC 2010 Web Track. Text REtrieval Conference.

See Also

https://trec.nist.gov

Index