Package ‘errint’

October 13, 2022

Type Package
Title Builds Error Intervals
Version 1.0
Date 2017-01-22
Author Jesus Prada [aut,cre]
Maintainer Jesus Prada <jesus.prada@estudiante.uam.es>
Description Builds and analyzes error intervals for a particular model predictions assuming different distributions for noise in the data.
Depends VGAM, rootSolve
License GPL-2
LazyData TRUE
BuildVignettes TRUE
RoxygenNote 5.0.1
URL http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47
NeedsCompilation no
Repository CRAN
Date/Publication 2017-01-23 06:35:35

R topics documented:

acc_intervals ......................................................... 2
best_distribution ................................................... 3
df_intervals .......................................................... 4
df_intervals.default ................................................. 6
error_interval ......................................................... 7
error_interval.default .............................................. 8
int_lap ................................................................. 9
measure ............................................................... 12
measure.default ................................................... 13
print.df_intervals ................................................... 14
Description

`int_intervals` computes the real accuracy of a given error intervals for a particular set of errors and a particular error function.

Usage

```r
acc_intervals(interv, errors, f = function(x, y) { abs(x - y) },
             tol = 10^-8)
```

Arguments

- `interv`: error interval.
- `errors`: set of errors.
- `f`: error function to be used to compute error between real `x (interv)` and predicted `y (errors)` values. See also 'Details'.
- `tol`: used to normalize residual values to (0,1) when beta is the assumed distribution. See also 'Details'.

Details

`f` must be a function that takes two arguments, `x` and `y`, and return a numeric value.

The formula used to normalize residual values to (0,1) when a Beta distribution is assumed is

\[
\frac{\phi}{\max|\phi| + \text{tol}}.
\]

Value

Returns an object of class `c("measure", "list")` with information of the interval accuracy.

Author(s)

Jesus Prada, <jesus.prada@estudiante.uam.es>
best_distribution

References

Link to the scientific paper


with theoretical background for this package is provided below.

http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47

See Also

measure error_interval

Examples

interv<-int_gau(rnorm(10),0.1)
acc_intervals(interv,rnorm(10))
acc_intervals(interv,rnorm(10),function(x,y){x-y})

best_distribution Distribution with Best Error Intervals

Description

best_distribution computes the distribution assumption that gives error intervals with the lower accuracy error for a given set of residuals.

Usage

best_distribution(phi, errors, dists = c("n", "nm", "l", "lm", "w", "b", "moge"), ...)

Arguments

phi residual values used to compute the error interval.
errors set of real errors corresponding to the predictions of a particular model.
dists character vector with the distribution assumptions to test. See also 'Details'.
... additional arguments to be passed to functions error_interval and acc_intervals.

Details

Allowed distribution assumptions are:

- "n": Zero-mu Gaussian
- "nm": General Gaussian
- "l": Zero-mu Laplace
• "lm": General Laplace
• "b": Beta
• "w": Weibull
• "moge": Moge

Value

Returns an object of class c("df_intervals", "data.frame") with information of the distribution assumption with lower accuracy error.

Author(s)

Jesus Prada, <jesus.prada@estudiante.uam.es>

References

Link to the scientific paper

with theoretical background for this package is provided below.
http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47

See Also

df_intervals error_interval acc_intervals

Examples

best_distribution(rnorm(10),rnorm(10),dists=c("n","b"))
Usage

df_intervals(distributions, errs)
as.df_intervals(x)
is.df_intervals(x)

Arguments

distributions vector containing the names of the distribution correspondind to each error.
erss vector of errors associated to intervals built under a particular distribution assumption indicated by 'distributions'.
x an R object.

Value

df_intervals returns an object of class c("df_intervals", "data.frame") with information regarding the error of intervals built under different distribution assumptions.
as.df_intervals returns an object of class c("df_intervals",class(x)) with information contained in x if possible. Returns x otherwise.
is.df_intervals returns TRUE if x is an R object with "df_intervals" as one of its classes. FALSE otherwise.

Author(s)

Jesus Prada, <jesus.prada@estudiante.uam.es>

References

Link to the scientific paper
with theoretical background for this package is provided below.
http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47

Examples

df_intervals("l",0.1)
df_intervals(c("l","lm","n","nm","b","w"),rep(0.1,6))

df<-data.frame(distribution=rnorm(10),error=rnorm(10))
as.df_intervals(df)
v<-c("a","b")
as.df_intervals(v)
### df_intervals.default

**Description**

`df_intervals` creates an object of class `c("df_intervals", "data.frame")`.

**Usage**

```r
## Default S3 method:
df_intervals(distributions, errs)
```

**Arguments**

- `distributions`: vector containing the names of the distribution corresponding to each error.
- `errs`: vector of errors associated to intervals built under a particular distribution assumption indicated by `distributions`.

**Value**

Returns an object of class `c("df_intervals", "data.frame")` with information regarding the error of intervals built under different distribution assumptions.

**Author(s)**

Jesus Prada, `<jesus.prada@estudiante.uam.es>`

**References**

Link to the scientific paper


with theoretical background for this package is provided below.

[http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47](http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47)

**Examples**

```r
df<-data.frame(distribution=rnorm(10),error=rnorm(10))
is.df_intervals(df)
res<-as.df_intervals(df)
is.df_intervals(res)
```

```r
df_intervals("l",0.1)
df_intervals(c("l","lm","n","nm","b","w"),rep(0.1,6))
```
Description

`error_interval` creates an object of class `c("error_interval", "list")`.  
`as.error_interval` attempts to coerce its argument `x` into an object of class `c("error_interval", class(x))`. If this is not possible `x` is returned unchanged.  
`is.error_interval` returns TRUE if `x` is an R object with "error_interval" as one of its classes. It returns FALSE otherwise.

Usage

```
error_interval(phi, s = 0.05, dist = "n", tol = 10^-6, ...)
```
```
as.error_interval(x)
```
```
is.error_interval(x)
```

Arguments

- `phi` a vector with residual values used to compute the error interval.  
- `s` confidence level, e.g. `s=0.05` for the standard 95 percent confidence interval.  
- `dist` assumed distribution for the noise in the data.  
- `tol` used to normalize residual values to (0,1) when beta is the assumed distribution. The formula used is `abs(phi)/(max(abs(phi))+tol)`.  
- `...` additional arguments to be passed to the low level error_interval building functions (see below).  
- `x` an R object.

Value

`error_interval` returns an object of class `c("error_interval","list")` with information regarding the error intervals built.  
`as.error_interval` returns an object of class `c("error_interval",class(x))` with information contained in `x` if possible. Returns `x` otherwise.  
`is.error_interval` returns TRUE if `x` is an R object with "error_interval" as one of its classes. FALSE otherwise.

Author(s)

Jesus Prada, <jesus.prada@estudiante.uam.es>
References


with theoretical background for this package is provided below.

http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47

Examples

```r
error_interval(rnorm(10))

error_interval(rnorm(10),s=0.1,dist="lm")
```

```r
l <- list(min=-1,max=1,err=0.05,s=0.1,dist="n",phi=rnorm(10))
as.error_interval(l)

v <- c("a","b")
as.error_interval(v)

l <- list(min=-1,max=1,err=0.05,s=0.1,dist="n",phi=rnorm(10))
is.error_interval(l)
res <- as.error_interval(l)
is.error_interval(res)
```

---

**error_interval.default**

*Error Intervals*

Description

`error_interval.default` creates an object of class `c("error_interval", "list")`.

Usage

```r
## Default S3 method:
error_interval(phi, s = 0.05, dist = "n", tol = 10^-6, ...)
```

Arguments

- `phi`: a vector with residual values used to compute the error interval.
- `s`: confidence level, e.g. `s=0.05` for the standard 95 percent confidence interval.
- `dist`: assumed distribution for the noise in the data.
used to normalize residual values to \((0,1)\) when beta is the assumed distribution. The formula used is \(\frac{\text{abs}(\phi)}{\max(\text{abs}(\phi))+\text{tol}}\).

... additional arguments to be passed to the low level error_interval building functions (see below).

**Value**

Returns an object of class `c("error_interval","list")` with information regarding the error intervals built.

**Author(s)**

Jesus Prada. &lt;jesus.prada@estudiante.uam.es&gt;

**References**

Link to the scientific paper


with theoretical background for this package is provided below.

http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47

**Examples**

```r
error_interval(rnorm(10))
error_interval(rnorm(10), s=0.1, dist="lm")
```

---

**Description**

`int_lap` computes the error interval of a set of residuals assuming a Laplace distribution with zero location for the noise.

`int_gau` computes the error interval of a set of residuals assuming a Gaussian distribution with zero mean for the noise.

`int_lap_mu` computes the error interval of a set of residuals assuming a Laplace distribution.

`int_gau_mu` computes the error interval of a set of residuals assuming a Gaussian distribution.

`int_beta` computes the error interval of a set of residuals assuming a Beta distribution.

`int_weibull` computes the error interval of a set of residuals assuming a Weibull distribution.

See also 'Details'.

`int_moge` computes the error interval of a set of residuals assuming a MOGE distribution.
Usage

`int_lap(phi, s)`

`int_gau(phi, s, ps = 0, threshold = 10^-2, upper = 10^6)`

`int_lap_mu(phi, s, ps = stats::median(phi, na.rm = T), threshold = 10^-2, upper = 10^6)`

`int_gau_mu(phi, s, ps = mean(phi, na.rm = T), threshold = 10^-2, upper = 10^6)`

`int_beta(phi, s, original_phi = phi, ps = 10^-4, threshold = 10^-4, upper = 1, m1 = mean(phi, na.rm = T), m2 = mean(phi^2, na.rm = T), alpha_0 = (m1 * (m1 - m2))/(m2 - m1^2), beta_0 = (alpha_0 * (1 - m1)/m1))`

`int_weibull(phi, s, ps = 10^-4, threshold = 10^-2, upper = 10^6, k_0 = 1)`

`int_moge(phi, s, ps = 10^-4, threshold = 10^-4, upper = 10^6, lambda_0 = 1, alpha_0 = 1, theta_0 = 1)`

Arguments

`phi`  
residual values used to compute the error interval.

`s`  
confidence level, e.g. s=0.05 for the standard 95 percent confidence interval.

`ps`  
minimum value to search for solution of the integral equation to solve. See also 'Details'.

`threshold`  
step size to increase `ps` after each iteration. See also 'Details'.

`upper`  
maximum value to search for solution of the integral equation to solve. See also 'Details'.

`original_phi`  
original \{φ\} values. Only used for beta distribution.

`m1`  
first moment of the residuals. Used to compute `alpha_0`.

`m2`  
second moment of the residuals. Used to compute `beta_0`.

`alpha_0`  
initial value for Newton-Raphson method for the parameter α. See also 'Details' and `multiroot`.

`beta_0`  
initial value for Newton-Raphson method for the parameter β. See also 'Details' and `multiroot`.

`k_0`  
initial value for Newton-Raphson method for the parameter κ. See also 'Details' and `multiroot`.

`lambda_0`  
initial value for Newton-Raphson method for the parameter λ.

`theta_0`  
initial value for Newton-Raphson method for the parameter θ.
Details

For the Zero-µ Laplace distribution the value of the corresponding integral equation has a closed solution of the form \( ps = -\sigma \log 2s \).

For the other distributions, starting with the initial value of \( ps \) passed as argument, the value, integral, of the corresponding integral expression is computed (see also 'References' for an in-depth explanation of this integral expression). If integral is smaller than \( 1-s \) then \( ps \) is increased by a step size of threshold value and integral is recomputed. If integral is greater or equal than 0 or if \( ps \) gets bigger than upper, the loop stops and the last value of \( ps \) will be its final value.

In addition, for the Beta distribution values of parameters \( \alpha \) and \( \beta \) are estimated using Newton-Raphson method.

For the Weibull distribution value of parameter \( \kappa \) is estimated using Newton-Raphson method and then estimated value of \( \lambda \) is computed using a closed form that depends on \( \kappa \).

For the MOGE distribution values of parameters \( \lambda \), \( \alpha \) and \( \theta \) are estimated using Newton-Raphson method.

See also 'References'.

Value

Returns an object of class c("error_interval","list") with information of the corresponding error interval.

Author(s)

Jesus Prada, <jesus.prada@estudiante.uam.es>

References

Link to the scientific paper


with theoretical background for this package is provided below.

http://link.springer.com/content/pdf/10.1007/978-3-319-19222-2_47

See Also

error_interval
p_laplace
p_gaussian
p_beta
p_weibull
multiroot
p_moge
Examples

```r
int_lap(rnorm(10),0.1)

int_gau(rnorm(10),0.1,0.1,10^-3,10^2)

int_lap_mu(rnorm(10),0.1,0.1,10^-3,10^2)

int_gau_mu(rnorm(10),0.1,0.1,10^-3,10^2)

int_beta(runif(10,0,0.99),0.1,alpha_0=1,beta_0=1)

int_weibull(abs(rnorm(10)),0.1,k_0=2)

int_moge(runif(10,0.01,0.99),0.1,lambda_0=2,alpha_0=3,theta_0=4)
```

---

**Description**

measure creates an object of class c("measure", "list").

as.measure attempts to coerce its argument x into an object of class c("measure", class(x)). If this is not possible x is returned unchanged.

is.measure returns TRUE if x is an R object with "measure" as one of its classes. It returns FALSE otherwise.

**Usage**

```r
measure(s, acc, f = function(x, y) { abs(x - y) })

as.measure(x)

is.measure(x)
```

**Arguments**

- `s`  confidence level, e.g. `s=0.05` for the standard 95 percent confidence interval.
- `acc` accuracy achieved by error intervals.
- `f`  function used to compute error of intervals. See also 'Details'.
- `x`  an R object.
**measure.default**

**Value**

`measure` returns an object of class `c("measure", "list")` with information regarding the error of a set of intervals.

`as.measure` returns an object of class `c("measure", class(x))` with information contained in `x` if possible. Returns `x` otherwise.

`is.measure` returns `TRUE` if `x` is an R object with "measure" as one of its classes. `FALSE` otherwise.

**Author(s)**

Jesus Prada, <jesus.prada@estudiante.uam.es>

**References**

Link to the scientific paper


with theoretical background for this package is provided below.

[http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47](http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47)

**Examples**

```r
measure(0.1, 0.7)

measure(0.1, 0.7, function(x, y) {y - x})

l <- list(s = 0.1, acc = 0.78, f = function(x, y) {abs(x - y)}, err = 0.02)
as.measure(l)

v <- c("a", "b")
as.measure(v)

l <- list(s = 0.1, acc = 0.78, f = function(x, y) {abs(x - y)}, err = 0.02)
is.measure(l)
res <- as.measure(l)
is.measure(res)
```

---

**measure.default**

**Measure**

**Description**

`measure` creates an object of class `c("measure", "list")`.

---
print.df_intervals

Usage

## Default S3 method:
measure(s, acc, f = function(x, y) { abs(x - y) })

Arguments

s  confidence level, e.g. s=0.05 for the standard 95 percent confidence interval.
acc accuracy achieved by error intervals.
f  function used to compute error of intervals. See also 'Details'.

Value

Returns an object of class c("measure","list") with information regarding the error of a set of intervals.

Author(s)

Jesus Prada, <jesus.prada@estudiante.uam.es>

References

Link to the scientific paper
with theoretical background for this package is provided below.
http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47

Examples

measure(0.1,0.7)
measure(0.1,0.7,function(x,y){y-x})

print.df_intervals   Printing Data Frames of Intervals

Description

print objects of class df_interval.

Usage

## S3 method for class 'df_intervals'
print(x, ...)

print.df_intervals   Printing Data Frames of Intervals

Description

print objects of class df_interval.

Usage

## S3 method for class 'df_intervals'
print(x, ...)

print.df_intervals   Printing Data Frames of Intervals

Description

print objects of class df_interval.
Arguments

x object of class df_interval to be printed.

... optional arguments.

Author(s)

Jesus Prada, <jesus.prada@estudiante.uam.es>

References

Link to the scientific paper


with theoretical background for this package is provided below.

http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47

See Also

df_intervals

Examples

```
res<-df_intervals(c("l","lm","n","nm","b","w"),rep(0.1,6))
print(res)
```

print.error_interval  Printing Error Intervals

Description

print objects of class error_interval.

Usage

```r
## S3 method for class 'error_interval'
print(x, ...)
```

Arguments

x object of class error_interval to be printed.

... optional arguments.

Author(s)

Jesus Prada, <jesus.prada@estudiante.uam.es>
References

Link to the scientific paper

with theoretical background for this package is provided below.

http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47

See Also

error_interval

Examples

res<-error_interval(rnorm(10))
print(res)

Description

print objects of class measure.

Usage

## S3 method for class 'measure'
print(x, ...)

Arguments

x object of class measure to be printed.
...
optional arguments.

Author(s)

Jesus Prada, <jesus.prada@estudiante.uam.es>

References

Link to the scientific paper

with theoretical background for this package is provided below.

http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47
See Also

measure

Examples

res<-measure(0.1,0.7)
print(res)

res<-error_interval(rnorm(10))
summary(res)
Description

print objects of class summary.measure.

Usage

```r
## S3 method for class 'summary.measure'
print(x, ...)
```

Arguments

- `x` object of class summary.measure to be printed.
- `...` optional arguments.

Author(s)

Jesus Prada, <jesus.prada@estudiante.uam.es>

References

Link to the scientific paper


with theoretical background for this package is provided below.

http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47

See Also

summary measure

Examples

```r
res<-measure(0.1,0.7)
summary(res)
```
**p_laplace**

**Probability Density Functions**

**Description**

\( p_{\text{laplace}} \) computes the probability density function of a random variable that has a Laplace distribution with parameters \( \mu \) and \( \sigma \).

\( p_{\text{gaussian}} \) computes the probability density function of a random variable that has a Gaussian distribution with parameters \( \mu \) and \( \sigma^2 \).

\( p_{\text{beta}} \) computes the probability density function of a random variable that has a Beta distribution with parameters \( \alpha \) and \( \beta \).

\( p_{\text{weibull}} \) computes the probability density function of a random variable that has a Weibull distribution with parameters \( \kappa \) and \( \lambda \).

\( p_{\text{moge}} \) computes the probability density function of a random variable that has a MOGE distribution with parameters \( \lambda, \alpha \) and \( \theta \).

**Usage**

\[ p_{\text{laplace}}(x, \ mu = 0, \ sigma = 1) \]

\[ p_{\text{gaussian}}(x, \ mu = 0, \ sigma\_cuad = 1) \]

\[ p_{\text{beta}}(x, \ alpha = 1, \ beta = 1) \]

\[ p_{\text{weibull}}(x, \ k = 1, \ lambda = 1) \]

\[ p_{\text{moge}}(x, \ lambda = 1, \ alpha = 1, \ theta = 1) \]

**Arguments**

- **x** vector of points which values we want to compute.
- **mu** location or mean parameter of the Laplace or Gaussian distribution, respectively.
- **sigma** scale parameter of the Laplace distribution.
- **sigma\_cuad** variance parameter of the Gaussian distribution.
- **alpha** shape1 parameter of the Beta distribution or second parameter of the MOGE distribution.
- **beta** shape2 parameter of the Beta distribution.
- **k** shape parameter of the Weibull distribution.
- **lambda** scale parameter of the Weibull distribution or first parameter of the MOGE distribution.
- **theta** third parameter of the MOGE distribution.
Value

Returns a numeric object corresponding to the value of the probability density function for the given x and distribution parameters.

Author(s)

Jesus Prada, <jesus.prada@estudiante.uam.es>

References

Link to the scientific paper


with theoretical background for this package is provided below.

http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47

See Also

dlaplace
dnorm
dbeta
dweibull

Examples

p_laplace(0.3)
p_laplace(0.3,mu=0.35,sigma=0.2)

p_gaussian(0.3)
p_gaussian(0.3,mu=0.35,sigma_cuad=0.2)

p_beta(0.3)
p_beta(0.3,alpha=0.35,beta=0.2)

p_weibull(0.3)
p_weibull(0.3,k=0.35,lambda=0.2)

p_moge(0.3)
p_moge(0.3,lambda=0.2,alpha=0.3,theta=0.4)
sort_distributions

Sort Distributions by Better Error Intervals

Description

sort_distributions orders a given set of distribution assumptions in order of intervals accuracy error in ascending or descending order.

Usage

sort_distributions(phi, errors, dists = c("n", "nm", "l", "lm", "w", "b", "moge"), decreasing = FALSE, ...)

Arguments

phi residual values used to compute the error interval.
errors set of real errors corresponding to the predictions of a particular model.
dists character vector with the distribution assumptions to test. See also 'Details'.
decreasing logical, indicating whether or not distributions should be ordered by decreasing accuracy error.
... additional arguments to be passed to functions error_interval and acc_intervals.

Details

Allowed distribution assumptions are:

- "n": Zero-mu Gaussian
- "nm": General Gaussian
- "l": Zero-mu Laplace
- "lm": General Laplace
- "b": Beta
- "w": Weibull
- "moge": Moge

Value

Returns an object of class c("df_intervals", "data.frame") with information of the distribution assumptions ordered by accuracy error.

Author(s)

Jesus Prada, <jesus.prada@estudiante.uam.es>
References

Link to the scientific paper


with theoretical background for this package is provided below.

http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47

See Also

df_intervals error_interval acc_intervals order

Examples

sort_distributions(rnorm(10),rnorm(10),decreasing=TRUE)

summary.error_interval

Error Intervals Summaries

Description

summary produces summaries for objects of class error_interval.

Usage

summary.error_interval(object, ...)

Arguments

object object of class error_interval to be printed.
... optional arguments.

Value

Object of class c("summary.error_interval","list") corresponding to the summary of x.

Author(s)

Jesus Prada, <jesus.prada@estudiante.uam.es>
References

Link to the scientific paper


with theoretical background for this package is provided below.

http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47

See Also

error_interval

Examples

res<-error_interval(rnorm(10))
summary(res)

summary.measure  Measures Summaries

Description

summary produces summaries for objects of class measure.

Usage

summary.measure(object, ...)

Arguments

object object of class measure to be printed.
...
optional arguments.

Value

Object of class c("summary.measure","list") corresponding to the summary of x.

Author(s)

Jesus Prada, <jesus.prada@estudiante.uam.es>
References

Link to the scientific paper
with theoretical background for this package is provided below.
http://link.springer.com/chapter/10.1007/978-3-319-19222-2_47

See Also

measure

Examples

res<-measure(0.1,0.7)
summary(res)
Index

acc_intervals, 2, 4, 22
as.df_intervals (df_intervals), 4
as.error_interval (error_interval), 7
as.measure (measure), 12

best_distribution, 3

dbeta, 20
df_intervals, 4, 4, 15, 22
df_intervals.default, 6
dlaplace, 20
dnorm, 20
dweibull, 20

error_interval, 3, 4, 7, 11, 16, 17, 22, 23
error_interval.default, 8

int_beta (int_lap), 9
int_gau (int_lap), 9
int_gau_mu (int_lap), 9
int_lap, 9
int_lap_mu (int_lap), 9
int_moge (int_lap), 9
int_weibull (int_lap), 9
is.df_intervals (df_intervals), 4
is.error_interval (error_interval), 7
is.measure (measure), 12

measure, 3, 12, 17, 18, 24
measure.default, 13
multiroot, 10, 11

order, 22

p_beta, 11
p_beta (p_laplace), 19
p_gaussian, 11
p_gaussian (p_laplace), 19
p_laplace, 11, 19
p_moge, 11
p_moge (p_laplace), 19
p_weibull, 11
p_weibull (p_laplace), 19
print.df_intervals, 14
print.error_interval, 15
print.measure, 16
print.summary.error_interval, 17
print.summary.measure, 18

sort_distributions, 21
summary, 17, 18
summary.error_interval, 22
summary.measure, 23

25