Package ‘Equitable’

March 11, 2019

Type Package

Title Equitable Transform

Version 0.1.0

Author C. Elphinstone, R. D. Elphinstone

Maintainer C. Elphinstone <cassandra.elphinstone@shaw.ca>

Description A method of scaling to maintain individuality
The equitable transformation maintains systematic differences between instances when the underlying pattern is a separable function in two variables + an addional function of one of these variables. This technique converts “equitably” between comparable instances in a dataset.
Applied to a data set, it removes noise superimposed on an underlying signal enabling patterns to be extracted more easily from a 2D data set. Elphinstone, C. and Henry, G. (submitted to Ecosphere, 2019) show the multipurpose uses for the equitable transform. Functions useful for analysing two dimensional data sets are included in this package. A two dimensional data set (d) is transformed via Td<-transformE(d) and a variety of matrices associated with the equitable transform is placed in Td. A signal data set can be made via d<-eg4(2,2) and noise can be added to make a noisy 2D data set (d_noise) in a variety of ways. If Td and Td_noise are the results from running transformE then plotsummary(Td_noise,Td), plotsummary(Td) and plotsummary(Td_noise) summarize and plot information about the two transformations.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Imports Hmisc,
SDMTools,
aplpack,
stats,
graphics,
grDevices

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

`AI_Bmult_NA` *Builds a simple (unweighted) equitable transform.*

Description

It is based on equitable slope (A) and shift (B) matrices and data set I Both A and B can be tored efficiently by only retaining 1 column from each matrix NA values are allowed If a transform exists, this function can construct a complete data set from either: 1. a complete data set 2. A data set of only an average profile in variable t or a profile for some individual or location 3. profile at one rbitrary location is available 4. scattered observations across the different "individuals" or locations x for different values of t Examples of each are given. Only 2*length of x dimension + length of t dimension are needed to reconstruct system If the data is not precisely equitable then more data provides a better estimate. # Errors are also given to show how closely the system matches the equitbale one that is assumed.

Usage

```
AI_Bmult_NA(A, I, B = NULL, zero = 0, maxA = NULL)
```

Arguments

`A`	equitable slope matrix NxN
`I`	two dimensional data set MxN that can contain many NA values
`B`	equitable shift matrix MxM
`zero`	Value to be subtracted from I to generate B default NULL (0)
`maxA`	Values in the slope matrix larger than this threshold are not used in the transform Default NULL

Value

Equitable transform of data that fills in many of the NA values based on the tranformed values

Examples

```
#case 1 not shown. It is less interesting unless it is a noisy or corrupted system
#case 2: Use an average profile with 2 vectors from the matrices to construct
#        transformed data. (2N+M data points)
# To illustrate the idea first use an an equitable data set to construct
#            equitable matrix A and shift matrix B
#from which vectors f an u are extracted. These are used to reconstruct the matrices when desired.

d<-eg4(2,2); Td<-transformE(d)  # make a data set and find the transform and associated matrices
f<-Td$E.s[,1]                   # form
u<-Td$E.b[,1]
I<-Td$smat;I[]<-NA; I[,"Row_Ave"]<-Td$smat[,"Row_Ave"]
# the 3 necessary components are made so show what they look like and then construct system
plot(f,type="b",main="f(x) profile used for Equitable matrix A")
plot(u,type="b",main="u(x) profile used with matrix A\nto make  for shift matrix B")
plot(I[, "Row_Ave"],type="b",,main="Average profile used for reconstruction")
imagenan(I,main="Only the average profile is defined")
A<-make_A(f=f)
B<-make_B(xref=1,orig=u,A=A,uflag=TRUE)
TI<-AI_Bmult_NA(A=A,I=I,B=B)
attributes(TI)
imagenan(TI$xme,main="Reconstructed data set from Average profile")
         #data successfully reconstructed

# case 3  profile at x=10 available but it is somewhat noisy
xref<-10; I<-Td$smat;I[]<-NA;
I[,xref]<-Td$smat[,xref]
plot(I[, xref],type="b",main="Profile (no Noise)")
I[,xref]<-I[,xref]+rnorm(nrow(I),mean=0,sd=(1/10*sd(I[,xref],na.rm=TRUE)))
plot(I[, xref],type="b",main="Profile used for reconstruction (with noise)")
imagenan(I,main=paste("data only from profile at location x=",xref))
TI<-AI_Bmult_NA(A=A,I=I,B=B)
attributes(TI)
imagenan(TI$xme,main=paste("Reconstructed data set from noisy profile at location x=",xref))
# case 3  profile at x=10 available but is completely different from original
xref<-10; I<-Td$smat;I[]<-NA;

I[,xref]<-eg5(2,2)[,xref]
plot(I[, xref],type="b",main="Average profile used for reconstruction")
imagenan(I,main=paste("data only from profile at location x=",xref))
TI<-AI_Bmult_NA(A=A,I=I,B=B)
attributes(TI)
imagenan(TI$xme,main=paste("Reconstructed data set from profile at location x=",xref))

#case 5 scattered information around the data set
#using random points over the data set only works
#if the system has the same equitable sense as the matrices
I<-Td$smat
numspaces<-nrow(I)
column_location<-sample(1:ncol(I), numspaces, replace=TRUE)
I_NA<-I; I_NA[]<-NA;g<-rep(NA,nrow(I))
for(t in 1:nrow(I_NA)){I_NA[t,column_location[t]]<-I[t,column_location[t]];
 g[t]<-I[t,column_location[t]]}
plot(g,type="b",main=paste("Data set from random points on x as function of t" ))
imagenan(I_NA,main=paste("Data set from random points on x" ))
A<-make_A(f=f)
```

```
B<-make_B(xref=1,orig=u,A=A,uflag=TRUE)
TI<-AI_Bmult_NA(A=A,I=I_NA,B=B)
attributes(TI)
imagenan(TI$xme,
main=paste("Data set reconstructed from random points on x as function of t" ))

#case 5 scattered information around the data set that is noisy
#using random points over the data set assumes the system has
# the same equitable sense as the matrices
I<-Td$smat ;g<-rep(NA,nrow(I))
numspaces<-10*nrow(I) ;g<-rep(NA,numspaces)
column_location<-sample(1:(ncol(I)-1), numspaces, replace=TRUE)
I_NA<-I; I_NA[]<-NA;g<-rep(NA,nrow(I))
Ieg4<-eg4(2,2)
Ieg4<-Ieg4+  rnorm(prod(dim(Ieg4)),mean=0,sd=(1/10*sd(Ieg4,na.rm=TRUE)))
imagenan(Ieg4,main="Noise added")
for(t in 1:numspaces){
  I_NA[t%%nrow(I)+1,column_location[t]]<-Ieg4[t%%nrow(I)+1,column_location[t]]
                         g[t]<-Ieg4[t%%nrow(I)+1,column_location[t]]
                         }
imagenan(I_NA,main=paste("Data set from random points on x" ))
A<-make_A(f=f)
B<-make_B(xref=1,orig=u,A=A,uflag=TRUE)
TI<-AI_Bmult_NA(A=A,I=I_NA,B=B)
attributes(TI)
imagenan(TI$xme,
main=paste("Data set reconstructed from random points",
 "(from eg4 with noise) on x as function of t" ))
imagenan(TI$xsd,
main=paste("Std Dev in reconstruction from random points\n",
" (from eg4 with noise) on x as function of t" ))
#error values show where the system was not equitable
```

a_b_bagplot *Creates bagplots of equitable intercepts versus 1-slope*

Description

Bagplots are made by varying the zero of ther data set to various values the variable t(dependent on options). If community.f has some nonnumeric trait then bagplots are also created for each category of the trait. Plots are made based on the reference column chosen.

Usage

```
a_b_bagplot(community.f = NULL, Td, refindex = "Row_Ave",
  xlim = NULL, ylim = NULL, main = " ", rownum = NULL,
  facname = NULL, fall = FALSE)
```

Arguments

`community.f`	nonnumeric character traits of the individuals in x: default NULL
`Td`	Transform info based on transformE

refindex	intercept values based on this refences x location or individual: Default "Row_Ave"
xlim	vector of c(lowest range,highest range) for 1-slope values default NULL function defines them
ylim	vector of c(lowest range,highest range) for intercept values default NULL function defines them
main	text heading for plot
rownum	vector of points on t to use as zeroes default NULL (about 10 t values chosen)
facname	name of factor characteristic
fall	use all values of t Default=FALSE

Value

index of best intersection of plotted sequences

Examples

```
d<-eg8(rmult=4,cmult=4,mf=10,mg= 0,mu=10,sdf=0.75,sdg=1,sdu=0.2)
Td<-transformE(d=d)
aa<-a_b_bagplot(Td=Td,xlim=c(-0.3,0.3),ylim=c(-4,4))
aa<-a_b_bagplot(Td=Td,xlim=c(-0.3,0.3),ylim=c(-5,5),rownum=seq(1,ncol(Td$smat),by=2))
aa<-a_b_bagplot(Td=Td,xlim=c(-0.2,0.2),ylim=c(-5,5),fall=TRUE)
```

eg4 *Example 4 of equitable system f(x) product of sqrt of column index and very large scale wave: u(x) linear in column index g(x) sum of two waves*

Description

Example 4 of equitable system f(x) product of sqrt of column index and very large scale wave: u(x) linear in column index g(x) sum of two waves

Usage

```
eg4(rmult = 1, cmult = 1)
```

Arguments

rmult	resolution of sampling of t variable rmult*15 is the number of points sampled
cmult	resolution of sampling of x variable cmult*10 is the number of points sampled

Value

matrix of equitable data

Examples

```
d<-eg4(1,1)       # 15x10 data set
d<-eg4(1,10)      # 15x100 data set
d<-eg4(10,1)      # 150x10 data set
d<-eg4(15,15)    # 150x100 data set
```

eg5 *Example 5 of equitable system f(x) product of sqrt of column index and short wavelength wave: u(x) sum of linear in column index with short wavelength g(x) sum of two waves*

Description

Example 5 of equitable system f(x) product of sqrt of column index and short wavelength wave: u(x) sum of linear in column index with short wavelength g(x) sum of two waves

Usage

```
eg5(rmult = 1, cmult = 1)
```

Arguments

`rmult`	resolution of sampling of t variable rmult*15 is the number of points sampled
`cmult`	resolution of sampling of x variable cmult*10 is the number of points sampled

Value

matrix of equitable data

Examples

```
d<-eg5(1,1)       # 15x10 data set
d<-eg5(1,10)      # 15x100 data set
d<-eg5(10,1)      # 150x10 data set
d<-eg5(15,15)    # 150x100 data set
```

eg7 *Example 7 of equitable system f(x) Absolute value of linear function in column index u(x) 0 g(x) step function in row index*

Description

Example 7 of equitable system f(x) Absolute value of linear function in column index u(x) 0 g(x) step function in row index

Usage

```
eg7(rmult = 1, cmult = 1)
```

Arguments

`rmult`	resolution of sampling of t variable rmult*15 is the number of points sampled
`cmult`	resolution of sampling of x variable cmult*10 is the number of points sampled

Value

matrix of equitable data

Examples

```
d<-eg7(1,1)       # 15x10 data set
d<-eg7(1,10)      # 15x100 data set
d<-eg7(10,1)      # 150x10 data set
d<-eg7(15,15)    # 150x100 data set
```

`eg8` *Example 8 of equitable system*

Description

f(x) , g(t) amd u(x) are samopled from normal random distributions. g is then ordered from lowest to highest Good for simulating equitable phenology systems

Usage

```
eg8(rmult = 1, cmult = 1, mf = 10, mg = 0, mu = 10, sdf = 0.75,
  sdg = 1, sdu = 0.2)
```

Arguments

`rmult`	resolution of sampling of t variable rmult*15 is the number of points sampled
`cmult`	resolution of sampling of x variable cmult*10 is the number of points sampled
`mf`	10 (default) mean of f(x)
`mg`	0 (default) mean of g(x)
`mu`	10 (default) mean of u(x)
`sdf`	0.75 (default) standard deviation of f(x)
`sdg`	1 (default) standard deviation of g(x)
`sdu`	0.2 (default) standard deviation of u(x)

Details

f(x) random normal distribution with mean mf and standard deviation sdf u(x) random normal distribution with mean mu and standard deviation sdu g(x) random normal distribution with mean mg and standard deviation sdg

Value

matrix of equitable data

Examples

```
d<-eg8(rmult=4,cmult=4)       # 60x40 data set
 #transform this data set and then show bagplot intersetions
Td<-transformE(d=d)
aa<-a_b_bagplot(Td=Td,xlim=c(-0.25,0.25),ylim=c(-5,5))

d<-eg8(3,3)       # 45x30 data set
d<-eg8(1,10)      # 15x100 data set
d<-eg8(10,1)      # 150x10 data set
d<-eg8(15,15)    # 150x100 data set
```

egsimple *Example "simple" of equitable system f(x) , g(t) amd u(x) are define directly as vectors to reate equitable system*

Description

Example "simple" of equitable system f(x) , g(t) amd u(x) are define directly as vectors to reate equitable system

Usage

```
egsimple(f = c(2, 4, 6, 8), g = c(-6, -3, 0, 1, 2), u = c(4, 3, 0,
  1))
```

Arguments

f	vector f(x) default c(2,4,6,8)
g	vector g(t) default c(-6,-3,0,1,2)
u	vector u(x) of same length as default f c(4,3,0,1) when NULL it is set to 0

Value

matrix of equitable data

Examples

```
egsimple()       # u(x)=c(4,3,0,1)
egsimple(u=NULL)      # u(x)=0
d<-egsimple(f=c(1,2,3),g=c(-1,0,1,2,3),u=c(5,-2,6))
d<-egsimple(f=c(1,2,3),g=c(-1,0,1,2,3),u=NULL)
```

imagenan *Plots False Colour Image*

Description

Displays a false colour image of two dimensional data set

Usage

```
imagenan(x, yline = 3, yma = 5, col = topo.colors(255),
  outside.below.color = "black", outside.above.color = "white",
  na.color = "gray", ...)
```

Arguments

`x`	2D data to display
`yline`	Number of lines in (left side) to where image begins
`yma`	Number of characters in (left side)
`col`	colour table
`outside.below.color`	Colour below threshold
`outside.above.color`	Colour above threshold
`na.color`	Colour of NAs (see `par`)
`...`	other plot parameters eg zlim list with min and max range of color table data e.g. zlim=c(2,5)

Value

None

Examples

```
d<-eg4()
imagenan(d)
imagenan(d,zlim=c(2,8))
imagenan(d,zlim=c(3,8),outside.above.color='red',outside.below.color='tan')
d=data.frame(cbind(c(1:4),c(2,5,NA,NA) ))
imagenan(d)
```

make_A *Construct equitable slope matrix from f(x) : image is displayed of matrix*

Description

Given f(x) assumed to be defined relative to x=xref, an equitable matrix is constructed

Usage

```
make_A(xref = 1, f)
```

Arguments

xref	1 (default) x reference value around which the matrix is constructed
f	a vector f(x) around which the equitbel matirx is constructed

Value

Equitable matrix

Examples

```
make_A(xref=1,f=c(1,2,3))  #simple example of equitable matrix
make_A(xref=3,f=c(1,2,3))  #reference not important
plot(eg4(3,3)[,1])
A<-make_A(xref=1,f=eg4(3,3)[,1])
#more complicated matrix based on large scale wave f(1)g(t)  in eg4
A<-make_A(xref=2,f=eg4(3,3)[20,])
 #more complicated matrix based on large scale wave f(x)g(20) in eg4
A<-make_A(f=eg5(6,6)[,1]) #more complicated matrix based on small scale wave f(1)g(t)  in eg5
plot(eg5(10,10)[20,],type="b")
 #more complicated matrix based on small scale wave f(x)g(20) in eg5
A<-make_A(xref=2,f=eg5(10,10)[20,])
 #low resolution  matrix based on small scale wave f(x)g(20) in eg5
A<-make_A(xref=2,f=eg5(2,2)[20,])
#more complicated matrix based on small scale wave f(20)g(t) in eg7
A<-make_A(xref=2,f=eg7(10,10)[,20])
#more complicated matrix based on small scale wave f(x)g(20) in eg7
A<-make_A(xref=2,f=eg7(10,10)[20,])
```

make_B *Construct Shift equitable matrix*

Description

Finds a shift equitbale matrix from a slope equitbel matrix and a sequence profile

Usage

```
make_B(xref = 1, orig, A, uflag = FALSE, pflag = TRUE)
```

Arguments

`xref`	Reference "location" to base shift matrix default set to 1 (for PCA dominant is first)
`orig`	2D Data set used when uflag FALSE profile at t=constant when uflag=TRUE (u(x) equivalent)
`A`	Slope equitable matrix
`uflag`	TRUE(default) or FALSE TRUE uses origas u(x): FALSE uses average profile (averaged over rows)
`pflag`	ignore

Value

B shift matrix B

Examples

```
d<-eg4(2,2); Td<-transformE(d)
B<-make_B(orig=Td$ET.x, A=Td$E.s)
B<-make_B(xref=5,orig=Td$ET.x, A=Td$E.s)
B<-make_B(xref=5,orig=Td$ET.x[1,], A=Td$E.s,uflag=TRUE)
```

`nastat` *mean and std dev on NA data set*

Description

Finds mean and standard deviaiton of 2 dimensional data set that contains NA values

Usage

```
nastat(d)
```

Arguments

`d`	data set

Value

list of mean $m and standard deviation $std

Examples

```
nastat(d=data.frame(cbind(c(1:4),c(2,5,NA,NA) )))  #
nastat(eg4())   # stats and image of example 4
nastat(eg7())   # stats and image of example 7
```

plotsome *Various types of plots of Equitable transform data dependent on options chosen*

Description

Uses functions plotimages plotindiv plotO_S_E_lscol plotversus plot_vsref. The function plotsummary uses this function to summarize the transform data

Usage

```
plotsome(T, num = 5, signal = NULL, limits = NULL, xlimits = NULL,
  indiv = FALSE, columns = FALSE, images = TRUE, density = FALSE,
  versus = FALSE, xvsref = NULL, yline = 3, yma = 5,
  transpose = FALSE, of = FALSE, lf = FALSE, ef = TRUE,
  avef = FALSE, errb = FALSE, row_unit = NULL, col_unit = NULL,
  z_unit = NULL, genname = NULL, stderror = FALSE, fcontour = TRUE)
```

Arguments

T	equitable transform info: output from transformE
num	5 default number of indiviual plot to be made
signal	2D data set representing the signal. Must be the same size as T$smat
limits	default NULL limits on yaxis of plots and range for rainbow colouring in images
xlimits	default NULL limits on xaxis of plots
indiv	default FALSE: plots individual row or column dependendent on transpose flag
columns	default FALSE TRUE:plot all columns on one plot
images	default TRUE: plots false colour images
density	ignore
versus	plots all individual vs reference default FALSE:
xvsref	column to be used as reference against which all others are plotted default NULL
yline	default 3 lines out from plot to display yaxis values
yma	default 5 cahracters out from plot to display ylabel
transpose	default FALSE TRUE: plot all rows on one plot
of	according to above flags plots original data default FALSE
lf	according to above flags plots least squares transformdata default FALSE
ef	according to above flags plots equitable transform data default FALSE
avef	according to above flags plots equitbale transform data formed using only average profile default FALSE
errb	according to above flags uses error bars when possible default FALSE
row_unit	row axis label default Row Number
col_unit	column axis label default Row Number
z_unit	label for quantity measured in data
genname	main title to be included
stderror	according to above flags uses error bars of standard error rather than standard deviaiton when possible default FALSE
fcontour	default TRUE: plots contour maps

Value

None

Examples

```
d<-eg4(2,2)
Td<-transformE(d)
d_noise<-d+rnorm(prod(dim(d)),mean=0,sd=(1/4*sd(d,na.rm=TRUE)))
Td_noise<-transformE(d_noise)
plotsome(T=Td_noise,lf=TRUE,of=TRUE)
# default only plots imagef of signal original and equitable
plotsome(T=Td_noise,signal=Td$smat,of=TRUE)
plotsome(T=Td_noise,signal=Td$smat,indiv=TRUE,xvsref=ncol(Td$smat))
plotsome(T=Td_noise,signal=Td$smat,columns=TRUE,images=FALSE,
lf=TRUE,of=TRUE)
plotsome(T=Td_noise,signal=Td$smat,columns=TRUE,images=FALSE,
transpose=TRUE,lf=TRUE,of=TRUE)
plotsome(T=Td_noise,signal=Td$smat,indiv=TRUE,of=TRUE,lf=TRUE)
plotsome(T=Td_noise,signal=Td$smat,indiv=TRUE,of=TRUE,errb=TRUE)
plotsome(T=Td_noise,signal=Td$smat,indiv=TRUE,of=TRUE,errb=TRUE,stderror=TRUE)
#num= NULL  all individuals are plotted
plotsome(T=Td_noise,signal=Td$smat,indiv=TRUE,of=TRUE,num=NULL)
# plots images and transforms vs original and signal
plotsome(T=Td_noise,signal=Td$smat,of=TRUE,lf=TRUE,versus=TRUE)
```

plotsummary *Summary plots of transforms*

Description

Plots of various output from a transform: shows both Equitable and Least squares results Will compare them to the original data and to a signal is it is available Various formats for displaying the dat are used including images, contours and row/column plots with error bars showbn

Usage

```
plotsummary(Td_noise, Td = NULL, Td_old = NULL, row_unit = NULL,
  col_unit = NULL, z_unit = NULL, yline = 3, yma = 5,
  fintersect = FALSE, fsquares = FALSE, fave = FALSE, fall = FALSE,
  inc = NULL)
```

Arguments

`Td_noise` Output from the transform function transformE fro the data to be studied

`Td` NULL(Default) Output from the transform program for an underlying signal. Allows comparisons with undelrlying signal

`Td_old` ignore

`row_unit` name for the row dimension for axis plotting e.g. "Day number"

`col_unit` name for the row dimension for axis plotting e.g. "Year" `z_unit` name for the measured quantity e.g. "Temperature"

`yline` 3 (default) number of lines from image to start ylabel

`yma` 5 (default) distance in from margin to start images

`fintersect` FALSE (default) TRUE: plots intercept vs 1-slope for different zeroes

`fsquares` TRUE (default) line p[lots] produced of slope and shift square matrices

`fave` FALSE (default) TRUE: shows results with errors of performing averaging ion data

`fall` FALSE (default) TRUE: all "events are used to construct bagplots when finterswect is also TRUE

`inc` NULL(default) 10 colums plotted : inc when set is the increment in columns that the plots step through

Value

None

Examples

```
# first create a data set d and create the associated transforms.
# In this case d is eg7 with a resolution 3x higher than the lowest
#consider putting the graphs into a pdf file  by bracketing your
#commands beginning with pdf(file=”foo.pdf) and ending with dev.off()
#(includes last column as average
# sequence profile : use Ave=FALSE to eliminate this column )
d<-eg7(3,3);Td<-transformE(d)
#when the data is perfectly equitable many plots
#are identical for the different transforms
plotsummary(Td)
#points (even for average profile) have no error
# in perfectly equitable system as they are specified by f,g, and u
plotsummary(Td,fave=TRUE)
               #add noise to this signal data set
#find the std dev of the overall signal and add normally distributed noise
sdd<-sd(d,na.rm=TRUE)
# that has a std. dev that is some fraction (fac) of this signal std dev
#set the fraction of noise relative to the standard deviaiton of the signal
fac<-1/3
#add to signal a normal distribution of noise with this std dev.
d_noise<-d+rnorm(prod(dim(d)),mean=0,sd=fac*sdd)
d_noise<-matrix(d_noise,nrow=nrow(d),ncol=ncol(d))
rownames(d_noise)<-rownames(d); colnames(d_noise)<-colnames(d)
Td_noise<-transformE(d_noise)  #transform the noisy data
#shows how the transform looks compared to the original data
plotsummary(Td_noise)
#shows how the data looks compared to the signal data
plotsummary(Td_noise,Td)
#change the label spacing on the images to fit in the yaxis numbers
plotsummary(Td_noise,yline=5,yma=8)
plotsummary(Td_noise,Td,yline=5,yma=10, fave=TRUE,
 row_unit=”Day Number”, col_unit=”Year”,
 z_unit=”Temperature (C)”,inc=1)
 # plot averages of the data /signal and
 #compare to averages with error due to equitable system
# 45x30 data set of 3 sets of random numbers coupled together
```

```
#in an equitable system
d<-eg8(3,3)
Td<-transformE(d,Ave=TRUE)
#data set entirely equitable but rows and column values have random distribution
plotsummary(Td_noise=Td,fave=TRUE)
# averages along rows and columns show large error but
#system is entirely specified by f,g,u
#no errors in knowing equitable average values as they are entiely
# specified in system
```

`runstatsNS` *mean and std dev of transform differenced with the signal data set*

Description

Finds mean and standard deviations of transforms and residuals between transfgorms and signal

Usage

```
runstatsNS(Tx, T_noise, extraf = FALSE)
```

Arguments

`Tx`	signal transform information from transformE
`T_noise`	noisy transform information from transformE
`extraf`	ignore

Value

None

Examples

```
#first construct transfor of data and trnasform of signal
d<-eg4(2,2)
Td<-transformE(d)
d_noise<-d+rnorm(prod(dim(d)),mean=0,sd=(1/4*sd(d,na.rm=TRUE)))
Td_noise<-transformE(d_noise)
runstatsNS(Tx=Td_noise,T=Td)  #
```

`transformE` *Equitable transform*

Description

Creates an equitable transform and returns information regarding it

Usage

```
transformE(d, Ave = TRUE, cAve = FALSE, Zero = FALSE, zero = NULL,
  diagonal = TRUE, imageplot = FALSE, old = NULL)
```

Arguments

`d`	2D data to be transformed
`Ave`	Include a new column with Row averages `"TRUE"`(Default) or `"FALSE"`
`cAve`	Include a new row with Column averages `"TRUE"` or `"FALSE"`(Default)
`Zero`	if TRUE subtract a zero from data set. Default FALSE
`zero`	Value to be subtracted off of all data when transforming Default=0
`diagonal`	Include diagonals of matrices when transforming `"TRUE"`(Default) or `"FALSE"`
`imageplot`	TRUE plots image of data Default FALSE
`old`	Default NULL (could use old transform slopes to mask new slope transform)

Value

Output from the TransformE function

Running Td<-transformE(d) gives Td that contains several variables (see summary(Td))

(access variables via Td$variable_name).

Td$smat is original data set :view using imagenan(Td$smat)

Equitable Transform: Td$ET.x matrix of transformed data :view using imagenan(Td$ET.x)

least squared Transform: Td$l.s.x :view using imagenan(Td$l.s.x)

Equitable Transform based only on average column: Td$Ave.ET.x (assumes Ave=TRUE)

l.s. prefix indicates LEAST SQUARES result

s=slope, b=intercept sse =std error of slope bse=std error of intercept

r2=coef. of determination N: # of points in fit pslope: prob. for no correlation

node== indication if fit is due to a node p_par:sequences approximately parallel

zero: subtracted value

Examples of LEAST SQUARES variables:

l.s.s l.s.sse

l.s.b l.s.bse l.s.r2 l.s.N l.s.pslope l.s.node l.s.p_par l.s.zero

# e.g. view using imagenan(Td$l.s.s)

l.s.x= Least squared transform std. dev. errors at each point: l.s.xsd

l.s.Es l.s.Eb l.s.Ep : masked matrices of best values included slope=Es intercept=Eb prob= Ep

E prefix indicates EQUITABLE result

Equitable slopes: E.s intercepts: E.b

errors: E.rtestxsd, E.rtestbsd

#view using imagenan(Td$E.s)

with convergence information (E.rtestxm,E.rtestbm) Values of r^2 after convergence functions

E.numrun: # runs to get convergence

first iteration slopes/intercepts: (E.s1,E.b1) with std. dev. errors (E.sd1, E.bsd1)

E.s E.numrun E.rtestxm E.rtestxsd E.s1 E.sd1 E.sN E.snode

E.b E.numrun E.rtestbm E.rtestbsd E.b1 E.bsd1 E.bN

ET.x: Equitable tranform () with std. dev. errors at each point: ET.xsd

ET.N: number of points averaged to get point

ET.x ET.xsd ET.N

ET.Es ET.Eb ET.Ep : masked matrices included slope=Es interceprt=Eb prob= Ep

Ave.ET.x Ave.ET.xsd Ave.ET.N Ave.ET.Es Ave.ET.Eb Ave.ET.Ep

transform based on only average column: masked matrices included

Examples

```
# Find a transform using a signal with no noise and then
# add noise and show the results for a noisy data set.
# A researcher with a data set can simply use transformE and plotsummary
#  on their data set placed in variable d
# d is an example (4) of a two dimensional separable signal
d<-eg4(3,3)
Td<-transformE(d, Ave=TRUE)  #Run and equitable transform on the data
#creates summary plots of the data comparing sequences in various ways
plotsummary(Td)
             #add noise to this signal data set
#find the std dev of the overall signal and add normally distributed noise
# that has a std. dev that is some fraction (fac) of this signal std dev
#let the fraction be 1/2 the standard deviaiton of the signal
#add to signal a normal distribution of noise with this std dev.
d_noise<-d+rnorm(prod(dim(d)),mean=0,sd=1/2*sd(d,na.rm=TRUE))
      # Once you have a data set (named d_noise) and
      #you want to find if there is an underlying pattern, run the transform
#Ave= TRUE includes an additional sequence of averages, if not desired set to FALSE
Td_noise<-transformE(d_noise, Ave=TRUE)
 #summary plots of the transform data compared to the thwe original
plotsummary(Td_noise)
 # if you want to include units for the rows columns and data add in the units
plotsummary(Td_noise,row_unit="Year", col_unit="Day Number",z_unit="Temperature")
#shows statistics relative to Signal :Cnonly run this if you already have a signal
runstatsNS(Td,Td_noise)
#shows signal along with noisy data and transforms
#plotsummary puts togetwher numerous calls to plotsome, and plotsquares
plotsummary(Td_noise,Td)
#shows signal along with noisy data and transforms
plotsummary(Td_noise,Td)
#plotsummary puts togetwher numerous calls to plotsome, and plotsquares

plotsummary(Td_noise,Td,row_unit="Year", col_unit="Day Number",z_unit="Temperature")
plotsummary(Td_noise,Td,row_unit="Year", col_unit="Day Number",z_unit="Temperature")

#if you already know the "signal" (in d), create the separable
# Transforms into Td
      #Any of the above stepts could be run with data sets
      # having less than about 450 sequences
      # other examples include eg0, eg1,eg2,eg3,e4,eg5, eg6,eg7,eg8,egrand
      #resolutions of these examples can be altered by changing rmult and cmult
d<-eg5(1,1)
d<-eg5(2,2)
d<-eg5(5,5)
d<-eg5(15,15)
d<-eg5(1,15)
d<-eg5(15,1)
```

xvsrefplot *Plots individual sequences against a reference sequence*

Description

Equitable transform with errors, original and Equitable transform line parameters can be shown If number of point in variable t is less than 26 then each t point is colured and labelled in the legend

Usage

```
xvsrefplot(Td, cgroup = NULL, ylim = NULL, ref = NULL, br = "",
  pf = TRUE, extranames = NULL, err95 = TRUE, fitl = TRUE)
```

Arguments

Td	transform information from transformE
cgroup	cgroup is the vector of columns to be plotted. Default NULL;10 colums across matrix are plotted
ylim	vector of min and max for plots to display default (NULL) function calculates same for all
ref	reference column against which all columns are plotted. Default NULL; column index with largest variation
br	text for headings of plots
pf	plotflag default TRUE
extranames	extra names to use for legend of t points
err95	2*std. dev/sqrt(N) where std deviation on the slope is found when calculating equitable slopes Default=TRUE
fitl	Default TRUE. Fitted line is displayed

Value

column index with largest variation amongst columns that were plotted

Examples

```
#first construct transfor of data and transform of signal
d<-eg4(1,2)
d_noise<-d+rnorm(prod(dim(d)),mean=0,sd=(1/4*sd(d,na.rm=TRUE)))
Td_noise<-transformE(d_noise)
xvsrefplot(Td=Td_noise)
xvsrefplot(Td=Td_noise,ref="Row_Ave")
nc<-seq(1,ncol(Td_noise$smat), by=1)
xvsrefplot(Td=Td_noise,cgroup=nc,ref=ncol(Td_noise$smat),
br=paste0( "Equitable Profiles"))
xvsrefplot(Td=Td_noise,cgroup=c(1,4,9,12),ref=7,
br=paste0( "Equitable Profiles"))
xvsrefplot(Td=Td_noise,cgroup=c(1,4,9,12),ref=7,
br=paste0( "Equitable Profiles with std dev"),fitl=FALSE)
xvsrefplot(Td=Td_noise,cgroup=c(12),ref=7,
br=paste0( "Equitable Profiles"))
xvsrefplot(Td=Td_noise,cgroup=c(12),ref=7,
br=paste0( "Equitable Profiles"),err95=FALSE)

d<-eg5(3,3)
d_noise<-d+rnorm(prod(dim(d)),mean=0,sd=(1/2*sd(d,na.rm=TRUE)))
Td_noise<-transformE(d_noise)
xvsrefplot(Td=Td_noise)
```

Index