Package 'mistat'

April 18, 2021

Type Package

Title Data Sets, Functions and Examples from the Book: ``Modern Industrial Statistics'' by Kenett, Zacks and Amberti

Version 2.0.3

Date 2021-04-18

Author Daniele Amberti

Depends R (>= 3.5)

Imports graphics, grDevices, methods, stats, utils

Suggests e1071, mvtnorm, AcceptanceSampling, boot, car, Dodge, tseries, qcc, DoE.base, FrF2, rsm, LearnBayes, ggplot2, grid, DiceEval, DiceKriging, DiceDesign, DiceView, lhs, survival, rpart, fdapace, randomForest, xgboost

Maintainer Daniele Amberti <daniele.amberti@gmail.com>

Description Provide all the data sets and statistical analysis applications used in ``Modern Industrial Statistics: with applications in R, MINITAB and JMP'' by R.S. Kenett and S. Zacks with contributions by D. Amberti, John Wiley and Sons, 2021, which is a third revised and expanded revision of ``Modern Industrial Statistics: Design and Control of Quality and Reliability'', R. Kenett and S. Zacks, Duxbury/Wadsworth Publishing, 1998.

License GPL (>= 2)

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2021-04-18 13:20:03 UTC

R topics documented:

`mistat-package` *The Modern Industrial Statistics Package*

Description

This R package is providing all the data sets and statistical analysis of *Modern Industrial Statistics, with applications using R, MINITAB and JMP* by R.S. Kenett and S. Zacks with contributions by D. Amberti, John Wiley and Sons, 2013. This second revised and expanded second edition.

Details

Package:	mistat
Type:	Package
Date:	2012-08-22
License:	GPL >= 2

Author(s)

Daniele Amberti

Maintainer: Daniele Amberti <amberti@inwind.it>

See Also

Bootstrap Resampling, Quality Control Charts, Operating Characteristics of an Acceptance Sampling Plan, Quality Control Charts, Fractional Factorial 2-level designs.

Examples

```
data(OELECT)
data(OELECT1)

randomizationTest(list(a=OELECT, b=OELECT1),
                  R=500, calc=mean,
                  fun=function(x) x[1]-x[2],
                  seed=123)

Ps <- pistonSimulation(
  m = rep(60, 100),
  s = rep(0.02, 100),
  v0 = rep(0.01, 100),
  k = rep(5000, 100),
  p0 = rep(110000, 100),
  t = c(rep(296,35), 296*1.1^(1:65)),
  t0 = rep(360, 100),
  each = 1,
  seed = 123,
  check = FALSE)

head(Ps)

cusumArl(mean= 0.0,
         N=100,
         limit=5000,
         seed=123)

powerCircuitSimulation(seed=123, each=3)

set.seed(123)

Ttf <- rgamma(50,
              shape=2,
              scale=100)

Ttr <- rgamma(50,
              shape=2,
              scale=1)

AvailEbd <- availDis(ttf=Ttf,
                     ttr=Ttr,
                     n=1000, seed=123)


RenewEbd <- renewDis(ttf=Ttf,
                     ttr=Ttr,
```

```
time=1000,
n=1000)
```

ABC *ABC*

Description

A customer satisfaction survey, ABC.csv. The data consists of 266 responses to a questionnaire with a question on overall satisfaction (q1) and responses to 125 other questions. Variables named as "q" and a number are satisfaction or agreement levels on a 0-5 scale, Variables named as "qi" represent importance on a 1-3 scale.

Usage

```
data("ABC")
```

Format

A data frame with 266 observations on the following 134 variables.

- `q1` a numeric vector
- `q4` a numeric vector
- `q5` a numeric vector
- `q6` a numeric vector
- `qi6` a numeric vector
- `q7` a numeric vector
- `qi7` a numeric vector
- `q8` a numeric vector
- `qi8` a numeric vector
- `q9` a numeric vector
- `qi9` a numeric vector
- `q10` a numeric vector
- `qi10` a numeric vector
- `q11` a numeric vector
- `q12` a numeric vector
- `qi12` a numeric vector
- `q13` a numeric vector
- `qi13` a numeric vector
- `q14` a numeric vector
- `qi14` a numeric vector
- `q15` a numeric vector

`qi15` a numeric vector

`q16` a numeric vector

`qi16` a numeric vector

`q17` a numeric vector

`q18` a numeric vector

`qi18` a numeric vector

`q19` a numeric vector

`qi19` a numeric vector

`q20` a numeric vector

`qi20` a numeric vector

`q21` a numeric vector

`qi21` a numeric vector

`q22` a numeric vector

`qi22` a numeric vector

`q23` a numeric vector

`qi23` a numeric vector

`q24` a numeric vector

`qi24` a numeric vector

`q25` a numeric vector

`q26` a numeric vector

`qi26` a numeric vector

`q27` a numeric vector

`qi27` a numeric vector

`q28` a numeric vector

`qi28` a numeric vector

`q29` a numeric vector

`qi29` a numeric vector

`q30` a numeric vector

`qi30` a numeric vector

`q31` a numeric vector

`q32` a numeric vector

`qi32` a numeric vector

`q33` a numeric vector

`qi33` a numeric vector

`q34` a numeric vector

`qi34` a numeric vector

`q35` a numeric vector

qi35 a numeric vector

q36 a numeric vector

qi36 a numeric vector

q37 a numeric vector

qi37 a numeric vector

q38 a numeric vector

q39 a numeric vector

q40 a numeric vector

q41 a numeric vector

qi41 a numeric vector

q42 a numeric vector

q43 a numeric vector

q44 a numeric vector

q45 a numeric vector

qi45 a numeric vector

q46 a numeric vector

qi46 a numeric vector

q47 a numeric vector

qi47 a numeric vector

q48 a numeric vector

qi48 a numeric vector

q49 a numeric vector

q50 a numeric vector

qi50 a numeric vector

q51 a numeric vector

qi51 a numeric vector

q52 a numeric vector

qi52 a numeric vector

q53 a numeric vector

qi53 a numeric vector

q54 a numeric vector

qi54 a numeric vector

q55 a numeric vector

qi55 a numeric vector

q56 a numeric vector

qi56 a numeric vector

q57 a numeric vector

`q58` a numeric vector

`qi58` a numeric vector

`q59` a numeric vector

`qi59` a numeric vector

`q60` a numeric vector

`qi60` a numeric vector

`q61` a numeric vector

`qi61` a numeric vector

`q62` a numeric vector

`qi62` a numeric vector

`q63` a numeric vector

`qi63` a numeric vector

`q64` a numeric vector

`qi64` a numeric vector

`q65` a numeric vector

`q66` a numeric vector

`qi66` a numeric vector

`q67` a numeric vector

`q68` a numeric vector

`q70` a numeric vector

`q71` a numeric vector

`q72` a numeric vector

`q73` a numeric vector

`q74` a numeric vector

`q75` a numeric vector

`q76` a numeric vector

`q77` a numeric vector

`q78` a numeric vector

`q79` a numeric vector

`q80` a numeric vector

`q81` a numeric vector

`var1` Continent, a factor with levels `Europe`

`var3` Country, a factor with levels `Benelux` `France` `Germany` `Italy` `UK` `Israel`

`var4` Segmentation, a factor with levels `Other` `Silver` `Gold` `Platinum`

`var6` Age of ABC's equipment, a factor with levels `less than 1` `1-2` `2-3` `3-4` `more than 4`

`var9` Profitability, a factor with levels `Profitable` `Break-Even` `Below Break-Even`

`var11` Position, a factor with levels `Owner` `Management` `Technical management` `Technical staff` `Operator` `Administrator` `Other`

`customerSeniority` a numeric vector

`country` a factor with levels `Benelux` `France` `Germany` `Israel` `Italy` `UK`

Source

Kenett, R and Salini, S. (2013) *Modern Analysis of Customer Surveys*.

References

The ABC Company, Questionnaire and Data

Examples

```
data(ABC)
barplot(table(ABC$q1, ABC$q4))
```

ALMPIN *Aluminium Pins (6 dimensions)*

Description

Records of 6 dimension variables (a subset of 2 in ALMPIN2) measured in mm on 70 alluminium pins used in airplanes, in order of production.

Usage

```
data(ALMPIN)
```

Format

A data frame with 70 observations on the following 6 variables.

`diam1` pin diameter at specified location, a numeric vector

`diam2` pin diameter at specified location, a numeric vector

`diam3` pin diameter at specified location, a numeric vector

`capDiam` diameter of the cap on top of the pin, a numeric vector

`lenNocp` length of the pin without the cap, a numeric vector

`lenWcp` length of the pin with the cap, a numeric vector

Details

The aluminum pins are inserted with air-guns in pre-drilled holes in order to combine critical airplane parts such as wings, engine supports and doors.

The measurements were taken in a computerized numerically controlled (CNC) metal cutting operation. The six variables are Diameter 1, Diameter 2, Diameter 3, Cap Diameter, Lengthncp and Lengthwcp. All the measurements are in millimeters. The first three variables give the pin diameter at three specified locations. Cap Diameter is the diameter of the cap on top of the pin. The last two variables are the length of the pin, without and with the cap, respectively.

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(ALMPIN)

cor(ALMPIN)

plot(ALMPIN)
```

ARMA *Random realization of ARMA prcess*

Description

Creates a random realization an of ARMA prcess.

Usage

```
ARMA(n, a, b)
```

Arguments

n	length of the ARMA process to be generated.
a	vector p of parameters.
b	vector q of parameters.

Value

a vector with values from the simulated AMRA process of lenght n + max(p, q)

Author(s)

Shelemyahu Zacks

See Also

predARMA

Examples

```
ARMA(100,c(0.1, 0.2, 0.3), c(0.1, 0.2))
```

`availDis` *Availability Distribution*

Description

Provide the Empirical Bootstrap Distribution of the asymptotic availability index A_∞, based on observed samples of failure times and repair times.

Usage

```
availDis(ttf, ttr, n, seed = NA, printSummary = TRUE)
```

Arguments

`ttf`	numeric vector of Time To Failure
`ttr`	numeric vector of Time To Repair
`n`	the number of bootstrap replicates
`seed`	a single value, interpreted as an integer. If specified make the simulation replicable.
`printSummary`	logical, if `TRUE` print the Mean Time To Failure, Mean Time To Repair and the asymptotic availability

Value

A numeric vector of lenght `n` with simulated availabilities

Author(s)

Daniele Amberti

References

Kenett, R., Zacks, S. with contributions by Amberti, D. *Modern Industrial Statistics: with applications in R, MINITAB and JMP*. Wiley.

See Also

`renewDis`

Examples

```
set.seed(123)

Ttf <- rgamma(50,
              shape=2,
              scale=100)
```

```
Ttr <- rgamma(50,
              shape=2,
              scale=1)

AvailEbd <- availDis(ttf=Ttf,
                     ttr=Ttr,
                     n=1000)
```

BLEMISHES *Number of Blemishes on Ceramic Plates*

Description

Blemishes found on each of 30 ceramic plates.

Usage

```
data(BLEMISHES)
```

Format

A data frame with 30 observations:

`plateID` a factor

`count` an integer vector

Details

Blemishes will affect the final product's (hybrid micro electronic components) electrical performance and its overall yield

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(BLEMISHES)
table(factor(BLEMISHES$count, levels=0:5))
```

CAR *Car*

Description

Records on 109 different car models, including number of cylinders, origin, turn diamater, horsepower, and number of miles per gallon in city driving.

Usage

```
data(CAR)
```

Format

A data frame with 109 observations on the following 5 variables.

`cyl` Number of cylinders, an integer vector

`origin` Car origin, 1 = US; 2 = Europe; 3 = Asia, an integer vector

`turn` Turn diamater, a numeric vector

`hp` Horsepower, a numeric vector

`mpg` Miles per gallon in city driving, a numeric vector

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(CAR)

with(data=CAR, expr=table(cyl, origin))
```

COAL *Number of Coal Mine Disasters*

Description

Data on the number of coal mine disasters (explosions) in England, per year, for the period 1850 to 1961.

Usage

```
data(COAL)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(COAL)
Bp <- barplot(COAL)

axis(side=1,
     labels=seq(
       from=1850,
       to=1960,
       by=10),
     at=Bp[rep(c(TRUE,
                 rep(FALSE, 9)),
               10)])

rm(Bp)
```

COMPURESP *Computer Response Time Optimization*

Description

The experiment described here was part of an extensive effort to optimize a UNIX operating system.

Usage

```
data(COMPURESP)
```

Format

A data frame with 18 observations on the following 10 variables.

F a factor with levels 1 2, representing KMCs used

B a factor with levels 1 2 3, representing File Distribution

C a factor with levels 1 2 3, representing Memory Size

D a factor with levels 1 2 3, representing System Buffers

E a factor with levels 1 2 3, representing Sticky Bits

A a factor with levels 1 2 3, representing Disk Drives

G a factor with levels 1 2 3, representing INODE Table Entries

H a factor with levels 1 2 3, representing Other System Tables

Mean mean time taken for the system to complete commands execution

SN S/N ratio $\eta = -10 log_{10}(\frac{1}{n}\sum_i y_i^2)$

Details

The experiment described here was part of an extensive effort to optimize a UNIX operating system running on a VAX 11-780 machine. The machine had 48 user terminal ports, two remote job entry links, four megabytes of memory, and five disk drives. The typical number of users logged on at a given time was between 20 to 30.

1. Problem Definition. Users complained that the system performance was very poor, especially in the afternoon. The objective of the improvement effort was to both minimize response time and reduce variability in response.

2. Response variable. In order to get an objective measurement of the response time two specific representative commands called 'standard' and 'trivial' were used. The 'standard' command consisted of creating, editing and removing a file. The 'trivial' command was the UNIX system 'date' command. Response times were measured by submitting these commands every 10 minutes and clocking the time taken for the system to complete their execution.

Source

Pao, Phadke and Sherrerd (1985)

Examples

```
data(COMPURESP)

layout(matrix(1:4, 2, byrow=TRUE))

with(COMPURESP,
     interaction.plot(
       x.factor=F,
       trace.factor=rep(0, length(F)),
       response=SN,
       legend=FALSE,
       type="b",
       pch=15:18,
       ylim=c(-17, -10)))

with(COMPURESP,
     interaction.plot(
       x.factor=B,
       trace.factor=rep(0, length(B)),
       response=SN,
       legend=FALSE,
       type="b",
       pch=15:18,
       ylim=c(-17, -10)))

with(COMPURESP,
     interaction.plot(
       x.factor=C,
       trace.factor=rep(0, length(C)),
       response=SN,
       legend=FALSE,
```

```
        type="b",
        pch=15:18,
        ylim=c(-17, -10)))

with(COMPURESP,
     interaction.plot(
       x.factor=D,
       trace.factor=rep(0, length(D)),
       response=SN,
       legend=FALSE,
       type="b",
       pch=15:18,
       ylim=c(-17, -10)))

layout(1)
```

CONTACTLEN *Length of the Electrical Contacts*

Description

length (in cm) of the electrical contacts of relays in samples of size five, taken hourly from a running process.

Usage

```
data(CONTACTLEN)
```

Format

A numeric matrix with five columns representing a sample and twenty rows representing hourly samples.

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(CONTACTLEN)

library(qcc)

qcc(CONTACTLEN, type="xbar")
```

cusumArl — *Cumulative Sum Control Charts Average Run Length*

Description

Computes the ARL function by simulation

Usage

```
cusumArl(..., randFunc = rnorm, N = 100, limit = 10000, seed = NA,
         kp = 1, km = -1, hp = 3, hm = -3, side = "both",
         printSummary = TRUE)
```

Arguments

`...`	arguments such as mean, lambda or sd to be passed to the appropriate random genneration function
`randFunc`	a random generation function
`N`	the number of replicates
`limit`	safety parameter, stop rule for procedures with very long ARL
`seed`	a single value, interpreted as an integer. If specified make the simulation replicable.
`kp`	K^+ parameter of the control scheme
`km`	K^- parameter of the control scheme
`hp`	h^+ parameter of the control scheme
`hm`	h^- parameter of the control scheme
`side`	a character string specifying the side of the control scheme, must be one of "both" (default), "upper" or "lower"
`printSummary`	logical, if `TRUE` print a summary of the cusum ARL

Value

a list with elements:

`rls`	a numeric vector representing the Run Length of the simulation
`statistics`	a numeric vector with summary statistics
`run`	a list of length `N` elements each of which has single numeric elements `violationLower`, `violationUpper` and `rl`

Author(s)

Daniele Amberti

References

Kenett, R., Zacks, S. with contributions by Amberti, D. *Modern Industrial Statistics: with applications in R, MINITAB and JMP*. Wiley.

Examples

```
cusumArl(mean=1, seed=123, N=100, limit=1000)

cusumArl(size=100, prob=0.05, kp=5.95, km=3.92, hp=12.87, hm=-8.66,
  randFunc=rbinom, seed=123, N=100, limit=2000)

cusumArl(lambda=10, kp=12.33, km=8.41, hp=11.36, hm=-12.91,
  randFunc=rpois, seed=123, N=100, limit=2000)
```

`cusumPfaCed` *Cusum Probability of False Alarm and Conditional Expected Delay*

Description

Compute the Probability of False Alarm, PFA, and the Conditional Expected Delay, CED, for the Normal, Binomial and Poisson distributions

Usage

```
cusumPfaCedBinom(size0 = 0, prob0 = 1, size1 = 0, prob1 = 1,
                 tau = 10, N = 100, limit = 10000, seed = NA,
                 kp = 1, km = -1, hp = 3, hm = -3, side = "both",
                 printSummary = TRUE)

cusumPfaCedNorm(mean0 = 0, sd0=1, mean1=0, sd1=1,
                tau=10, N=100, limit=10000, seed=NA,
                kp=1, km=-1, hp=3, hm=-3, side="both",
                printSummary = TRUE)

cusumPfaCedPois(lambda0 = 0, lambda1=1,
                tau=10, N=100, limit=10000, seed=NA,
                kp=1, km=-1, hp=3, hm=-3, side="both",
                printSummary = TRUE)
```

Arguments

`size0`	number of trials (zero or more)
`prob0`	probability of success on each trial
`size1`	number of trials (zero or more) after a process level change
`prob1`	probability of success on each trial after a process level change
`mean0`	distribution mean

`sd0`	distribution standard deviation
`mean1`	distribution mean after a process level change
`sd1`	distribution standard deviation after a process level change
`lambda0`	(non-negative) mean
`lambda1`	(non-negative) mean after a process level change
`tau`	time on which the process level change occurs
`N`	the number of replicates
`limit`	safety parameter, stop rule for procedures with very long ARL
`seed`	a single value, interpreted as an integer. If specified make the simulation replicable.
`kp`	K^+ parameter of the control scheme
`km`	K^- parameter of the control scheme
`hp`	h^+ parameter of the control scheme
`hm`	h^- parameter of the control scheme
`side`	a character string specifying the side of the control scheme, must be one of "both" (default), "upper" or "lower"
`printSummary`	logical, if `TRUE` print a summary of the cusum PFA and CED

Value

a list with elements:

`rls`	a numeric vector representing the Run Length of the simulation
`statistics`	a numeric vector with summary statistics
`run`	a list of length `N` elements each of which has single numeric elements `violationLower`, `violationUpper` and `rl`

Author(s)

Daniele Amberti

References

Kenett, R., Zacks, S. with contributions by Amberti, D. *Modern Industrial Statistics: with applications in R, MINITAB and JMP*. Wiley.

Examples

```
cusumPfaCedNorm(mean1=1.5,
                tau=100,
                N=100,
                limit=1000,
                seed=123)
```

CYCLT *50 Cycle Times*

Description

50 cycle times (in seconds) of a piston operating at fixed operating conditions set at the minimal levels of seven control factors.

Usage

```
data(CYCLT)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(CYCLT)

summary(CYCLT)

plot(CYCLT, type="b")
```

DISS *Dissolution Data*

Description

Dissolution data of a new product and a reference approved product.

Usage

```
data(DISS)
```

Format

A data frame with 12 observations on the following 4 variables.

`batch` a factor with levels `REF` `TEST`

`tablet` a factor with levels `1` `2` `3` `4` `5` `6`

`min15` a numeric vector

`min90` a numeric vector

Source

Tsong et al., (1996).

Examples

```
data(DISS)
## maybe str(DISS) ; plot(DISS) ...
```

DISSOLUTION *DISSOLUTION*

Description

12 test and reference tablets measured under dissolution conditions at 5, 10, 15, 20, 30 and 45 seconds. The level of dissolution recorded for the tested generic product is ideally identical to the brand reference.

Usage

```
data("DISSOLUTION")
```

Format

A data frame with 144 observations on the following 4 variables.

`Product` a factor with levels `R` `T`

`Time` a numeric vector

`Label` a factor with levels `T1` `T10` `T10R` `T11` `T11R` `T12` `T12R` `T1R` `T2` `T2R` `T3` `T3R` `T4` `T4R` `T5` `T5R` `T6` `T6R` `T7` `T7R` `T8` `T8R` `T9` `T9R`

`Data` a numeric vector

Examples

```
data(DISSOLUTION)
Test <- subset(DISSOLUTION, Product == "T")
Test <- reshape(data = Test, v.names = "Data", timevar = "Label",idvar = "Time", direction = "wide")
summary(Test)
# library(fdapace)
# Model <- FPCA(Ly = Test[, grep("Data", colnames(Test))],
#  Lt = Test[, rep("Time", length(grep("Data", colnames(Test))))],
#  optns = list(maxK = 5, methodSelectK = 5))
# plot(Model)
```

dlmLg — *Dynamic Linear Model with linear growth*

Description

Dynamic Linear Model with linear growth.

Usage

```
dlmLg(x, C0, v, W, M0)
```

Arguments

`x`	a vector of data X_t.
`C0`	prior covariance matrix
`v`	prior variance of V.
`W`	prior variance of W.
`M0`	prior mean.

Value

a vector with values from the Dynamic Linear Model.

Author(s)

Shelemyahu Zacks

Examples

```
C0 <- matrix(c(0.22325, -0.00668, -0.00668, 0.00032), nrow = 2, byrow = TRUE)
W <- matrix(c(0.3191, -0.0095, -0.0095, 0.0004), nrow = 2, byrow = TRUE)
M0 <- matrix(c(134.234, -0.3115), nrow = 2)
v <- 0.1

data(DOW1941)
plot(DOW1941$Date, DOW1941$DOW1941,
     type="l",
     ylab="Dow Jones 1941", xlab="Date")

lines(DOW1941$Date, dlmLg(DOW1941$DOW1941, C0 = C0, v = v, W = W, M0 = M0))
```

DOJO1935 *Dow-Jones Financial Index 1935*

Description

The Dow-Jones financial index for the 300 business days of 1935.

Usage

```
data(DOJO1935)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(DOJO1935)

plot(DOJO1935,
     type="b",
     ylab="Dow Jones")
```

DOW1941 *Dow-Jones Financial Index 1941*

Description

The Dow-Jones daily index of 1941.

Usage

```
data("DOW1941")
```

Format

A data frame with 302 observations on the following 5 variables.

`DOW1941` Dow-Jones daily value

`Date` a POSIXlt

`Day` a progressive number

`DayOfWeek` a factor representing the weekday with levels `1 2 3 4 5 6`

`Month` a factor representing the month with levels `1 2 3 4 5 6 7 8 9 10 11 12`

Examples

```
data(DOW1941)

plot(DOW1941$Date, DOW1941$DOW1941,
     type="l",
     ylab="Dow Jones 1941", xlab="Date")
```

---

dynOAB — *Dynamic programming of the optimal One-Armed Bandits*

Description

Dynamic programming of the optimal One-Armed Bernoulli Bandits process

Usage

```
dynOAB(N, al)
```

Arguments

`N`	number of trials.
`al`	the known probability of reward on arm A.

Value

For dynOAB the matrix of maximal predicted rewards. For dynOAB2 the optimal predicted reward.

Author(s)

Shelemyahu Zacks

See Also

`simOAB`

Examples

```
dynOAB(10, 0.5)
dynOAB2(10, 0.5)
```

ELECFAIL — *Failures of an Electronic Device*

Description

50 failure times of an electronic device.

Usage

```
data(ELECFAIL)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(ELECFAIL)

hist(ELECFAIL)
```

ELECINDX — *Bernoulli Sample on OELECT Data*

Description

Bernoulli sample in which, we give a circuit in OELECT the value 1 if its electric output is in the interval (216, 224) and the value 0 otherwise.

Usage

```
data(ELECINDX)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

See Also

OELECT

Examples

```
data(ELECINDX)

qbinom(p=0.5, size=100, prob=mean(ELECINDX))
```

`ETCHRATE` *Data on the Rate of Etching*

Description

Rate of removal of field oxide in a semiconductor plasma etching process.

Usage

```
data(ETCHRATE)
```

Source

Digital Equipment Corporation (1988).

Examples

```
data(ETCHRATE)

hist(ETCHRATE)
```

`ETCHRATETWO` *Data on the Rate of Etching (two samples)*

Description

Rate of removal of field oxide in two different semiconductor plasma etching processes, A and B.

Usage

```
data(ETCHRATETWO)
```

Format

A data frame with 12 observations on the following 2 variables.

`A` a numeric vector, rate of etching, sample A

`B` a numeric vector, rate of etching, sample B

Source

Digital Equipment Corporation (1988).

Examples

```
data(ETCHRATETWO)

boxplot(values ~ ind, data=stack(ETCHRATETWO))
```

FAILTIME *Failure Times*

Description

Failure times of 20 electric generators (in hr).

Usage

```
data(FAILTIME)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(FAILTIME)

library(survival)

SuRe <- survreg(
  Surv(time=FAILTIME) ~ 1 ,
  dist = "exponential")

summary(SuRe)
```

FILMSP *Film Speed*

Description

Data gathered from 217 rolls of film. The data consists of the film speed as measured in a special lab.

Usage

```
data(FILMSP)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(FILMSP)

hist(FILMSP)
```

FLEXPROD *The Quinlan Experiment at Flex Products*

Description

Flex Products is a subcontractor of General Motors, manufacturing mechanical speedometer cables. The basic cable design has not changed for fifteen years and General Motors had experienced many disappointing attempts at reducing the speedometer noise level.

Usage

```
data(FLEXPROD)
```

Format

A data frame with 16 observations on the following 16 variables.

- `A` Liner O.D., a factor with levels `1` `2`
- `B` Liner Die, a factor with levels `1` `2`
- `C` Liner Material, a factor with levels `1` `2`
- `D` Liner Line Speed, a factor with levels `1` `2`
- `E` Wire Braid Type, a factor with levels `1` `2`
- `F` Braiding Tension, a factor with levels `1` `2`
- `G` Wire Diameter, a factor with levels `1` `2`
- `H` Liner Tension, a factor with levels `1` `2`
- `I` Liner Temperature, a factor with levels `1` `2`
- `J` Coating Material, a factor with levels `1` `2`
- `K` Coating Dye Type, a factor with levels `1` `2`
- `L` Melt Temperature, a factor with levels `1` `2`
- `M` Screen Pack, a factor with levels `1` `2`
- `N` Cooling Method, a factor with levels `1` `2`
- `O` Line Speed, a factor with levels `1` `2`
- `SN` Signal to noise ratio, a numeric vector

Details

Problem Definition: the product under investigation is an extruded thermoplastic speedometer casing used to cover the mechanical speedometer cable on automobiles. Excessive shrinkage of the casing is causing noise in the mechanical speedometer cable assembly.

Response variable: the performance characteristic in this problem is the post extrusion shrinkage of the casing. The percent shrinkage is obtained by measuring approximately 600mm of casing that has been properly conditioned (A), placing that casing in a two hour heat soak in an air circulating oven, reconditioning the sample and measuring the length (B). Shrinkage is computed as: $Shrinkage = 100 * (A - B)/A$.

Factor Levels: Existing (1) - Changed (2)

Number of Replications: four random samples of 600mm from the 3000 feet manufactured at each experimental run.

Data Analysis: signal to noise ratios (SN) are computed for each experimental run and analyzed using main effect plots and an ANOVA. Savings are derived from Loss function computations.

The signal to noise formula used by Quinlan is:

$$\eta = -10log_{10}(\frac{1}{n}\sum y^2)$$

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(FLEXPROD)

aov(SN ~ . , data=FLEXPROD)
```

`FOOD` *FOOD*

Description

Nutritional data from 961 different food items

Usage

```
data("FOOD")
```

Format

A data frame with 961 observations on the following 7 variables.

`Fat.grams` a numeric vector, fat in grams

`FoodEnergy.calories` a numeric vector, food energy in calories

`Carbohydrates.grams` a numeric vector, carbohydrates in grams

`Protein.grams` a numeric vector, protein in grams

`Cholesterol.mg` a numeric vector, cholesterol in milligrams

`Weight.grams` a numeric vector, weight in grams

`SaturatedFat.grams` a numeric vector, saturated fat in grams

Examples

```
data(FOOD)
plot(FOOD)
```

GASOL — *Distillation Properties of Crude Oils*

Description

32 measurements of distillation properties of crude oils.

Usage

```
data(GASOL)
```

Format

A data frame with 32 observations on the following 5 variables.

`x1` crude oil gravity (API), a numeric vector

`x2` crude oil vapour pressure (psi), a numeric vector

`astm` crude oil ASTM 10% point ($Fahrenheit$), a numeric vector

`endPt` gasoline ASTM endpoint ($Fahrenheit$), a numeric vector

`yield` yield of gasoline (in percentage of crude oil), a numeric vector

Source

Daniel and Wood (1971) pp. 165

Examples

```
data(GASOL)

LmYield <- lm(yield ~ 1 + astm + endPt,
              data=GASOL)

summary(LmYield)
```

GASTURBINE *Gas Turbine Cycle Times*

Description

125 gas turbine cycle times divided in 25 samples of 5 observations.

Usage

```
data(GASTURBINE)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(GASTURBINE)

plot(rowMeans(GASTURBINE), type="b")
```

HADPAS *Resistance Values of Hybrids*

Description

Several resistance measurements (Ω) of five types of resistances (Res 3, Res 18, Res 14, Res 7 and Res 20), which are located in six hybrid micro circuits simultaneously manufactured on ceramic substrates. There are altogether 192 records for 32 ceramic plates.

Usage

```
data(HADPAS)
```

Format

A data frame with 192 observations on the following 7 variables.

`diska` ceramic plate, a numeric vector

`hyb` hybrid micro circuit, a numeric vector

`res3` a numeric vector

`res18` a numeric vector

`res14` a numeric vector

`res7` a numeric vector

`res20` a numeric vector

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(HADPAS)

boxplot(HADPAS$res3 ~ HADPAS$hyb)
```

HYBRID *Resistance Values of Res 3*

Description

A subset of data in HADPAS, only variable `res3` is recorded. HYBRID contains values for hybrids 1 to 3, HYBRID1 contains hybrid 1 data and HYBRID2 contains values of hybrids 1 and 2.

Usage

```
data(HYBRID)
```

Format

A data frame (a vector for HYBRID1) with 32 observations on the following variables.

`hyb1` resistance measurements (Ω) of Res 3, a numeric vector

`hyb2` resistance measurements (Ω) of Res 3, a numeric vector

`hyb3` resistance measurements (Ω) of Res 3, a numeric vector

Source

See HADPAS

Examples

```
data(HYBRID)

lapply(HYBRID, var)
```

INSERTION *Components Insertions into a Board*

Description

Data represents a large number of insertions with $k = 9$ different components. The result of each trial (insertion) is either Success (no insertion error) or Failure (insertion error).

Usage

```
data(INSERTION)
```

Format

A data frame with 9 (k) observations on the following 3 variables.

`comp` Component, a factor with levels `C1` `C2` `C3` `C4` `C5` `C6` `C7` `C8` `C9`

`fail` Failure, a numeric vector

`succ` Success, a numeric vector

Details

Components are:

`C1`: Diode

`C2`: 1/2 Watt Canister

`C3`: Jump Wire

`C4`: Small Corning

`C5`: Large Corning

`C6`: Small Bullet

`C7`: 1/8 Watt Dogbone

`C8`: 1/4 Watt Dogbone

`C9`: 1/2 Watt Dogbone

Source

See `PLACE`

Examples

```
data(INSERTION)

barplot(INSERTION$fail /
          (INSERTION$fail + INSERTION$succ) *
          100,
        names.arg=INSERTION$comp,
        ylab= "Percentage")
```

IPL *Number of Computer Crashes per Month*

Description

Number of computer crashes per month, due to power failures experienced at a computer center, over a period of 28 months. After a crash, the computers are made operational with an "Initial Program Load".

Usage

```
data(IPL)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(IPL)

plot(IPL, type="b")
```

JANDEFECT *January Number of Defects in Daily Samples*

Description

Number of defective items found in random samples of size $n = 100$, drawn daily from a production line in January.

Usage

```
data(JANDEFECT)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(JANDEFECT)

plot(JANDEFECT, type="b")
```

KEYBOARDS *New Designs of Feyboards for Desktop Computers*

Description

The design of the keyboard might have effect on the speed of typing or on the number of typing `errors`. Noisy factors are `typist` or type of `job`. Letters A, B, C, D of variable `keyboard` denote the designs.

Usage

```
data(KEYBOARDS)
```

Format

A data frame with 25 observations on the following 4 variables.

`typist` The typist, a factor with levels `1` `2` `3` `4` `5`

`job` The type of job, a factor with levels `1` `2` `3` `4` `5`

`keyboard` Keyboard design, a factor with levels `A` `B` `C` `D` `E`

`errors` Number of typing errors, a numeric vector

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(KEYBOARDS)

boxplot(errors ~ keyboard, data=KEYBOARDS, ylab="Errors")
```

LATHYPPISTON *Latin Hypercube Design for the Piston Simulator*

Description

A Latin Hypercube Design for the 7 `pistonSimulation` arguments and Its response in seconds.

Usage

```
data(LATHYPPISTON)
```

Format

A data frame with 14 observations on the following 8 variables.

`m` a numeric vector

`s` a numeric vector

`v0` a numeric vector

`k` a numeric vector

`p0` a numeric vector

`t` a numeric vector

`t0` a numeric vector

`seconds` a numeric vector

Source

Kenett, R., Zacks, S. with contributions by Amberti, D. *Modern Industrial Statistics: with applications in R, MINITAB and JMP*. Wiley.

See Also

`pistonSimulation`

Examples

```
data(LATHYPPISTON)

library(DiceEval)

Dice <- km(design=LATHYPPISTON[, !names(LATHYPPISTON) %in% "seconds"],
           response=LATHYPPISTON[,"seconds"])

library(DiceView)

sectionview(Dice,
            center=colMeans(LATHYPPISTON[, !names(LATHYPPISTON) %in% "seconds"]),
            conf_lev=c(0.5, 0.9, 0.95),
```

```
            title="", col_sur="darkgrey", lwd=2,
            Xname=colnames(LATHYPPISTON[, !names(LATHYPPISTON) %in% "seconds"]))

layout(1)
```

`mahalanobisT2` *Mahalanobis $T\hat{\ }2$*

Description

Mahalanobis T^2 and Confidence Region

Usage

```
mahalanobisT2(x, factor.name, response.names = names(x)[!names(x) %in% factor.name],
conf.level=0.95, compare.to = NA, plot = FALSE)
```

Arguments

`x`	a data frame
`factor.name`	single character indicating column name of the experiment factor to test, the first level is used as a reference
`response.names`	vector of characters indicating columns names of the responses
`conf.level`	confidence level for the Confidence Region
`compare.to`	a vector of length `length(response.names)` to be compared to the result in terms of Mahalanobis T^2
`plot`	logical, if `TRUE` also a plot is produced

Value

a list with components:

`coord`	matrix with transformed coordinates of variables in `response.names`
`mahalanobis`	vector containing Lower Control Region, Center and Upper Control Region of Mahalanobis T^2
`mahalanobis.compare`	single value, Mahalanobis T^2 of `compare.to`

Author(s)

Daniele Amberti

References

Kenett, R., Zacks, S. with contributions by Amberti, D. *Modern Industrial Statistics: with applications in R, MINITAB and JMP*. Wiley.

Tsong et al, (1996).

Examples

```
data(DISS)

mahalanobisT2(DISS[, c("batch", "min15", "min90")],
              factor.name="batch",
              conf.level=0.90,
              compare.to=c(15, 15))
```

---

masPred1 *Moving Average Smoothing Predictor*

Description

A moving average smoother is a sequence which replaces X_t by a fitted polynomial based on the window of size n = 2m + s around X_t. The simplest smoother is the linear one.

Usage

```
masPred1(x, m, s)
```

Arguments

x	a vector of data X_t.
m	the m to define the window size.
s	the s to define the window size.

Value

a vector with values form the linear smoother predictor.

Author(s)

Shelemyahu Zacks

Examples

```
set.seed(123)
x <- 1:20 + rnorm(20, 0, 0.1)
masPred1(x, m = 3, s= 1)
masPred1(x, m = 3, s= 3)

data(DOW1941)
plot(DOW1941$Date, DOW1941$DOW1941,
     type="l",
     ylab="Dow Jones 1941", xlab="Date")
lines(DOW1941$Date, masPred1(DOW1941$DOW1941, m = 3, s= 1))
```

MPG *Gasoline Consumption of Cars by Origin*

Description

Gasoline consumption (in miles per gallon in city driving) of cars by origin. There are 3 variables representing samples of sizes $n1 = 58$, $n2 = 14$ and $n3 = 37$.

Usage

```
data(MPG)
```

Format

A data frame with 58 observations on the following 3 variables.

`origin1` Gasoline consumption, a numeric vector

`origin2` Gasoline consumption, a numeric vector

`origin3` Gasoline consumption, a numeric vector

Source

See CAR

Examples

```
data(MPG)

library(boot)

set.seed(123)

B <- apply(MPG, MARGIN=2,
           FUN=boot,
           statistic=function(x, i){
             var(x[i], na.rm = TRUE)
           },
           R = 500)

Bt0 <- sapply(B,
              FUN=function(x) x$t0)

Bt <-  sapply(B,
              FUN=function(x) x$t)

Bf <- max(Bt0)/min(Bt0)

FBoot <- apply(Bt, MARGIN=1,
               FUN=function(x){
```

```
          max(x)/min(x)
        })

Bf

quantile(FBoot, 0.95)

sum(FBoot >= Bf)/length(FBoot)

rm(Bt0, Bt, Bf, FBoot)
```

---

nrwm *Normal Random Walk*

Description

Normal Random Walk with a Bayesian prediction.

Usage

```
nrwm(n, v, w, c)
```

Arguments

n	length.
v	prior variance of V_0.
w	prior variance of W_0.
c	variance of the prior Normal Distribution.

Value

nrw	realizations of a Normal Random Walk
Mt	posterior expectation m_t

Author(s)

Shelemyahu Zacks

Examples

```
nrwm(n = 10, v = 5, w = 8, c = 0.5)
```

OELECT *Electric Voltage Outputs of Rectifying Circuits*

Description

99 electric voltage outputs of a rectifying circuit (V).

Usage

```
data(OELECT)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(OELECT)

summary(OELECT)

mean(OELECT)
```

OELECT1 *Electric Voltage Outputs of Another Rectifying Circuit*

Description

25 electric voltage outputs of a rectifying circuit (V).

Usage

```
data(OELECT1)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(OELECT)

data(OELECT1)

randomizationTest(list(a=OELECT, b=OELECT1),
                  R=500, calc=mean,
                  fun=function(x) x[1]-x[2])
```

OTURB *Cycle Times of a Piston*

Description

100 cycle times (s) of a piston, as from `pistonSimulation`.

Usage

```
data(OTURB)
```

References

See `pistonSimulation`

Examples

```
data(OTURB)

plot(OTURB, type="b")
```

OTURB1 *Cycle Times of a Piston from the Piston Simulator*

Description

50 cycle times (in s)of a piston gerated with `pistonSimulation(seed=123)`. Cycle times are rounded to 3 decimals.

Usage

```
data(OTURB1)
```

References

See `pistonSimulation`

Examples

```
data(OTURB1)

REF <- round(pistonSimulation(seed=123)$seconds, 3)

plot(OTURB1, type="b", lwd=6)

lines(REF, col=2, lwd=2)

sum(OTURB1 - REF)
```

OTURB2 *Sample Mean and Standard Deviation of Cycle Times of a Piston*

Description

In this data frame we have three variables. In the first we have the sample size. In the second and third we have the sample means and standard deviation.

Usage

```
data(OTURB2)
```

Format

A data frame with 10 observations on the following 3 variables.

`groupSize` a numeric vector

`xbar` a numeric vector

`std` a numeric vector

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(OTURB2)
plot(OTURB2$xbar, type="b")
plot(OTURB2$std, type="b")
```

PBX *Software Errors Found in Testing a PBX*

Description

Software errors found in testing a Private Branch Exchange electronic switch. Errors are labeled according to the software unit where they occurred (e.g. "EKT", Electronic Key Telephone).

Usage

```
data(PBX)
```

Format

The format is: Named num [1:5] 473 252 110 100 65 - attr(*, "names")= chr [1:5] "GEN" "VHS" "HI" "LO" ...

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(PBX)

barplot(PBX)
```

`pistonSimulation` *The Piston Simulator*

Description

A simulator of a piston moving whithin a cylinder. The piston's performance is measured by the time it takes to complete one cycle, in seconds. Several factors can affect the piston's performance, they are listed in the arguments section.

Usage

```
pistonSimulation(m = 60, s = 0.02, v0 = 0.01,
                 k = 5000, p0 = 110000, t = 296,
                 t0 = 360, each = 50, seed = NA,
                 check = TRUE)
```

Arguments

`m`	the impact pressure determined by the piston weight (kg). A single value or a vector of length *n*.
`s`	the piston surface area (m^2). A single value or a vector.
`v0`	the initial volume of the gas inside the piston (m^3). A single value or a vector of length *n*.
`k`	the spring coefficient (N/m^3). A single value or a vector of length *n*.
`p0`	the atmospheric pressure (N/m^2). A single value or a vector of length *n*.
`t`	the surrounding ambient temperature (K). A single value or a vector of length *n*.
`t0`	the filling gas temperature (K). A single value or a vector of length *n*.
`each`	non-negative integer. Each element of previous parameters is repeated each times.

seed	a single value, interpreted as an integer. If specified make the simulation replicable.
check	if TRUE (the default) then a formal check on piston parameters is perfomed

Details

Factors affect the Cycle Time s via a chain of nonlinear equations:

$$s = 2\pi\sqrt{\frac{M}{k + S^2 \frac{P_0 V_0}{T_0} \frac{T}{V^2}}}$$

where

$$V = \frac{S}{2k}\sqrt{A^2 + 4k\frac{P_0 V_0}{T_0}T - a}$$

and

$$A = P_0 S + 19.62M - \frac{kV_0}{S}$$

Value

A data frame, a matrix-like structure, with each * n rows and with columns:

m	numeric	value of m
s	numeric	value of s
v0	numeric	value of v0
k	numeric	value of k
p0	numeric	value of p0
t	numeric	value of t
t0	numeric	value of t0
seconds	numeric	time to complete one cycle (s)

Author(s)

Daniele Amberti

References

Kenett, R., Zacks, S. with contributions by Amberti, D. *Modern Industrial Statistics: with applications in R, MINITAB and JMP*. Wiley.

See Also

powerCircuitSimulation, simulationGroup, LATHYPPISTON

Examples

```
Ps <- pistonSimulation(
  m = rep(60, 100),
  s = rep(0.02, 100),
  v0 = rep(0.01, 100),
```

```
  k = rep(5000, 100),
  p0 = rep(110000, 100),
  t = c(rep(296,35), 296*1.1^(1:65)),
  t0 = rep(360, 100),
  each = 1,
  seed = 123,
  check = FALSE)
head(Ps)
tail(Ps)
plot(Ps$seconds)
```

PLACE *Displacements of Electronic Components on Printed Circuit Boards*

Description

The observations are the displacements (position errors) of electronic components on printed circuit boards. There are 26 boards. 16 components are placed on each board. Each component has to be placed at a specific location (x, y) on a board and with correct orientation θ.

Usage

```
data(PLACE)
```

Format

A data frame with 416 observations on the following 4 variables.

`crcBrd` Circuit board number, a numeric vector

`xDev` Error in placement along the x-axis, a numeric vector

`yDev` Error in placement along the y-axis, a numeric vector

`tDev` Error in orientation θ, a numeric vector

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(PLACE)

plot(PLACE[,-1])

boxplot(xDev ~ crcBrd, data=PLACE,
        ylab="xDev", xlab="crcBrd")

PLACE$code <- factor(c(rep("lDev", 9*16),
```

```
                    rep("mDev", 3*16),
                    rep("hDev", 14*16)))

plot(PLACE[,"xDev"], PLACE[,"yDev"],
     pch=as.integer(PLACE[,"code"]),
     main="", xlab="xDev", ylab="yDev")

grid()
```

---

powerCircuitSimulation   *The Power Circuit Simulator*

Description

A simulator of a voltage conversion power circuit. The target output voltage of the power circuit is 220 volts DC. The circuit consists of 10 resistances labeled *A* to *J*, and 3 transistors, labeled *K* to *M*. These components can be purchased with different tolerance grades.

Usage

```
powerCircuitSimulation(rsA = 8200, rsB = 220000, rsC = 1000,
                       rsD = 33000, rsE = 56000, rsF = 5600,
                       rsG = 3300, rsH = 58.5, rsI = 1000,
                       rsJ = 120, trK = 130, trL = 100,
                       trM = 130,
                       tlA = 5, tlB = 10, tlC = 10,
                       tlD = 5, tlE = 5, tlF = 5,
                       tlG = 10, tlH = 5, tlI = 5,
                       tlJ = 5, tlK = 5, tlL = 10,
                       tlM = 5,
                       each = 50, seed = NA)
```

Arguments

rsA	the resistance (Ω) of *A*. A single value or a vector of length *n*.
rsB	the resistance (Ω) of *B*. A single value or a vector of length *n*.
rsC	the resistance (Ω) of *C*. A single value or a vector of length *n*.
rsD	the resistance (Ω) of *D*. A single value or a vector of length *n*.
rsE	the resistance (Ω) of *E*. A single value or a vector of length *n*.
rsF	the resistance (Ω) of *F*. A single value or a vector of length *n*.
rsG	the resistance (Ω) of *G*. A single value or a vector of length *n*.
rsH	the resistance (Ω) of *H*. A single value or a vector of length *n*.
rsI	the resistance (Ω) of *I*. A single value or a vector of length *n*.

`rsJ`	the resistance (Ω) of *J*. A single value or a vector of length *n*.
`trK`	the resistance (Ω) of *K*. A single value or a vector of length *n*.
`trL`	the resistance (Ω) of *L*. A single value or a vector of length *n*.
`trM`	the resistance (Ω) of *M*. A single value or a vector of length *n*.
`tlA`	the tolerance of *A*. It is a number > 0 (e.g. 5% is 5.0)
`tlB`	the tolerance of *B*. It is a number > 0 (e.g. 5% is 5.0)
`tlC`	the tolerance of *C*. It is a number > 0 (e.g. 5% is 5.0)
`tlD`	the tolerance of *D*. It is a number > 0 (e.g. 5% is 5.0)
`tlE`	the tolerance of *E*. It is a number > 0 (e.g. 5% is 5.0)
`tlF`	the tolerance of *F*. It is a number > 0 (e.g. 5% is 5.0)
`tlG`	the tolerance of *G*. It is a number > 0 (e.g. 5% is 5.0)
`tlH`	the tolerance of *H*. It is a number > 0 (e.g. 5% is 5.0)
`tlI`	the tolerance of *I*. It is a number > 0 (e.g. 5% is 5.0)
`tlJ`	the tolerance of *J*. It is a number > 0 (e.g. 5% is 5.0)
`tlK`	the tolerance of *K*. It is a number > 0 (e.g. 5% is 5.0)
`tlL`	the tolerance of *L*. It is a number > 0 (e.g. 5% is 5.0)
`tlM`	the tolerance of *M*. It is a number > 0 (e.g. 5% is 5.0)
`each`	non-negative integer. Each element of previous parameters is repeated `each` times.
`seed`	a single value, interpreted as an integer. If specified make the simulation replicable.

Details

Factors affect the voltage output V via a chain of nonlinear equations:

$$V = \frac{136.67(a + \frac{b}{Z(10)}) + d(c+e)\frac{g}{f} - h}{1 + d\frac{e}{f} + b[frac1Z(10) + 0.006(1 + \frac{13.67}{Z(10)})] + 0.08202a}$$

where

$$a = \frac{Z(2)}{Z(1)+Z(2)}$$

$$b = \frac{1}{Z(12)+Z(13)}(Z(3) + \frac{Z(1)Z(2)}{Z(1)+Z(2)}) + Z(9)$$

$$c = Z(5) + Z(7)/2$$

$$d = Z(11)\frac{Z(1)Z(2)}{Z(1)+Z(2)}$$

$$e = Z(6) + Z(7)/2$$

$$f = (c+e)(1+Z(11))Z(8) + ce$$

$$g = 0.6 + Z(8)$$

$$h = 1.2$$

with $Z(1), \ldots, Z(10)$ resistances in Ω of the 10 resistances and $Z(11), Z(12), Z(13)$ are the h_{FE} values of three transistors.

Value

A data frame, a matrix-like structure, with each * n rows and with columns:

rsA	numeric	value of `rsA`
rsB	numeric	value of `rsB`
rsC	numeric	value of `rsC`
rsD	numeric	value of `rsD`
rsE	numeric	value of `rsE`
rsF	numeric	value of `rsF`
rsG	numeric	value of `rsG`
rsH	numeric	value of `rsH`
rsI	numeric	value of `rsI`
rsJ	numeric	value of `rsJ`
trK	numeric	value of `trK`
trL	numeric	value of `trL`
trM	numeric	value of `trM`
tlA	numeric	value of `tlA`
tlB	numeric	value of `tlB`
tlC	numeric	value of `tlC`
tlD	numeric	value of `tlD`
tlE	numeric	value of `tlE`
tlF	numeric	value of `tlF`
tlG	numeric	value of `tlG`
tlH	numeric	value of `tlH`
tlI	numeric	value of `tlI`
tlJ	numeric	value of `tlJ`
tlK	numeric	value of `tlK`
tlL	numeric	value of `tlL`
tlM	numeric	value of `tlM`
volts	numeric	output in volts (V)

Author(s)

Daniele Amberti

References

Kenett, R., Zacks, S. with contributions by Amberti, D. *Modern Industrial Statistics: with applications in R, MINITAB and JMP*. Wiley.

See Also

`pistonSimulation`, `simulationGroup`

Examples

```
powerCircuitSimulation(seed=123, each=3)
```

PRED *Soldering Points*

Description

1,000 records on variable `x` and `y`. `x` is the number of soldering points on a board, and `y` is the number of defective soldering points.

Usage

```
data(PRED)
```

Format

A data frame with 1000 observations on the following 2 variables.

- `x` Number of soldering points, a numeric vector
- `y` Number of defective soldering points, a numeric vector

Details

Electronic systems such as television sets, radios or computers contain printed circuit boards with electronic components positioned in patterns determined by design engineers. After assembly (either by automatic insertion machines or manually) the components are soldered to the board. In the relatively new Surface Mount Technology minute components are simultaneously positioned and soldered to the boards. The occurrence of defective soldering points impacts the assembly plant productivity and is therefore closely monitored

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(PRED)

library(boot)

set.seed(123)

YRatioPred <- boot(data=PRED$x,
                   statistic=function(x,i){
                     mean(x[i[1:100]])*7.495/148.58
                     },
                   R=1000)$t

hist(YRatioPred, main="",
     xlab="",
     xlim=c(7, 8))
```

pred1 — *Linear Predictor for Covariance Stationary Time Series*

Description

An Optimal Linear Predictor for Covariance Stationary Time Series.

Usage

```
pred1(x, n = 10)
```

Arguments

`x`	a vector of data X_t.
`n`	the n to define the window size.

Value

a vector with values form the linear predictor.

Author(s)

Shelemyahu Zacks

Examples

```
pred1(rnorm(20), n = 10)
```

predARMA — *Prediction of an AR 3 process*

Description

For isslustration purposes this implements the prediction of an ARMA(3, 0) process.

Usage

```
predARMA(X, a)
```

Arguments

`X`	a vector of data X_t.
`a`	vector p of parameters.

Value

a vector with values from the quadratic predictor.

Author(s)

Shelemyahu Zacks

See Also

`ARMA`

Examples

```
set.seed(123)
predARMA(ARMA(100,c(0.1, 0.2, 0.3), c(0.1, 0.2)), c(0.1, 0.2, 0.3))
```

`predPoly` *Quadratic Predictor*

Description

A quadratic polynomial fitted to the last n observations. We then extrapolate to estimate f(t + s) with s >= 1.

Usage

```
predPoly(x, n, s)
```

Arguments

`x`	a vector of data X_t.
`n`	the n to define the window size.
`s`	the extrapolation parametr s.

Value

a vector with values from the quadratic predictor.

Author(s)

Shelemyahu Zacks

Examples

```
data(DOW1941)
plot(DOW1941$Date, DOW1941$DOW1941,
     type="l",
     ylab="Dow Jones 1941", xlab="Date")

lines(DOW1941$Date, predPoly(DOW1941$DOW1941, n = 20, s= 1))
```

---

randomizationTest *Randomization Test*

Description

A function to perform randomization test

Usage

```
randomizationTest(list, R = 500, calc, fun = NA,
                  seed = NA, printSummary = TRUE)
```

Arguments

list	a list with two or more numeric vectors
R	
calc	a function to be applied to every vector in `list`
fun	a function to be applied to a vector (e.g. `x`) of length `length(list)`, containing result of function `calc`
seed	a single value, interpreted as an integer. If specified make the simulation replicable.
printSummary	logical, if `TRUE` print a summary of the randomization test

Value

The silently returned value is an object of class "boot"

Author(s)

Daniele Amberti

References

Kenett, R., Zacks, S. with contributions by Amberti, D. *Modern Industrial Statistics: with applications in R, MINITAB and JMP*. Wiley.

See Also

boot

Examples

```
data(OELECT)
data(OELECT1)

# test difference in mean:
randomizationTest(list(a=OELECT, b=OELECT1),
                  R=500, calc=mean,
                  fun=function(x) x[1]-x[2],
                  seed=123)
```

---

renewDis *Renewals Disribution*

Description

Provide the Empirical Bootstrap Distribution of the number of renewals in a specified time interval.

Usage

```
renewDis(ttf, ttr, time, n, printSummary = TRUE)
```

Arguments

ttf	numeric vector of Time To Failure
ttr	numeric vector of Time To Repair
time	numeric value representing the time horizon on which number of renewals are calculated
n	the number of bootstrap replicates
printSummary	logical, if TRUE print the Mean Number of Renewals, and a summary of renewals values

Value

A numeric vector of lenght n with simulated number of renewals

Author(s)

Daniele Amberti

References

Kenett, R., Zacks, S. with contributions by Amberti, D. *Modern Industrial Statistics: with applications in R, MINITAB and JMP*. Wiley.

See Also

availDis

Examples

```
set.seed(123)

Ttf <- rgamma(50,
              shape=2,
              scale=100)

Ttr <- rgamma(50,
              shape=2,
              scale=1)

RenewEbd <- renewDis(ttf=Ttf,
                     ttr=Ttr,
                     time=1000,
                     n=1000)
```

RNORM10 *Random Sample from N(10, 1)*

Description

Random sample of size $n = 28$ from the normal distribution $N(10, 1)$.

Usage

```
data(RNORM10)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(RNORM10)

plot(RNORM10, type="b")

abline(h=10, lwd=2, col=2)
```

SeasCom *Monthly demand for a commodity*

Description

Monthly demand for a seasonal commodity during 102 months.

Usage

```
data("SeasCom")
```

Format

The format is: num [1:102] 72 56.4 64.9 59.9 51.6 ...

Examples

```
data(SeasCom)

plot(SeasCom,
     type="b",
     ylab="Dow Jones")
```

SENSORS *SENSORS*

Description

174 measurements from 63 sensors tracking performance of a system under test. Each test generates values for these 63 sensors and a status determined by the automatic test equipment. The test results are coded as Pass (corresponding to 'Good' and Fail (all other categories).

Usage

```
data("SENSORS")
```

Format

A data frame with 174 observations on the following 65 variables.

`sensor01` a numeric vector

`sensor02` a numeric vector

`sensor03` a numeric vector

`sensor04` a numeric vector

`sensor05` a numeric vector

sensor06 a numeric vector
sensor07 a numeric vector
sensor08 a numeric vector
sensor09 a numeric vector
sensor10 a numeric vector
sensor11 a numeric vector
sensor12 a numeric vector
sensor13 a numeric vector
sensor14 a numeric vector
sensor15 a numeric vector
sensor16 a numeric vector
sensor17 a numeric vector
sensor18 a numeric vector
sensor19 a numeric vector
sensor20 a numeric vector
sensor21 a numeric vector
sensor22 a numeric vector
sensor23 a numeric vector
sensor24 a numeric vector
sensor25 a numeric vector
sensor26 a numeric vector
sensor27 a numeric vector
sensor28 a numeric vector
sensor29 a numeric vector
sensor30 a numeric vector
sensor31 a numeric vector
sensor32 a numeric vector
sensor33 a numeric vector
sensor34 a numeric vector
sensor35 a numeric vector
sensor36 a numeric vector
sensor37 a numeric vector
sensor38 a numeric vector
sensor39 a numeric vector
sensor40 a numeric vector
sensor41 a numeric vector
sensor42 a numeric vector

`sensor43` a numeric vector

`sensor44` a numeric vector

`sensor45` a numeric vector

`sensor46` a numeric vector

`sensor47` a numeric vector

`sensor48` a numeric vector

`sensor49` a numeric vector

`sensor50` a numeric vector

`sensor51` a numeric vector

`sensor52` a numeric vector

`sensor53` a numeric vector

`sensor54` a numeric vector

`sensor55` a numeric vector

`sensor56` a numeric vector

`sensor57` a numeric vector

`sensor58` a numeric vector

`sensor59` a numeric vector

`sensor60` a numeric vector

`sensor61` a numeric vector

`sensor62` a numeric vector

`sensor63` a numeric vector

`testResult` a factor with levels `Brake` `Good` `Grippers` `IMP` `ITM` `Motor` `SOS` `Velocity Type I` `Velocity Type II`

`status` a factor with levels `Fail` `Pass`

Examples

```
data(SENSORS)
library(rpart)
rpart(status ~ . , data=SENSORS[, c("status", "sensor18", "sensor55")])
```

`shroArlPfaCed` *ARL, PFA and CED of Shiryayev-Roberts procedure*

Description

Average Run Length, the Probability of False Alarm and the Conditional Expected Delay, given that the alarm is given after the change-point for Normal and Poisson cases

Usage

```
shroArlPfaCedNorm(mean0 = 0, mean1 = NA, sd = 1, n = 10,
                  delta = 1, tau = NA, N = 100, limit = 10000,
                  seed = NA, w = 19, printSummary = TRUE)

shroArlPfaCedPois(lambda0 = 10, lambda1 = NA, delta = 1,
                  tau = NA, N = 100, limit = 10000, seed = NA,
                  w = 19, printSummary = TRUE)
```

Arguments

`mean0`	value of the mean of a normal distributed process
`mean1`	optional value of the mean after a shift in a normal process, ignored if `delta` is not `NA`
`sd`	standard deviation of a normal distributed process
`n`	sample size
`lambda0`	mean of a Poisson distributed process
`lambda1`	optional value of the mean after a shift in a Poisson process, ignored if `delta` is not `NA`
`delta`	value of the shift from `mean0` or `lambda0`, set to `NA` if the alternative specification with `mean1` or `lambda1` is needed
`tau`	location of the point of change in the process parameter `mean0` or `lambda0`, if `NA` simulation is perfomed without any shift: `mean1`, `lambda1` and `delta` are ignored
`N`	the number of replicates
`limit`	safety parameter, stop rule for procedures with very long ARL
`seed`	a single value, interpreted as an integer. If specified make the simulation replicable.
`w`	Shiryayev-Roberts statistics used as the stopping threshold
`printSummary`	logical, if `TRUE` print a summary of the Shiryayev-Roberts ARL, PFA and CED

Value

a list with elements:

`rls`	a numeric vector representing the Run Length of the simulation
`statistics`	a numeric vector with summary statistics
`run`	a list of length `N` elements each of which has single numeric elements `violationLower`, `violationUpper` and `rl`

Author(s)

Daniele Amberti

References

Kenett, R., Zacks, S. with contributions by Amberti, D. *Modern Industrial Statistics: with applications in R, MINITAB and JMP*. Wiley.

Examples

```
shroArlPfaCedNorm(mean0=10,
                  sd=3,
                  n=5,
                  delta=0.5,
                  tau=10,
                  w=99,
                  seed=123)

shroArlPfaCedPois(lambda0=5,
                  delta=0.5,
                  tau=10,
                  w=99,
                  seed=123)

shroArlPfaCedNorm(mean0=15,
                  sd=3,
                  n=5,
                  delta=0.5,
                  tau=NA,
                  w=99,
                  seed=123)
```

---

simOAB — *Bayesian One-Armed Bernoulli Bandits process*

Description

Simulate the expected number of trials on arm B before switching to the known arm A, and the expected reward.

Usage

```
simOAB(N, p, al, k, gam, Ns)
```

Arguments

N	number of trials.
p	the probability of reward on arm B (unknown).
al	the known probability of reward on arm A.
k	the initial sample size on arm B.
gam	Bayesian confidence level.
Ns	number of runs in the simulation.

Value

`MeanValueStoppingTime`
: mean value at the stopping time

`StandardDeviationST`
: standard deviation of the value at the stopping time

`MeanValueExpectedReward`
: mean value of the expected reward

`StandardDeviationST`
: standard deviation of the expected reward

Author(s)

Shelemyahu Zacks

See Also

dynOAB

Examples

```
set.seed(123)
simOAB(N = 50, p = 0.6, al = 0.5, k = 10, gam = 0.95, Ns = 1000)
```

---

simulationGroup *Simulation Group*

Description

Add a column named `group` to an object of class "mistatSimulation".

Usage

```
simulationGroup(x, n)
```

Arguments

`x`	an object of class "mistatSimulation"
`n`	size of the group or sample

Value

Add a column named `group` to an object of class "mistatSimulation".

Author(s)

Daniele Amberti

See Also

pistonSimulation, powerCircuitSimulation

Examples

```
simulationGroup(pistonSimulation(each=20), 5)
simulationGroup(powerCircuitSimulation(each=20), 5)
```

SOCELL — *Short Circuit Current of Solar Cells*

Description

Short circuit current (ISC) of 16 solar cells measured at three time epochs, one month apart.

Usage

```
data(SOCELL)
```

Format

A data frame with 16 observations on the following 3 variables.

`t1` ISC at time epoch 1, a numeric vector

`t2` ISC at time epoch 2, a numeric vector

`t3` ISC at time epoch 3, a numeric vector

Details

Telecommunication sattelites are powered while in orbit by solar cells. Tadicell, a solar cells producer that supplies several satellite manufacturers, was requested to provide data on the degradation of its solar cells over time. Tadicell engineers performed a simulated experiment in which solar cells were subjected to temperature and illumination changes similar to those in orbit and measured the short circuit current ISC (ampers), of solar cells at three different time periods, in order to determine their rate of degradation.

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(SOCELL)

LmISC <- lm(t2 ~ 1 + t1,
            data=SOCELL)

summary(LmISC)
```

SOLDEF *Solder Defects*

Description

Solder defects on 380 printed circuits boards of varying size.

Usage

```
data(SOLDEF)
```

Details

In SOLDEF we present results of testing batches of circuit boards for defects in solder points, after wave solderings. The batches includes boards of similar design. There were close to 1,000 solder points on each board. The results Xtare number of defects per 10^6 points *(PPM)*. The quality standard is $\lambda_0 = 100(PPM)$. λ_t values below λ_0 represent high quality soldering. In this data file there are N = 380 test results. Only 78 batches had an X_t value greater than $\lambda_0 = 100$.

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(SOLDEF)

hist(SOLDEF)
```

STEELROD *50 Measurements of the Length of Steel Rods in cm*

Description

Steel rods are used in the car and truck industry to strengthen vehicle structures. Steel rods supplied by Urdon Industries are produced by a process adjusted to obtain rods of length 20 cm. However, due to natural fluctuations in the production process, the actual length of the rods varies around the nominal value of 20 cm.

Usage

```
data(STEELROD)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(STEELROD)

plot(STEELROD,
     ylab = "Steel rod Length",
     xlab = "Index")
```

STRESS *Stress Levels*

Description

Results of a 33 factorial experiment to investigate the effects of three factors A, B, C on the stress levels of a membrane Y. The first three columns of the data provide the levels of the three factors, and column 4 presents the stress values.

Usage

```
data(STRESS)
```

Format

A data frame with 27 observations on the following 4 variables.

`A` levels of factor A, a numeric vector

`B` levels of factor B, a numeric vector

`C` levels of factor C, a numeric vector

`stress` stress levels of a membrane Y, a numeric vector

Source

Oikawa and Oka (1987)

Examples

```
data(STRESS)

summary(
  aov(stress ~ (A+B+C)^3 +I(A^2)+I(B^2)+I(C^2),
      data=STRESS))
```

SYSTEMFAILURE *SYSTEMFAILURE*

Description

208 observations on systems operating at 90 geographically dispersed sites.

Usage

```
data("SYSTEMFAILURE")
```

Format

A data frame with 208 observations on the following 5 variables.

`CustomerNumber` a factor, customer ID

`Censor` a numeric vector

`TimeStamp` a numeric vector

`SystemMaturity` a factor with levels `Mature` `Young`

`Country` a factor with levels `AUSTRALIA` `AUSTRIA` `BELGIUM` `BRAZIL` `BULGARIA` `CANADA` `CHINA` `DENMARK` `FINLAND` `GERMANY` `HUNGARY` `INDIA` `ITALY` `JAPAN` `NETHERLANDS` `NEW ZEALAND` `POLAND` `PORTUGAL` `REPUBLIC OF KOREA` `ROMANIA` `RUSSIAN FEDERATION` `SOUTH AFRICA` `SPAIN` `SWITZERLAND` `TURKEY` `UNITED KINGDOM` `UNITED STATES`

Details

Twelve systems are new installed and are labeled as 'Young'. All the other systems are labeled 'Mature'. Out of the 208 observations, 68 report time stamps of a failure (uncensored). The other observations are censored, as indicated by the value 1 in the Censor variable column. A measure of time, the time stamp, is recorded for each observation in the data. This variable presented in operational units (activity time), at time of observation. The bigger the time, the longer the system performed. The observations with a value 0 of the Censor variable, represent length of operation till failure of the systems.

Examples

```
data(SYSTEMFAILURE)
summary(subset(SYSTEMFAILURE, subset = Censor == 0, select = "TimeStamp"))
```

THICKDIFF *Difference in Thickness*

Description

Difference between the thickness of the grown silicon layer and its target value.

Usage

```
data(THICKDIFF)
```

Source

E. Yashchin (1991)

Examples

```
data(THICKDIFF)

plot(THICKDIFF, type="b")
```

toeplitz *Toeplitz matrix*

Description

partial lag correlation Toeplitz matrix.

Usage

```
toeplitz(a)
```

Arguments

a	An array containing the estimated acf from function `acf`.

Value

a matrix.

Author(s)

Shelemyahu Zacks

Examples

```
toeplitz(acf(nottem, 5)$acf)
```

TSQ — *T^2 values of PLACE data*

Description

368 T^2 values corresponding to the vectors (x, y, θ) of displacements (position errors) of electronic components on printed circuit boards.

Usage

```
data(TSQ)
```

Source

See PLACE

Examples

```
data(TSQ)

plot(TSQ, type="b")
```

VENDOR — *Number of cycles required until latch failure*

Description

Number of cycles reqiored until latch failure in 30 floppy disk drives from three different disk vendors.

Usage

```
data(VENDOR)
```

Format

A data frame with 10 observations on the following 3 variables.

- `vendor1` number of cycles required until latch failure for vendor A_1, a numeric vector
- `vendor2` number of cycles required until latch failure for vendor A_2, a numeric vector
- `vendor3` number of cycles required until latch failure for vendor A_3, a numeric vector

Details

Three different vendors are considered for supplying cases for floppy disk drives. The question is whether the latch mechanism that opens and closes the disk loading slot is sufficiently reliable. In order to test the reliability of this latch, three independent samples of cases, each of size $n = 10$, were randomly selected from the production lots of these vendors. The testing was performed on a special apparatus that opens and closes a latch, until it breaks. The number of cycles required until latch failure was recorded. In order to avoid uncontrollable environmental factors to bias the results, the order of testing of cases of different vendors was completely randomized. In data VENDOR there are the results of this experiment, arranged in 3 columns. Column 1 represents the sample from vendor A_1; column 2 that of vendor A_2 and column 3 of vendor A_3.

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(VENDOR)

VENDOR <- stack(VENDOR)

VENDOR$ind <- as.factor(VENDOR$ind)

VENDOR$values <- sqrt(VENDOR$values)

confint(lm(values ~ -1 + ind,
           data=VENDOR))
```

WEIBUL *Random sample from a Weibull distribution*

Description

Values of a random sample of size $n = 50$ from a Weibull distribution.

Usage

```
data(WEIBUL)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(WEIBUL)

hist(WEIBUL)
```

YARNSTRG *Yarn strength*

Description

Yarn strength is typically analyzed on a logarithmic scale. This logarithmic transformation produces data that is more symmetrically distributed. in YARNSTRG data there are $n = 100$ values of $Y = \ln(X)$ where X is the yarn-strength in $lb./22yarns$ of woolen fibers.

Usage

```
data(YARNSTRG)
```

Source

Kenett, R. and Zacks, S. (1998) *Modern Industrial Statistics: The Design and Control of Quality and Reliability*. Duxbury Press.

Examples

```
data(YARNSTRG)

hist(YARNSTRG,
     breaks=6,
     main="",
     xlab = "Log yarn strength")
```

Index