The DAAG Package

August 3, 2006

Version 0.84

Date 2006-June-17

Title Data Analysis And Graphics

Author John Maindonald and W. John Braun

Maintainer W. John Braun <braun@stats.uwo.ca>


LazyLoad true

LazyData true

Depends R (>= 2.0.1)

Suggests leaps, oz

ZipData no

License Unlimited use and distribution.

URL http://www.stats.uwo.ca/DAAG

R topics documented:

ACF1 ......................................................... 4
CVbinary .................................................... 5
CVlm ......................................................... 6
Cars93.summary .......................................... 7
Lottario ..................................................... 8
Manitoba.lakes .......................................... 8
ais .......................................................... 9
allbacks .................................................... 10
anesthetic ............................................... 11
ant111b ................................................... 12
antigua .................................................... 13
appletaste ............................................... 13
austpop ................................................... 14
bestset.noise ........................................... 15
R topics documented:

- biomass .................................................. 16
- bomsoi ..................................................... 17
- bomsoi2001 ............................................... 19
- boston ...................................................... 22
- bounce ..................................................... 23
- capstring .................................................. 24
- carprice .................................................... 24
- cfseal ..................................................... 24
- cities ....................................................... 26
- coding ..................................................... 28
- compareTreeccaIs ........................................ 29
- component.residual ...................................... 30
- cottonworkers ............................................ 31
- cuckoohosts ............................................... 32
- cuckoos ................................................... 33
- cv.binary .................................................. 34
- cv.lm ........................................................ 35
- datafile ................................................... 36
- dengue ..................................................... 36
- dewpoint ................................................... 37
- elastic1 .................................................... 38
- elastic2 .................................................... 39
- elasticband ............................................... 40
- fossilfuel .................................................. 41
- frogs ........................................................ 42
- frostedflakes .............................................. 43
- fruitohms ................................................... 44
- geophones .................................................. 45
- hardcopy .................................................... 45
- head.injury ............................................... 46
- headInjury ................................................ 47
- hills ......................................................... 48
- hills2000 ................................................... 49
- houseprices ............................................... 50
- humanpower ............................................... 51
- ironslag .................................................... 52
- jobs .......................................................... 53
- kiwishade ................................................... 54
- leafshape .................................................... 56
- leafshape17 ............................................... 57
- leaftemp ..................................................... 58
- leaftemp.all ............................................... 59
- litters ....................................................... 59
- logisticsim ............................................... 60
- measles ..................................................... 61
- mifem ....................................................... 62
- mignonette ................................................ 63
- milk .......................................................... 63
- modelcars ................................................ 64
- monica ...................................................... 65
- moths ....................................................... 66
- multilap .................................................... 66
R topics documented:
nsw74demo ................................................................. 67
nsw74psid1 ............................................................... 68
nsw74psid3 ............................................................... 69
nsw74psidA ............................................................... 70
obounce ................................................................. 71
oddbooks ............................................................... 71
onesamp ................................................................. 72
onet.permutation ....................................................... 73
onetPermutation ....................................................... 73
oneway.plot ............................................................ 74
onewayPlot ............................................................ 75
orings ................................................................... 76
overlap.density ........................................................ 76
overlapDensity ......................................................... 77
ozone ................................................................... 78
pair65 ................................................................. 79
panel.corr ............................................................. 80
panelCorr ............................................................. 80
panelplot ............................................................. 81
pause ................................................................. 82
poissonsim ............................................................ 82
possum ................................................................. 83
possumsites ........................................................... 85
powerplot ............................................................. 85
poxetc ................................................................. 86
press ................................................................. 87
primates .............................................................. 88
qreference ............................................................ 88
races2000 ............................................................. 89
rainforest ............................................................. 90
rareplants ........................................................... 91
rice ................................................................. 92
roller ................................................................. 93
science ............................................................... 93
seedrates ............................................................. 95
show.colors .......................................................... 96
simulateLinear ........................................................ 96
socssupport .......................................................... 97
softbacks ............................................................. 98
sorption .............................................................. 99
spam7 ................................................................. 100
stVincent ............................................................ 100
sugar ................................................................. 101
tinting ............................................................... 102
toycars ............................................................... 103
two65 ................................................................. 103
twot.permutation ..................................................... 104
twotPermutation ..................................................... 105
vif ................................................................. 106
vince111b ............................................................ 107
vlt ................................................................. 107
wages1833 ............................................................ 108
Aberrant Crypt Foci in Rat Colons

Description
Numbers of aberrant crypt foci (ACF) in the section 1 of the colons of 22 rats subjected to a single dose of the carcinogen azoxymethane (AOM), sacrificed at 3 different times.

Usage
ACF1

Format
This data frame contains the following columns:

- **count** The number of ACF observed in section 1 of each rat colon
- **endtime** Time of sacrifice, in weeks following injection of AOM

Source
Ranjana P. Bird, Faculty of Human Ecology, University of Manitoba, Winnipeg, Canada.

References

Examples
```r
sapply(split(ACF1$count, ACF1$endtime), var)
plot(count ~ endtime, data=ACF1, pch=16)
pause()
print("Poisson Regression - Example 8.3")
ACF.glm0 <- glm(formula = count ~ endtime, family = poisson, data = ACF1)
summary(ACF.glm0)

# Is there a quadratic effect?
pause()

ACF.glm <- glm(formula = count ~ endtime + I(endtime^2),
family = poisson, data = ACF1)
summary(ACF.glm)

# But is the data really Poisson? If not, try quasipoisson:
pause()

ACF.glm <- glm(formula = count ~ endtime + I(endtime^2),
family = quasipoisson, data = ACF1)
summary(ACF.glm)
```
**CVbinary**  
*Cross-Validation for Regression with a Binary Response*

**Description**

This function gives internal and cross-validation measures of predictive accuracy for regression with a binary response. The data are randomly assigned to a number of ‘folds’. Each fold is removed, in turn, while the remaining data is used to re-fit the regression model and to predict at the deleted observations.

**Usage**

```r
CVbinary(obj=frogs.glm, rand=NULL, nfolds=10, print.details=TRUE)
```

**Arguments**

- `obj`  
  a `glm` object

- `rand`  
  a vector which assigns each observation to a fold

- `nfolds`  
  the number of folds

- `print.details`  
  logical variable (TRUE = print detailed output, the default)

**Value**

- the order in which folds were deleted

- internal estimate of accuracy

- cross-validation estimate of accuracy

**Author(s)**

J.H. Maindonald

**See Also**

`glm`

**Examples**

```r
frogs.glm <- glm(pres.abs ~ log(distance) + log(NoOfPools),  
                 family=binomial, data=frogs)
CVbinary(frogs.glm)
```
Cross-Validation for Linear Regression

Description

This function gives internal and cross-validation measures of predictive accuracy for ordinary linear regression. The data are randomly assigned to a number of ‘folds’. Each fold is removed, in turn, while the remaining data is used to re-fit the regression model and to predict at the deleted observations.

Usage

CVlm(df = houseprices, form.lm = formula(sale.price ~ area), m=3, dots = FALSE, seed=29, plotit=TRUE, printit=TRUE)

Arguments

df a data frame
form.lm a formula object
m the number of folds
dots uses pch=16 for the plotting character
seed random number generator seed
plotit if TRUE, a plot is constructed on the active device
printit if TRUE, output is printed to the screen

Value

For each fold, a table listing

the residuals
ms = the overall mean square of prediction error

Author(s)

J.H. Maindonald

See Also

lm

Examples

CVlm()
Description

The Cars93.summary data frame has 6 rows and 4 columns created from information in the Cars93 data set in the Venables and Ripley MASS package. Each row corresponds to a different class of car (e.g. Compact, Large, etc.).

Usage

Cars93.summary

Format

This data frame contains the following columns:

- **Min.passengers**  minimum passenger capacity for each class of car
- **Max.passengers**  maximum passenger capacity for each class of car
- **No.of.cars**  number of cars in each class
- **abbrev**  a factor with levels C Compact, L Large, M Mid-Size, S Sm Small, Sp Sporty, V Van

Source


References

MASS library

Examples

type <- Cars93.summary$abbrev
type <- Cars93.summary[,4]
type <- Cars93.summary[,"abbrev"]
type <- Cars93.summary[[4]]  # Take the object that is stored
                              # in the fourth list element.
type
pause()

attach(Cars93.summary)
  # R can now access the columns of Cars93.summary directly
abbrev
detach("Cars93.summary")
pause()

# To change the name of the abbreviated variable (the fourth column)
names(Cars93.summary)[4] <- "code"
pause()

# To change all of the names, try
names(Cars93.summary) <- c("minpass","maxpass","number","code")
Lottario  
*Ontario Lottery Data*

**Description**

The data frame *Lottario* is a summary of 122 weekly draws of an Ontario lottery, beginning in November, 1978. Each draw consists of 7 numbered balls, drawn without replacement from an urn consisting of balls numbered from 1 through 39.

**Usage**

Lottario

**Format**

This data frame contains the following columns:

- **Number**: the integers from 1 to 39, representing the numbered balls
- **Frequency**: the number of occurrences of each numbered ball

**Source**

The Ontario Lottery Corporation

**References**


**Examples**

```r
order(Lottario$Frequency)[33:39]  # the 7 most frequently chosen numbers
```

Manitoba.lakes  
*The Nine Largest Lakes in Manitoba*

**Description**

The *Manitoba.lakes* data frame has 9 rows and 2 columns. The areas and elevations of the nine largest lakes in Manitoba, Canada. The geography of Manitoba (a relatively flat province) can be divided crudely into three main areas: a very flat prairie in the south which is at a relatively high elevation, a middle region consisting of mainly of forest and Precambrian rock, and a northern region which drains more rapidly into Hudson Bay. All water in Manitoba, which does not evaporate, eventually drains into Hudson Bay.

**Usage**

Manitoba.lakes
Format

This data frame contains the following columns:

- **elevation** a numeric vector consisting of the elevations of the lakes (in meters)
- **area** a numeric vector consisting of the areas of the lakes (in square kilometers)

Source

The CANSIM data base at Statistics Canada.

Examples

```r
plot(Manitoba.lakes)
plot(Manitoba.lakes[-1,])
```

ais  
**Australian athletes data set**

Description

These data were collected in a study of how data on various characteristics of the blood varied with sport body size and sex of the athlete.

Usage

```r
data(ais)
```

Format

A data frame with 202 observations on the following 13 variables.

- **rcc** red blood cell count, in $10^{12} l^{-1}$
- **wcc** white blood cell count, in $10^{12}$ per liter
- **hc** hematocrit, percent
- **hg** hemoglobin concentration, in g per decaliter
- **ferr** plasma ferritins, ng $dl^{-1}$
- **bmi** Body mass index, kg cm$^{-2} 10^2$
- **ssf** sum of skin folds
- **pcBfat** percent Body fat
- **lhm** lean body mass, kg
- **ht** height, cm
- **wt** weight, kg
- **sex** a factor with levels f m
- **sport** a factor with levels B_Ball Field Gym Netball Row Swim T_400m T_Sprnt Tennis W_Polo
Details

Do blood hemoglobin concentrations of athletes in endurance-related events differ from those in power-related events?

Source

These data were the basis for the analyses that are reported in Telford and Cunningham (1991).

References


allbacks

Measurements on a Selection of Books

Description

The allbacks data frame gives measurements on the volume and weight of 15 books, some of which are softback (pb) and some of which are hardback (hb). Area of the hardback covers is also included.

Usage

allbacks

Format

This data frame contains the following columns:

- **volume** book volumes in cubic centimeters
- **area** hard board cover areas in square centimeters
- **weight** book weights in grams
- **cover** a factor with levels hb hardback, pb paperback

Source

The bookshelf of J. H. Maindonald.

Examples

```r
print("Multiple Regression - Example 6.1")
attach(allbacks)
volume.split <- split(volume, cover)
weight.split <- split(weight, cover)
plot(weight.split$hb ~ volume.split$hb, pch=16, xlim=range(volume), ylim=range(weight),
     ylab="Weight (g)", xlab="Volume (cc)"
points(weight.split$pb ~ volume.split$pb, pch=16, col=2)
pause()
allbacks.lm <- lm(weight ~ volume+area)
summary(allbacks.lm)
```
### anesthetic

**Description**

Thirty patients were given an anesthetic agent maintained at a predetermined level (conc) for 15 minutes before making an incision. It was then noted whether the patient moved, i.e. jerked or twisted.

**Usage**

```r
anesthetic
```

**Format**

This data frame contains the following columns:

- **move**: a binary numeric vector coded for patient movement (0 = no movement, 1 = movement)
- **conc**: anesthetic concentration
- **logconc**: logarithm of concentration
- **nomove**: the complement of move

**Details**

The interest is in estimating how the probability of jerking or twisting varies with increasing concentration of the anesthetic agent.
print("Logistic Regression - Example 8.1.4")

z <- table(anesthetic$nomove, anesthetic$conc)
tot <- apply(z, 2, sum)  # totals at each concentration
prop <- z[2, ]/tot       # proportions at each concentration
oprop <- sum(z[2, ])/sum(tot)  # expected proportion moving if concentration had no effect
conc <- as.numeric(dimnames(z)[[2]])
plot(conc, prop, xlab = "Concentration", ylab = "Proportion", xlim = c(.5, 2.5),
     ylim = c(0, 1), pch = 16)
chw <- par()$cxy[1]
text(conc - 0.75 * chw, prop, paste(tot), adj = 1)
abline(h = oprop, lty = 2)
pause()

anes.logit <- glm(nomove ~ conc, family = binomial(link = logit),
                   data = anesthetic)
anova(anes.logit)
summary(anes.logit)

ant111b

Averages by block of corn yields, for treatment 111 only

Description

These data frames have averages by blocks (parcels) for the treatment 111.

Usage

ant111b

Format

A data frame with 36 observations on 9 variables.

site a factor with levels (ant111b) DBAN LFAN NSAN ORAN OVAN TEAN WEAN WLAN
parcel a factor with levels I II III IV
code a numeric vector
island a numeric vector
id a numeric vector
plot a numeric vector
trt a numeric vector
ears a numeric vector
harvwt a numeric vector
antigua

Source

Averages by block of yields for the Antigua Corn data

Description
These data frames have yield averages by blocks (parcels). The ant11ib data set is a subset of this.

Usage
antigua

Format
A data frame with 324 observations on 7 variables.

id  a numeric vector
site  a factor with 8 levels.
block  a factor with levels I II III IV
plot  a numeric vector
trt  a factor consisting of 12 levels
ears  a numeric vector; note that -9999 is used as a missing value code.
harvwt  a numeric vector; the average yield

Source

appletaste

Tasting experiment that compared four apple varieties

Description
Each of 20 tasters each assessed three out of the four varieties. The experiment was conducted according to a balanced incomplete block design.

Usage
data(appletaste)
Format

A data frame with 60 observations on the following 3 variables.

aftertaste a numeric vector Apple samples were rated for aftertaste, by making a mark on a continuous scale that ranged from 0 (extreme dislike) to 150 (like very much).

panelist a factor with levels abcdefghijklmnopqrstuvwxyz

product a factor with levels 298 493 649 937

Examples

data(appletaste)
appletaste.aov <- aov(aftertaste ~ panelist + product, data=appletaste)
termplot(appletaste.aov)


---

austpop

Population figures for Australian States and Territories

Description


Usage

austpop

Format

This data frame contains the following columns:

year a numeric vector

NSW New South Wales population counts

Vic Victoria population counts

Qld Queensland population counts

SA South Australia population counts

WA Western Australia population counts

Tas Tasmania population counts

NT Northern Territory population counts

ACT Australian Capital Territory population counts

Aust Population counts for the whole country

Source

Australian Bureau of Statistics
**Examples**

```r
def_print("Looping - Example 1.7")

growth.rates <- numeric(8)
for (j in seq(2,9)) {
growth.rates[j-1] <- (austpop[9, j]-austpop[1, j])/austpop[1, j]
}
growth.rates <- data.frame(growth.rates)
row.names(growth.rates) <- names(austpop[c(-1,-10)])
# Note the use of row.names() to name the rows of the data frame

growth.rates

def_print("Avoiding Loops - Example 1.7b")
sapply(austpop[,c(-1,10)], function(x){(x[9]-x[1])/x[1]})

def_print("Plot - Example 1.8a")
attach(austpop)
plot(year, ACT, type="l") # Join the points ("l" = "line")
detach(austpop)

def_print("Exerice 1.12.9")
attach(austpop)
oldpar <- par(mfrow=c(2,4))
for (i in 2:9){
plot(austpop[,1], log(austpop[, i]), xlab="Year",
     ylab=names(austpop)[i], pch=16, ylim=c(0,10))
par(oldpar)
detach(austpop)
```

---

**bestset.noise**

**Best Subset Selection Applied to Noise**

**Description**

Best subset selection applied to completely random noise. This function demonstrates how variable selection techniques in regression can often err in suggesting that more variables be included in a regression model than necessary.

**Usage**

```r
bestset.noise(m=100, n=40)
```

**Arguments**

- `m` the number of observations to be simulated.
- `n` the number of predictor variables in the simulated model.
Details

A set of \( n \) predictor variables are simulated as independent standard normal variates, in addition to a response variable which is also independent of the predictors. The best three variable model relating the response to the predictors is selected using functions from the leaps package. (The leaps package must be installed in order for this function to work.)

Value

`bestset.noise` returns a list obtained from the `summary.lm` function.

Author(s)

J.H. Maindonald

See Also

`lm`

Examples

```r
leaps.out <- try(require(leaps, quietly=TRUE))
leaps.out.log <- is.logical(leaps.out)
if ((leaps.out.log==TRUE)&(leaps.out==TRUE))
bestset.noise(20,6) # 'best' 3-variable regression for 20 simulated observations
# on 7 unrelated variables (including the response)
```

---

### biomass

**Biomass Data**

#### Description

The `biomass` data frame has 135 rows and 8 columns. The `rainforest` data frame is a subset of this one.

#### Usage

`biomass`

#### Format

This data frame contains the following columns:

- `dbh` a numeric vector
- `wood` a numeric vector
- `bark` a numeric vector
- `fac26` a factor with 3 levels
- `root` a numeric vector
- `rootsk` a numeric vector
- `branch` a numeric vector
- `species` a factor with levels Acacia mabellae, C. fraseri, Acmena smithii, B. myrtifolia
**Source**

J. Ash, Australian National University

**References**


---

**Southern Oscillation Index Data**

**Description**

The Southern Oscillation Index (SOI) is the difference in barometric pressure at sea level between Tahiti and Darwin. Annual SOI and Australian rainfall data, for the years 1900-2001, are given. Australia’s annual mean rainfall is an area-weighted average of the total annual precipitation at approximately 370 rainfall stations around the country.

**Usage**

bomsoi

**Format**

This data frame contains the following columns:

- **Year** a numeric vector
- **Jan** average January SOI values for each year
- **Feb** average February SOI values for each year
- **Mar** average March SOI values for each year
- **Apr** average April SOI values for each year
- **May** average May SOI values for each year
- **Jun** average June SOI values for each year
- **Jul** average July SOI values for each year
- **Aug** average August SOI values for each year
- **Sep** average September SOI values for each year
- **Oct** average October SOI values for each year
- **Nov** average November SOI values for each year
- **Dec** average December SOI values for each year
- **SOI** a numeric vector consisting of average annual SOI values
- **avrain** a numeric vector consisting of a weighted average annual rainfall at a large number of Australian sites
- **NTrain** Northern Territory rain
- **northRain** north rain
- **seRain** southeast rain
- **eastRain** east rain
- **southRain** south rain
- **swRain** southwest rain
Australian Bureau of Meteorology web pages:


Examples

```r
plot(ts(bomsoi[, 15:14], start=1900),
     panel=function(y,...)panel.smooth(1900:2005, y,...))
pause()

# Check for skewness by comparing the normal probability plots for different a, e.g.
par(mfrow = c(2,3))
for (a in c(50, 100, 150, 200, 250, 300))
  qnorm(log(bomsoi[, "avrain"] - a))
  # a = 250 leads to a nearly linear plot

par(mfrow = c(1,1))
plot(bomsoi$SOI, log(bomsoi$avrain - 250), xlab = "SOI",
     ylab = "log(avrain = 250)"
lines(lowess(bomsoi$SOI)$y, lowess(log(bomsoi$avrain - 250))$y, lwd=2)
  # NB: separate lowess fits against time
lines(lowess(bomsoi$SOI, log(bomsoi$avrain - 250)))
pause()

xbomsoi <-
  with(bomsoi, data.frame(SOI=SOI, cuberootRain=avrain^0.33))
xbomsoi$trendSOI <- lowess(xbomsoi$SOI)$y
xbomsoi$trendRain <- lowess(xbomsoi$cuberootRain)$y
rainpos <- pretty(bomsoi$avrain, 5)
with(xbomsoi,
  (plot(cuberootRain ~ SOI, xlab = "SOI",
       ylab = "Rainfall (cube root scale)", yaxt="n")
  axis(2, at = rainpos^0.33, labels=paste(rainpos))
  ## Relative changes in the two trend curves
  lines(lowess(cuberootRain ~ SOI))
  lines(lowess(trendRain ~ trendSOI), lwd=2)
)
pause()

xbomsoi$detrendRain <-
  with(xbomsoi, cuberootRain - trendRain + mean(trendRain))
xbomsoi$detrendSOI <-
  with(xbomsoi, SOI - trendSOI + mean(trendSOI))
oldpar <- par(mfrow=c(1,2), pty="m")
plot(cuberootRain ~ SOI, data = xbomsoi,
     ylab = "Rainfall (cube root scale)", yaxt="n")
axis(2, at = rainpos^0.33, labels=paste(rainpos))
```
with(xbomsoi, lines(lowess(cuberootRain ~ SOI)))
plot(detrendRain ~ detrendSOI, data = xbomsoi,
    xlab="Detrended SOI", ylab = "Detrended rainfall", yaxt="n")
axis(2, at = rainpos^0.33, labels=paste(rainpos))
with(xbomsoi, lines(lowess(detrendRain ~ detrendSOI)))
pause()
par(oldpar)
attach(xbomsoi)
xbomsoi.ma0 <- arima(detrendRain, xreg=detrendSOI, order=c(0,0,0))
# ordinary regression model
xbomsoi.ma12 <- arima(detrendRain, xreg=detrendSOI,
    order=c(0,0,12))
# regression with MA(12) errors -- all 12 MA parameters are estimated
xbomsoi.ma12
pause()
xbomsoi.ma12s <- arima(detrendRain, xreg=detrendSOI,
    seasonal=list(order=c(0,0,1), period=12))
# regression with seasonal MA(1) (lag 12) errors -- only 1 MA parameter
# is estimated
xbomsoi.ma12s
pause()
xbomsoi.maSel <- arima(x = detrendRain, order = c(0, 0, 12),
    xreg = detrendSOI, fixed = c(0, 0, 0,
    NA, rep(0, 4), NA, 0, NA, NA, NA, NA),
    transform.pars=FALSE)
# error term is MA(12) with fixed 0's at lags 1, 2, 3, 5, 6, 7, 8, 10
# NA's are used to designate coefficients that still need to be estimated
# transform.pars is set to FALSE, so that MA coefficients are not
# transformed (see help(arima))
detach(xbomsoi)
pause()
Box.test(resid(lm(detrendRain ~ detrendSOI, data = xbomsoi)),
    type="Ljung-Box", lag=20)
pause()
attach(xbomsoi)
xbomsoi2.maSel <- arima(x = detrendRain, order = c(0, 0, 12),
    xreg = poly(detrendSOI,2), fixed = c(0,
    0, 0, NA, rep(0, 4), NA, 0, rep(NA,5)),
    transform.pars=FALSE)
xbomsoi2.maSel
qqnorm(resid(xbomsoi.maSel, type="normalized"))
detach(xbomsoi)
Description

The Southern Oscillation Index (SOI) is the difference in barometric pressure at sea level between Tahiti and Darwin. Annual SOI and Australian rainfall data, for the years 1900-2001, are given. Australia’s annual mean rainfall is an area-weighted average of the total annual precipitation at approximately 370 rainfall stations around the country.

Usage

bomsoi2001

Format

This data frame contains the following columns:

- **Year**: a numeric vector
- **Jan**: average January SOI values for each year
- **Feb**: average February SOI values for each year
- **Mar**: average March SOI values for each year
- **Apr**: average April SOI values for each year
- **May**: average May SOI values for each year
- **Jun**: average June SOI values for each year
- **Jul**: average July SOI values for each year
- **Aug**: average August SOI values for each year
- **Sep**: average September SOI values for each year
- **Oct**: average October SOI values for each year
- **Nov**: average November SOI values for each year
- **Dec**: average December SOI values for each year
- **SOI**: a numeric vector consisting of average annual SOI values
- **avrain**: a numeric vector consisting of a weighted average annual rainfall at a large number of Australian sites

Source

Australian Bureau of Meteorology web pages:


References


See Also

bomsoi
Examples

```r
bomsoi <- bomsoi2001
plot(ts(bomsoi[, 15:14], start=1900),
     panel=function(y,...)panel.smooth(1900:2001, y,...))
pause()

# Check for skewness by comparing the normal probability plots for
# different a, e.g.
par(mfrow = c(2,3))
for (a in c(50, 100, 150, 200, 250, 300))
  qqnorm(log(bomsoi[, "avrain"] - a))
  # a = 250 leads to a nearly linear plot
pause()

par(mfrow = c(1,1))
plot(bomsoi$SOI, log(bomsoi$avrain - 250), xlab = "SOI",
     ylab = "log(avrain = 250)"
lines(lowess(bomsoi$SOI)$y, lowess(log(bomsoi$avrain - 250))$y, lwd=2)
  # NB: separate lowess fits against time
lines(lowess(bomsoi$SOI, log(bomsoi$avrain - 250)))
pause()

xbomsoi <-
  with(bomsoi, data.frame(SOI=SOI, cuberootRain=avrain^0.33))
xbomsoi$trendSOI <- lowess(xbomsoi$SOI)$y
xbomsoi$trendRain <- lowess(xbomsoi$cuberootRain)$y
rainpos <- pretty(bomsoi$avrain, 5)
with(xbomsoi,
  {plot(cuberootRain ~ SOI, xlab = "SOI",
       ylab = "Rainfall (cube root scale)", yaxt="n")
   axis(2, at = rainpos^0.33, labels=paste(rainpos))
   ## Relative changes in the two trend curves
   lines(lowess(cuberootRain ~ SOI))
   lines(lowess(trendRain ~ trendSOI), lwd=2)
})
pause()

xbomsoi$detrendRain <-
  with(xbomsoi, cuberootRain - trendRain + mean(trendRain))
xbomsoi$detrendSOI <-
  with(xbomsoi, SOI - trendSOI + mean(trendSOI))
oldpar <- par(mfrow=c(1,2), pty="s")
plot(cuberootRain ~ SOI, data = xbomsoi,
     ylab = "Rainfall (cube root scale)", yaxt="n")
axis(2, at = rainpos^0.33, labels=paste(rainpos))
with(xbomsoi, lines(lowess(cuberootRain ~ SOI)))
plot(detrendRain ~ detrendSOI, data = xbomsoi,
     xlab="Detrended SOI", ylab = "Detrended rainfall", yaxt="n")
axis(2, at = rainpos^0.33, labels=paste(rainpos))
with(xbomsoi, lines(lowess(detrendRain ~ detrendSOI)))
pause()

par(oldpar)
attach(xbomsoi)
xbomsoi.ma0 <- arima(detrendRain, xreg=detrendSOI, order=c(0,0,0))
```
# ordinary regression model
xbomsoi.ma12 <- arima(detrendRain, xreg=detrendSOI, 
    order=c(0,0,12))
# regression with MA(12) errors -- all 12 MA parameters are estimated
xbomsoi.ma12
pause()

xbomsoi.ma12s <- arima(detrendRain, xreg=detrendSOI, 
    seasonal=list(order=c(0,0,1), period=12))
# regression with seasonal MA(1) (lag 12) errors -- only 1 MA parameter
# is estimated
xbomsoi.ma12s
pause()

xbomsoi.maSel <- arima(x = detrendRain, order = c(0, 0, 12),
    xreg = detrendSOI, fixed = c(0, 0, 0, NA, rep(0, 4), NA, 0, NA, NA, NA, NA),
    transform.pars=FALSE)
# error term is MA(12) with fixed 0's at lags 1, 2, 3, 5, 6, 7, 8, 10
# NA's are used to designate coefficients that still need to be estimated
# transform.pars is set to FALSE, so that MA coefficients are not
# transformed (see help(arima))
detach(xbomsoi)
pause()

Box.test(resid(lm(detrendRain ~ detrendSOI, data = xbomsoi)),
    type="Ljung-Box", lag=20)
pause()

attach(xbomsoi)
xbomsoi2.maSel <- arima(x = detrendRain, order = c(0, 0, 12),
    xreg = poly(detrendSOI,2), fixed = c(0, 0, 0, NA, rep(0, 4), NA, 0, rep(NA,5)),
    transform.pars=FALSE)

xbomsoi2.maSel
qqnorm(resid(xbomsoi.maSel, type="normalized"))
detach(xbomsoi)

bostonc

---

bostonc Boston Housing Data – Corrected

Description

The corrected Boston housing data (from http://lib.stat.cmu.edu/datasets/).

Usage

bostonc
bounce

Format
A single vector containing the contents of "boston_corrected.txt".

Source

Description
Return univariate plotting positions in which neighboring points are separated, if and as necessary, so that they are the specified minimum distance apart.

Usage
bounce(y, d, log = FALSE)

Arguments
y A numeric vector of plotting positions
d Minimum required distance between neighboring positions
log TRUE if values are will be plotted on a logarithmic scale.

Details
The centroid(s) of groups of points that are moved relative to each other remain the same.

Value
A vector of values such that, when plotted along a line, neighboring points are the required minimum distance apart.

Note
If values are plotted on a logarithmic scale, d is the required distance apart on that scale. If a base other than 10 is required, set log equal to that base. (Note that base 10 is the default for plot with log=TRUE.)

Author(s)
John Maindonald

See Also
See also onewayPlot

Examples
bounce(c(4, 1.8, 2, 6), d=.4)
bounce(c(4, 1.8, 2, 6), d=.1, log=TRUE)
capstring  
*Converts initial character of a string to upper case*

**Description**

This function is useful for use before plotting, if one wants capitalized axis labels or factor levels.

**Usage**

```r
capstring(names)
```

**Arguments**

- `names` a character vector

**Value**

a character vector with upper case initial values

**Author(s)**

W.J. Braun

**Examples**

```r
capstring(names(tinting)[c(3,4)])
library(lattice)
levels(tinting$agegp) <- capstring(levels(tinting$agegp))
xyplot(csoa ~ it | sex * agegp, data=tinting)
```

---

carprice  
*US Car Price Data*

**Description**

U.S. data extracted from Cars93, a data frame in the MASS package.

**Usage**

```r
carprice
```
Format

This data frame contains the following columns:

Type  Type of car, e.g. Sporty, Van, Compact
Min.Price  Price for a basic model
Price  Price for a mid-range model
Max.Price  Price for a ‘premium’ model
Range.Price  Difference between Max.Price and Min.Price
RoughRange  Rough.Range plus some $N(0,.0001)$ noise
gpm100  The number of gallons required to travel 100 miles
MPG.city  Average number of miles per gallon for city driving
MPG.highway  Average number of miles per gallon for highway driving

Source

MASS package

References


Examples

```r
print("Multicollinearity - Example 6.8")
pairs(carprice[,-c(1,8,9)])

carprice1.lm <- lm(gpm100 ~ Type+Min.Price+Price+Max.Price+Range.Price,
data=carprice)
round(summary(carprice1.lm)$coef,3)
pause()

alias(carprice1.lm)
pause()

carprice2.lm <- lm(gpm100 ~ Type+Min.Price+Price+Max.Price+RoughRange, data=carprice)
round(summary(carprice2.lm)$coef, 2)
pause()

carprice.lm <- lm(gpm100 ~ Type + Price, data = carprice)
round(summary(carprice.lm)$coef,4)
pause()

summary(carprice1.lm)$sigma  # residual standard error when fitting all 3 price variables
pause()

summary(carprice.lm)$sigma  # residual standard error when only price is used
pause()

vif(lm(gpm100 ~ Price, data=carprice))  # Baseline Price
pause()
```
Description

The cfseal data frame has 30 rows and 11 columns consisting of weight measurements for various organs taken from 30 Cape Fur Seals that died as an unintended consequence of commercial fishing.

Usage

cfseal

Format

This data frame contains the following columns:

- age  a numeric vector
- weight  a numeric vector
- heart  a numeric vector
- lung  a numeric vector
- liver  a numeric vector
- spleen  a numeric vector
- stomach  a numeric vector
- leftkid  a numeric vector
- rightkid  a numeric vector
- kidney  a numeric vector
- intestines  a numeric vector

Source

Examples

print("Allometric Growth - Example 5.7")

cfseal.lm <- lm(log(heart) ~ log(weight), data=cfseal); summary(cfseal.lm)
plot(log(heart) ~ log(weight), data = cfseal, pch=16, xlab = "Heart Weight (g, log scale)", ylab = "Body weight (kg, log scale)", axes=FALSE)
heartaxis <- 100*(2^seq(0,3))
bodyaxis <- c(20,40,60,100,180)
axis(1, at = log(bodyaxis), lab = bodyaxis)
axis(2, at = log(heartaxis), lab = heartaxis)
box()
abline(cfseal.lm)

cities

Populations of Major Canadian Cities (1992-96)

Description

Population estimates for several Canadian cities.

Usage

cities

Format

This data frame contains the following columns:

- CITY a factor, consisting of the city names
- REGION a factor with 5 levels (ATL=Atlantic, ON=Ontario, QC=Quebec, PR=Prairies, WEST=Alberta and British Columbia) representing the location of the cities
- POP1992 a numeric vector giving population in 1000’s for 1992
- POP1993 a numeric vector giving population in 1000’s for 1993
- POP1994 a numeric vector giving population in 1000’s for 1994
- POP1995 a numeric vector giving population in 1000’s for 1995
- POP1996 a numeric vector giving population in 1000’s for 1996

Source

Statistics Canada

Examples

cities$have <- factor((cities$REGION=="ON")|(cities$REGION=="WEST"))
plot(POP1996~POP1992, data=cities, col=as.integer(cities$have))
Dose-mortality data, for fumigation of codling moth with methyl bromide

Description

Data are from trials that studied the mortality response of codling moth to fumigation with methyl bromide.

Usage

data(codling)

Format

A data frame with 99 observations on the following 10 variables.

dose Injected dose of methyl bromide, in gm per cubic meter
tot Number of insects in chamber
dead Number of insects dying
pobs Proportion dying
cm Control mortality, i.e., at dose 0
ct Concentration-time sum
Cultivar a factor with levels BRAEBURN FUJI GRANNY Gala ROYAL Red Delicious Splendour
gp a factor which has a different level for each different combination of Cultivar, year and rep (replicate).
year a factor with levels 1988 1989
numcm a numeric vector: total number of control insects

Details

The research that generated these data was in part funded by New Zealand pipfruit growers. The published analysis was funded by New Zealand pipfruit growers. See also sorption.

Source

compareTreecalcs  Error rate comparisons for tree-based classification

Description

Compare error rates, between different functions and different selection rules, for an approximately equal random division of the data into a training and test set.

Usage

```r
compareTreecalcs(x = yesno ~ ., data = spam7, cp = 0.00025,
               fun = c("rpart", "randomForest"))
```

Arguments

- **x**: model formula
- **data**: an data frame in which to interpret the variables named in the formula
- **cp**: setting for the cost complexity parameter cp, used by rpart()
- **fun**: one or both of "rpart" and "randomForest"

Details

Data are randomly divided into two subsets, I and II. The function(s) are used in the standard way for calculations on subset I, and error rates returned that come from the calculations carried out by the function(s). Predictions are made for subset II, allowing the calculation of a completely independent set of error rates.

Value

If rpart is specified in fun, the following:

- **rpSEcvI**: the estimated cross-validation error rate when rpart() is run on the training data (I), and the one-standard error rule is used
- **rpcvI**: the estimated cross-validation error rate when rpart() is run on subset I, and the model used that gives the minimum cross-validated error rate
- **rpSEtest**: the error rate when the model that leads to rpSEcvI is used to make predictions for subset II
- **rptest**: the error rate when the model that leads to rpcvI is used to make predictions for subset II
- **nSErule**: number of splits required by the one standard error rule
- **nREmin**: number of splits to give the minimum error
- **rfcvI**: the out-of-bag (OOB) error rate when randomForest() is run on subset I
- **rfest**: the error rate when the model that leads to rfcvI is used to make predictions for subset II

Author(s)

John Maindonald
component.residual  Component + Residual Plot

Description

Component + Residual plot for a term in a \texttt{lm} model.

Usage

\begin{verbatim}
component.residual(lm.obj = mice12.lm, which = 1, xlab = "Component", ylab = "C+R")
\end{verbatim}

Arguments

- \texttt{lm.obj} \hspace{1cm} A \texttt{lm} object
- \texttt{which} \hspace{1cm} numeric code for the term in the \texttt{lm} formula to be plotted
- \texttt{xlab} \hspace{1cm} label for the x-axis
- \texttt{ylab} \hspace{1cm} label for the y-axis

Value

A scatterplot with a smooth curve overlaid.

Author(s)

J.H. Maindonald

See Also

\texttt{lm}

Examples

\begin{verbatim}
mice12.lm <- lm(brainwt ~ bodywt + lsize, data=litters)
oldpar <- par(mfrow = c(1,2))
component.residual(mice12.lm, 1, xlab = "Body weight", ylab= "t(Body weight) + e")
component.residual(mice12.lm, 2, xlab = "Litter size", ylab= "t(Litter size) + e")
par(oldpar)
\end{verbatim}
Description

Numbers are given in different categories of worker, in each of two investigations. The first source of information is the Board of Trade Census that was conducted on 1886. The second is a relatively informal survey conducted by US Bureau of Labor representatives in 1889, for use in official reports.

Usage

data(cottonworkers)

Format

A data frame with 14 observations on the following 3 variables.

- **census1886** Numbers of workers in each of 14 different categories, according to the Board of Trade wage census that was conducted in 1886
- **survey1889** Numbers of workers in each of 14 different categories, according to data collected in 1889 by the US Bureau of Labor, for use in a report to the US Congress and House of Representatives
- **avwage** Average wage, in pence, as estimated in the US Bureau of Labor survey

Details

The data in *survey1889* were collected in a relatively informal manner, by approaching individuals on the street. Biases might therefore be expected.

Source


References


Examples

data(cottonworkers)
str(cottonworkers)
plot(survey1889 ~ census1886, data=cottonworkers)
plot(I(avwage*survey1889) ~ I(avwage*census1886), data=cottonworkers)
Comparison of cuckoo eggs with host eggs

Description
These data compare mean length, mean breadth, and egg color, between cuckoos and their hosts.

Usage
cuckoohosts

Format
A data frame with 10 observations on the following 12 variables.

- **clength**: mean length of cuckoo eggs in given host’s nest
- **cl.sd**: standard deviation of cuckoo egg lengths
- **cbreadth**: mean breadth of cuckoo eggs in given host’s nest
- **cb.sd**: standard deviation of cuckoo egg breadths
- **cnum**: number of cuckoo eggs
- **hlength**: length of host eggs
- **hl.sd**: standard deviation of host egg lengths
- **hbreadth**: breadth of host eggs
- **hb.sd**: standard deviation of host egg breadths
- **hnum**: number of host eggs
- **match**: number of eggs where color matched
- **nomatch**: number of eggs where color did not match

Details
Although from the same study that generated data in the data frame **cuckoos**, the data do not match precisely. The cuckoo egg lengths and breadths are from the tables on page 168, the host egg lengths and breadths from Appendix IV on page 176, and the color match counts from the table on page 171.

Source
Latter, O.H., 1902. The egg of *cuculus canorus*. an inquiry into the dimensions of the cuckoo’s egg and the relation of the variations to the size of the eggs of the foster-parent, with notes on coloration, &c. *Biometrika*, 1:164–176.

Examples
cuckoo hosts
str(cuckoo hosts)
plot(cuckoo hosts)
with(cuckoo hosts,
    plot(c(clength,hlength),c(cbreadth,hbreadth),col=rep(1:2,c(6,6))))
**Description**

Length and breadth measurements of 120 eggs lain in the nests of six different species of host bird.

**Usage**

cuckoos

**Format**

This data frame contains the following columns:

- `length` the egg lengths in millimeters
- `breadth` the egg breadths in millimeters
- `species` a factor with levels `hedge.sparrow`, `meadow.pipit`, `pied.wagtail`, `robin`, `tree.pipit`, `wren`
- `id` a numeric vector

**Source**

Latter, O.H. (1902). The eggs of Cuculus canorus. An Inquiry into the dimensions of the cuckoo's egg and the relation of the variations to the size of the eggs of the foster-parent, with notes on coloration, &c. Biometrika i, 164.

**References**


**Examples**

```r
print("Strip and Boxplots - Example 2.1.2")
attach(cuckoos)
oldpar <- par(las = 2) # labels at right angle to axis.
stripchart(length ~ species)
boxplot(split(cuckoos$length, cuckoos$species),
   xlab="Length of egg", horizontal=TRUE)
detach(cuckoos)
par(oldpar)
pause()

print("Summaries - Example 2.2.2")
sapply(split(cuckoos$length, cuckoos$species), sd)
pause()

print("Example 4.1.4")
wren <- split(cuckoos$length, cuckoos$species)$wren
median(wren)
n <- length(wren)
sqrt(pi/2)*sd(wren)/sqrt(n) # this s.e. computation assumes normality
```

---

**cuckoos**  

**Cuckoo Eggs Data**

**Description**

Length and breadth measurements of 120 eggs lain in the nests of six different species of host bird.

**Usage**

cuckoos

**Format**

This data frame contains the following columns:

- `length` the egg lengths in millimeters
- `breadth` the egg breadths in millimeters
- `species` a factor with levels `hedge.sparrow`, `meadow.pipit`, `pied.wagtail`, `robin`, `tree.pipit`, `wren`
- `id` a numeric vector

**Source**

Latter, O.H. (1902). The eggs of Cuculus canorus. An Inquiry into the dimensions of the cuckoo’s egg and the relation of the variations to the size of the eggs of the foster-parent, with notes on coloration, &c. Biometrika i, 164.

**References**


**Examples**

```r
print("Strip and Boxplots - Example 2.1.2")
attach(cuckoos)
oldpar <- par(las = 2) # labels at right angle to axis.
stripchart(length ~ species)
boxplot(split(cuckoos$length, cuckoos$species),
   xlab="Length of egg", horizontal=TRUE)
detach(cuckoos)
par(oldpar)
pause()

print("Summaries - Example 2.2.2")
sapply(split(cuckoos$length, cuckoos$species), sd)
pause()

print("Example 4.1.4")
wren <- split(cuckoos$length, cuckoos$species)$wren
median(wren)
n <- length(wren)
sqrt(pi/2)*sd(wren)/sqrt(n) # this s.e. computation assumes normality
```
Description

This function gives internal and cross-validation measures of predictive accuracy for regression with
a binary response. The data are randomly assigned to a number of ‘folds’. Each fold is removed,
in turn, while the remaining data is used to re-fit the regression model and to predict at the deleted
observations.

Usage

cv.binary(obj=frogs.glm, rand=NULL, nfolds=10, print.details=TRUE)

Arguments

obj
  a glm object
rand
  a vector which assigns each observation to a fold
nfolds
  the number of folds
print.details
  logical variable (TRUE = print detailed output, the default)

Value

the order in which folds were deleted

internal estimate of accuracy

cross-validation estimate of accuracy

Author(s)

J.H. Maindonald

See Also

glm

Examples

frogs.glm <- glm(pres.abs ~ log(distance) + log(NoOfPools),
                 family=binomial, data=frogs)
cv.binary(frogs.glm)
cv.lm

Cross-Validation for Linear Regression

Description
This function gives internal and cross-validation measures of predictive accuracy for ordinary linear regression. The data are randomly assigned to a number of ‘folds’. Each fold is removed, in turn, while the remaining data is used to re-fit the regression model and to predict at the deleted observations.

Usage

```
cv.lm(df = houseprices, form.lm = formula(sale.price ~ area), m=3, dots = FALSE, seed=29, plotit=TRUE, printit=TRUE)
```

Arguments

- `df`: a data frame
- `form.lm`: a formula object
- `m`: the number of folds
- `dots`: uses pch=16 for the plotting character
- `seed`: random number generator seed
- `plotit`: if TRUE, a plot is constructed on the active device
- `printit`: if TRUE, output is printed to the screen

Value

For each fold, a table listing

- the residuals
- `ms` = the overall mean square of prediction error

Author(s)

J.H. Maindonald

See Also

- `lm`

Examples

- `cv.lm()`
**datafile**

Create an ASCII data file

**Description**

Invoking this function creates one of four data files used in Chapters 1 and 14 of DAAGUR.

**Usage**

```r
datafile(file = "fuel")
```

**Arguments**

- `file` character; "fuel", for fuel.txt; "fuel.csv", for fuel.csv; "oneBadRow", for oneBadRow.txt; "scan-demo", for scan-demo.txt.

**Value**

One of four ASCII files is produced, and output to the current working directory.

**Author(s)**

J.H. Maindonald

---

**dengue**

Dengue prevalence, by administrative region

**Description**

Data record, for each of 2000 administrative regions, whether or not dengue was recorded at any time between 1961 and 1990.

**Usage**

```r
data(dengue)
```

**Format**

A data frame with 2000 observations on the following 13 variables.

- **humid** Average vapour density: 1961-1990
- **humid90** 90th percentile of humid
- **temp** Average temperature: 1961-1990
- **temp90** 90th percentile of temp
- **h10pix** maximum of humid, within a 10 pixel radius
- **h10pix90** maximum of humid90, within a 10 pixel radius
- **trees** Percent tree cover, from satellite data
- **trees90** 90th percentile of trees
dewpoint

NoYes  Was dengue observed? (1=yes)
Xmin  minimum longitude
Xmax  maximum longitude
Ymin  minimum latitude
Ymax  maximum latitude

Details

This is derived from a data set in which the climate and tree cover information were given for each half degree of latitude by half degree of longitude pixel. The variable NoYes was given by administrative region. The climate data and tree cover data given here are 50th or 90th percentiles, where percentiles were calculated across pixels for an administrative region.

Source

Simon Hales, Environmental Research New Zealand Ltd.

References


Examples

str(dengue)
glm(NoYes ~ humid, data=dengue, family=binomial)
glm(NoYes ~ humid90, data=dengue, family=binomial)

dewpoint  Dewpoint Data

Description

The dewpoint data frame has 72 rows and 3 columns. Monthly data were obtained for a number of sites (in Australia) and a number of months.

Usage
dewpoint

Format

This data frame contains the following columns:

maxtemp  monthly minimum temperatures
mintemp  monthly maximum temperatures
dewpt  monthly average dewpoint for each combination of minimum and maximum temperature readings (formerly dewpoint)
Source

Dr Edward Linacre, visiting fellow in the Australian National University Department of Geography.

Examples

```r
print("Additive Model - Example 7.5")
require(splines)
attach(dewpoint)
ds.lm <- lm(dewpt ~ bs(maxtemp,5) + bs(mintemp,5), data=dewpoint)
ds.fit <- predict(ds.lm, type="terms", se=TRUE)
oldpar <- par(mfrow=c(1,2))
plot(maxtemp, ds.fit$fit[,1], xlab="Maximum temperature", ylab="Change from dewpoint mean",type="n")
lines(maxtemp,ds.fit$fit[,1])
lines(maxtemp,ds.fit$fit[,1]-2*ds.fit$se[,1],lty=2)
lines(maxtemp,ds.fit$fit[,1]+2*ds.fit$se[,1],lty=2)
plot(mintemp,ds.fit$fit[,2],xlab="Minimum temperature", ylab="Change from dewpoint mean",type="n")
ord<-order(mintemp)
lines(mintemp[ord],ds.fit$fit[ord,2])
lines(mintemp[ord],ds.fit$fit[ord,2]-2*ds.fit$se[ord,2],lty=2)
lines(mintemp[ord],ds.fit$fit[ord,2]+2*ds.fit$se[ord,2],lty=2)
detach(dewpoint)
par(oldpar)
```

---

**elastic1**

**Elastic Band Data Replicated**

**Description**

The elastic1 data frame has 7 rows and 2 columns giving, for each amount by which an elastic band is stretched over the end of a ruler, the distance that the band traveled when released.

**Usage**

elastic1

**Format**

This data frame contains the following columns:

- **stretch** the amount by which the elastic band was stretched
- **distance** the distance traveled

**Source**

J. H. Maindonald
**Examples**

```r
plot(elastic1)

print("Inline Functions - Example 12.2.2")
sapply(elastic1, mean)
pause()

sapply(elastic1, function(x)mean(x))
pause()

sapply(elastic1, function(x)sum(log(x)))
pause()

print("Data Output - Example 12.3.2")
write.table(elastic1, file="bandsframe.txt")
```

---

**elastic2**    *Elastic Band Data Replicated Again*

**Description**

The `elastic2` data frame has 9 rows and 2 columns giving, for each amount by which an elastic band is stretched over the end of a ruler, the distance that the band traveled when released.

**Usage**

`elastic2`

**Format**

This data frame contains the following columns:

- `stretch` the amount by which the elastic band was stretched
- `distance` the distance traveled

**Source**

J. H. Maindonald

**Examples**

```r
plot(elastic2)
pause()

print("Chapter 5 Exercise")

yrange <- range(c(elastic1$distance, elastic2$distance))
xrange <- range(c(elastic1$stretch, elastic2$stretch))
plot(distance ~ stretch, data = elastic1, pch = 16, ylim = yrange, xlim = xrange)
points(distance ~ stretch, data = elastic2, pch = 15, col = 2)
legend(xrange[1], yrange[2], legend = c("Data set 1", "Data set 2"), pch = 16)
```
elasticband <- lm(distance ~ stretch, data = elastic1)
elastic2.lm <- lm(distance ~ stretch, data = elastic2)
abline(elastic1.lm)
abline(elastic2.lm, col = 2)
summary(elastic1.lm)
summary(elastic2.lm)
pause()
predict(elastic1.lm, se.fit=TRUE)
predict(elastic2.lm, se.fit=TRUE)

elasticband  

Elastic Band Data

Description

The elasticband data frame has 7 rows and 2 columns giving, for each amount by which an elastic band is stretched over the end of a ruler, the distance that the band traveled when released.

Usage

elasticband

Format

This data frame contains the following columns:

- **stretch**: the amount by which the elastic band was stretched
- **distance**: the distance traveled

Source

J. H. Maindonald

Examples

print("Example 1.8.1")

attach(elasticband)  # R now knows where to find stretch and distance
plot(stretch, distance)  # Alternative: plot(distance ~ stretch)
detach(elasticband)
pause()

print("Output of Data Frames - Example 12.3.2")

write(t(elasticband), file="bands.txt", ncol=2)
sink("bands2.txt")
elasticband  # NB: No output on screen
sink()

print("Lists - Example 12.7")
elastic.lm <- lm(distance ~ stretch, data=elasticband)
names(elastic.lm)
elastic.lm$coefficients
elastic.lm[["coefficients"]]
pause()

elastic.lm[[1]]
pause()

elastic.lm[1]
pause()

options(digits=3)
elastic.lm$residuals
pause()

elastic.lm$call
pause()

mode(elastic.lm$call)

---

fossilfuel  Fossil Fuel Data

Description

Usage

fossilfuel

Format

This data frame contains the following columns:

- **year** a numeric vector giving the year the measurement was taken.
- **carbon** a numeric vector giving the total worldwide carbon emissions from fossil fuel use, in millions of tonnes.

Source


Examples

plot(fossilfuel)
The *frogs* data frame has 212 rows and 11 columns. The data are on the distribution of the Southern Corroboree frog, which occurs in the Snowy Mountains area of New South Wales, Australia.

**Usage**

*frogs*

**Format**

This data frame contains the following columns:

- **pres.abs** 0 = frogs were absent, 1 = frogs were present
- **northing** reference point
- **easting** reference point
- **altitude** altitude, in meters
- **distance** distance in meters to nearest extant population
- **NoOfPools** number of potential breeding pools
- **NoOfSites** (number of potential breeding sites within a 2 km radius
- **avrain** mean rainfall for Spring period
- **meanmin** mean minimum Spring temperature
- **meanmax** mean maximum Spring temperature

**Source**


**Examples**

```r
print("Multiple Logistic Regression - Example 8.2")

plot(northing ~ easting, data=frogs, pch=c(1,16)[frogs$pres.abs+1],
     xlab="Meters east of reference point", ylab="Meters north")

pause()

oldpar <- par(oma=c(2,2,2,2), cex=0.5)
pairs(frogs[,4:10])
par(oldpar)

pause()

oldpar <- par(mfrow=c(1,3))
for(nam in c("distance","NoOfPools","NoOfSites")){
y <- frogs[,nam]
...}
```
plot(density(y), main="", xlab=nam)
par(oldpar)
}
pause()

attach(frogs)
pairs(cbind(altitude, log(distance), log(NoOfPools), NoOfSites),
    panel=panel.smooth, labels=c("altitude", "log(distance)",
    "log(NoOfPools)", "NoOfSites"))
detach(frogs)

frogs.glm0 <- glm(formula = pres.abs ~ altitude + log(distance) +
    log(NoOfPools) + NoOfSites + avrain + meanmin + meanmax,
    family = binomial, data = frogs)
summary(frogs.glm0)
pause()

frogs.glm <- glm(formula = pres.abs ~ log(distance) + log(NoOfPools) +
    meanmin + meanmax, family = binomial, data = frogs)
oldpar <- par(mfrow=c(2,2))
termplot(frogs.glm, data=frogs)
par(oldpar)
pause()

termplot(frogs.glm, data=frogs, partial.resid=TRUE)
cv.binary(frogs.glm0)  # All explanatory variables
pause()
cv.binary(frogs.glm)   # Reduced set of explanatory variables
pause()

for (j in 1:4){
  rand <- sample(1:10, 212, replace=TRUE)
  all.acc <- cv.binary(frogs.glm0, rand=rand, print.details=FALSE)$acc.cv
  reduced.acc <- cv.binary(frogs.glm, rand=rand, print.details=FALSE)$acc.cv
  cat("\nAll:", round(all.acc,3), " Reduced:", round(reduced.acc,3))
}
Format

This data frame contains the following columns:

- **Lab** careful laboratory analysis measurements
- **HPLC** high performance liquid chromatography measurements

Source

W. J. Braun

---

| fruitohms  | Electrical Resistance of Kiwi Fruit |

Description

Data are from a study that examined how the electrical resistance of a slab of kiwifruit changed with the apparent juice content.

Usage

fruitohms

Format

This data frame contains the following columns:

- **juice** apparent juice content (percent)
- **ohms** electrical resistance (in ohms)

Source


Examples

```r
plot(ohms ~ juice, xlab="Apparent juice content (%)", ylab="Resistance (ohms)", data=fruitohms)
lines(lowess(fruitohms$juice, fruitohms$ohms), lwd=2)
pause()

require(splines)
attach(fruitohms)
plot(ohms ~ juice, cex=0.8, xlab="Apparent juice content (%)", ylab="Resistance (ohms)", type="n")
fruit.lmb4 <- lm(ohms ~ bs(juice,4))
ord <- order(juice)
lines(juice[ord], fitted(fruit.lmb4)[ord], lwd=2)
cl <- predict(fruit.lmb4, interval="confidence")
lines(juice[ord], cl[ord,"lwr"])
lines(juice[ord], cl[ord,"upr"])
```
The geophones data frame has 56 rows and 2 columns. Thickness of a layer of Alberta substratum as measured by a line of geophones.

This data frame contains the following columns:

- **distance**: location of geophone.
- **thickness**: time for signal to pass through substratum.

**Examples**

```r
plot(geophones)
lines(lowess(geophones, f=.25))
```

This function streamlines graphical output to the screen, pdf or ps files.

**Arguments**

- **width**
- **height**
- **color**: TRUE if plot is not black on white only
- **trellis**: TRUE if plot uses trellis graphics
- **device**: screen "", pdf or ps
- **path**: external file name
- **pointsize**: c(8,4)
- **horiz**: FALSE for landscape mode
**Value**

Graphical output to screen, pdf or ps file.

**Author(s)**

J.H. Maindonald

**See Also**

`postscript`

---

**head.injury**

*Minor Head Injury (Simulated) Data*

**Description**

The `head.injury` data frame has 3121 rows and 11 columns. The data were simulated according to a simple logistic regression model to match roughly the clinical characteristics of a sample of individuals who suffered minor head injuries.

**Usage**

`head.injury`

**Format**

This data frame contains the following columns:

- `age.65` age factor (0 = under 65, 1 = over 65).
- `amnesia.before` amnesia before impact (less than 30 minutes = 0, more than 30 minutes = 1).
- `basal.skull.fracture` (0 = no fracture, 1 = fracture).
- `GCS.decrease` Glasgow Coma Scale decrease (0 = no deterioration, 1 = deterioration).
- `GCS.13` initial Glasgow Coma Scale (0 = not ‘13’, 1 = ‘13’).
- `GCS.15.2hours` Glasgow Coma Scale after 2 hours (0 = not ‘15’, 1 = ‘15’).
- `high.risk` assessed by clinician as high risk for neurological intervention (0 = not high risk, 1 = high risk).
- `loss.of.consciousness` (0 = conscious, 1 = loss of consciousness).
- `open.skull.fracture` (0 = no fracture, 1 = fracture)
- `vomiting` (0 = no vomiting, 1 = vomiting)
- `clinically.important.brain.injury` any acute brain finding revealed on CT (0 = not present, 1 = present).

**References**

Description

The head Injury data frame has 3121 rows and 11 columns. The data were simulated according to a simple logistic regression model to match roughly the clinical characteristics of a sample of individuals who suffered minor head injuries.

Usage

headInjury

Format

This data frame contains the following columns:

- age.65 age factor (0 = under 65, 1 = over 65).
- amnesia.before amnesia before impact (less than 30 minutes = 0, more than 30 minutes = 1).
- basal.skull.fracture (0 = no fracture, 1 = fracture).
- GCS.decrease Glasgow Coma Scale decrease (0 = no deterioration, 1 = deterioration).
- GCS.13 initial Glasgow Coma Scale (0 = not ‘13’, 1 = ‘13’).
- GCS.15.2hours Glasgow Coma Scale after 2 hours (0 = not ‘15’, 1 = ’15’).
- high.risk assessed by clinician as high risk for neurological intervention (0 = not high risk, 1 = high risk).
- loss.of.consciousness (0 = conscious, 1 = loss of consciousness).
- open.skull.fracture (0 = no fracture, 1 = fracture)
- vomiting (0 = no vomiting, 1 = vomiting)
- clinically.important.brain.injury any acute brain finding revealed on CT (0 = not present, 1 = present).

References

Scottish Hill Races Data

Description
The record times in 1984 for 35 Scottish hill races.

Usage
hills

Format
This data frame contains the following columns:

- **dist** distance, in miles (on the map)
- **climb** total height gained during the route, in feet
- **time** record time in hours

Source
Also, in MASS library, with time in minutes.

References
A.C. Atkinson (1988) Transformations unmasked. Technometrics 30, 311-318. [ "corrects" the time for Knock Hill from 78.65 to 18.65. It is unclear if this based on the original records.]

Examples

```r
print("Transformation - Example 6.4.3")
pairs(hills, labels=c("dist\n(miles)", "climb\n(feet)", "time\n(hours)"))
pause()

pairs(log(hills), labels=c("dist\n(log(miles))", "climb\n(log(feet))", "time\n(log(hours))"))
pause()

hills0.loglm <- lm(log(time) ~ log(dist) + log(climb), data = hills)
oldpar <- par(mfrow=c(2,2))
plot(hills0.loglm)
pause()

hills.loglm <- lm(log(time) ~ log(dist)+log(climb)+log(dist):log(climb), data=hills[-18,])
summary(hills.loglm)
plot(hills.loglm)
pause()

hills2.loglm <- lm(log(time) ~ log(dist)+log(climb)+log(dist):log(climb),
data=hills[-18,])
```
```r
anova(hills.loglm, hills2.loglm)
pause()

step(hills2.loglm)
pause()

summary(hills.loglm, corr=TRUE)$coef
pause()

summary(hills2.loglm, corr=TRUE)$coef
par(oldpar)
pause()

print("Nonlinear - Example 6.9.4")
hills.nls0 <- nls(time ~ (dist^alpha) *(climb^beta), start =
c(alpha = .909, beta = .260), data = hills[-18,])
summary(hills.nls0)
plot(residuals(hills.nls0) ~ predict(hills.nls0)) # residual plot
pause()

hills$climb.mi <- hills$climb/5280
hills.nls <- nls(time ~ alpha + beta*dist + gamma*(climb.mi^delta),
               start=c(alpha = 1, beta = 1, gamma = 1, delta = 1), data=hills[-18,])
summary(hills.nls)
plot(residuals(hills.nls) ~ predict(hills.nls)) # residual plot
```

**hills2000**

**Scottish Hill Races Data - 2000**

**Description**

The record times in 2000 for 77 Scottish hill races. We believe the data are, for the most part, trustworthy. However, the `dist` variable for Caerketon (record 58) seems to have been variously recorded as 1.5 mi and 2.5 mi.

**Usage**

hills2000

**Format**

This data frame contains the following columns:

- **h** male record time in hours
- **m** plus minutes
- **s** plus seconds
- **h0** female record time in hours
- **m0** plus minutes
- **s0** plus seconds
- **dist** distance, in miles (on the map)
- **climb** total height gained during the route, in feet
- **time** record time in hours
- **timef** record time in hours for females
houseprices

Source

The Scottish Running Resource, http://www.hillrunning.co.uk

Examples

pairs(hills2000)

houseprices Aranda House Prices

Description

The houseprices data frame consists of the floor area, price, and the number of bedrooms for a sample of houses sold in Aranda in 1999. Aranda is a suburb of Canberra, Australia.

Usage

houseprices

Format

This data frame contains the following columns:

area  a numeric vector giving the floor area
bedrooms a numeric vector giving the number of bedrooms
sale.price a numeric vector giving the sale price in thousands of Australian dollars

Source

J.H. Maindonald

Examples

plot(sale.price~area, data=houseprices)
pause()

coplot(sale.price~area|bedrooms, data=houseprices)
pause()

print("Cross-Validation - Example 5.5.2")

houseprices.lm <- lm(sale.price ~ area, data=houseprices)
summary(houseprices.lm)$sigma^2

pause()

cv.lm()
pause()

print("Bootstrapping - Example 5.5.3")

houseprices.fn <- function (houseprices, index){
  house.resample <- houseprices[index,]
  house.lm <- lm(sale.price ~ area, data=house.resample)
  coef(house.lm)[2]  # slope estimate for resampled data
require(boot)  # ensure that the boot package is loaded
houseprices.boot <- boot(houseprices, R=999, statistic=houseprices.fn)

houseprices1.fn <- function (houseprices, index){
  house.resample <- houseprices[index,]
  house.lm <- lm(sale.price ~ area, data=house.resample)
  predict(house.lm, newdata=data.frame(area=1200))
}

houseprices1.boot <- boot(houseprices, R=999, statistic=houseprices1.fn)
boot.ci(houseprices1.boot, type="perc")  # "basic" is an alternative to "perc"

houseprices2.fn <- function (houseprices, index){
  house.resample <- houseprices[index,]
  house.lm <- lm(sale.price ~ area, data=house.resample)
  houseprices$sale.price-predict(house.lm, houseprices)  # resampled prediction errors
}

n <- length(houseprices$area)
R <- 200
houseprices2.boot <- boot(houseprices, R=R, statistic=houseprices2.fn)
house.fac <- factor(rep(1:n, rep(R, n)))
plot(house.fac, as.vector(houseprices2.boot$t), ylab="Prediction Errors", xlab="House")
pause()
plot(apply(houseprices2.boot$t,2, sd)/predict.lm(houseprices.lm, se.fit=TRUE)$se.fit, ylab="Ratio of Bootstrap SE's to Model-Based SE's", xlab="House", pch=16)
abline(1,0)

humanpower

### humanpower

**Oxygen uptake versus mechanical power, for humans**

### Description

The data set from Daedalus project.

### Usage

data(humanpower1)

### Format

A data frame with 28 observations on the following 3 variables.

- **wattsPerKg** a numeric vector: watts per kilogram of body weight
- **o2** a numeric vector: ml/min/kg
- **id** a factor with levels 1 - 5 (humanpower1) or 1 - 4 (humanpower2), identifying the different athletes
Details

Data in humanpower1 are from investigations (Bussolari 1987) designed to assess the feasibility of a proposed 119 kilometer human powered flight from the island of Crete – in the initial phase of the Daedalus project. Data are for five athletes – a female hockey player, a male amateur tri-athlete, a female amateur triathlete, a male wrestler and a male cyclist – who were selected from volunteers who were recruited through the news media. Data in humanpower2 are for four out of the 25 applicants who were selected for further testing, in the lead-up to the eventual selection of a pilot for the Daedalus project (Nadel and Bussolari 1988).

Source


References


Examples

str(humanpower1)
plot(humanpower1)
lm(o2 ~ id + wattsPerKg:id, data=humanpower1)
lm(o2 ~ id + wattsPerKg:id, data=humanpower2)

ironslag

Iron Content Measurements

Description

The ironslag data frame has 53 rows and 2 columns. Two methods for measuring the iron content in samples of slag were compared, a chemical and a magnetic method. The chemical method requires greater effort than the magnetic method.

Usage

ironslag

Format

This data frame contains the following columns:

chemical a numeric vector containing the measurements coming from the chemical method
magnetic a numeric vector containing the measurement coming from the magnetic method
Source

Examples
iron.lm <- lm(chemical ~ magnetic, data = ironslag)
oldpar <- par(mfrow = c(2,2))
plot(iron.lm)
par(oldpar)

jobs

---

Canadian Labour Force Summary Data (1995-96)

Description
The number of workers in the Canadian labour force broken down by region (BC, Alberta, Prairies, Ontario, Quebec, Atlantic) for the 24-month period from January, 1995 to December, 1996 (a time when Canada was emerging from a deep economic recession).

Usage
jobs

Format
This data frame contains the following columns:

- **BC**  monthly labour force counts in British Columbia
- **Alberta**  monthly labour force counts in Alberta
- **Prairies**  monthly labour force counts in Saskatchewan and Manitoba
- **Ontario**  monthly labour force counts in Ontario
- **Quebec**  monthly labour force counts in Quebec
- **Atlantic**  monthly labour force counts in Newfoundland, Nova Scotia, Prince Edward Island and New Brunswick
- **Date**  year (in decimal form)

Details
These data have been seasonally adjusted.

Source
Statistics Canada
Examples

print("Multiple Variables and Times - Example 2.1.4")
sapply(jobs, range)
pause()

matplot(jobs[,7], jobs[,7], type="l", xlim=c(95,97.1))
# Notice that we have been able to use a data frame as the second argument to matplot().
# For more information on matplot(), type help(matplot)
text(rep(jobs[24,7], 6), jobs[24,1:6], names(jobs)[1:6], adj=0)
pause()

sapply(log(jobs[,,-7]), range)
apply(sapply(log(jobs[,,-7]), range), 2, diff)
pause()

oldpar <- par(mfrow=c(2,3))
range.log <- sapply(log(jobs[,,-7]), 2, range)
maxdiff <- max(apply(range.log, 2, diff))
range.log[2,] <- range.log[1,] + maxdiff
titles <- c("BC Jobs","Alberta Jobs","Prairie Jobs",
"Ontario Jobs", "Quebec Jobs", "Atlantic Jobs")
for (i in 1:6){
plot(jobs$Date, log(jobs[,i], 2), type = "l", ylim = range.log[,i],
    xlab = "Time", ylab = "Number of jobs", main = titles[i])
}
par(oldpar)

kiwishade

Kiwi Shading Data

Description

The kiwishade data frame has 48 rows and 4 columns. The data are from a designed experiment that compared different kiwifruit shading treatments. There are four vines in each plot, and four plots (one for each of four treatments: none, Aug2Dec, Dec2Feb, and Feb2May) in each of three blocks (locations: west, north, east). Each plot has the same number of vines, each block has the same number of plots, with each treatment occurring the same number of times.

Usage

kiwishade

Format

This data frame contains the following columns:

yield  Total yield (in kg)
plot  a factor with levels east.Aug2Dec, east.Dec2Feb, east.Feb2May, east.none,
    west.Dec2Feb, west.Feb2May, west.none
block  a factor indicating the location of the plot with levels east, north, west
shade  a factor representing the period for which the experimenter placed shading over the vines; with levels: none no shading, Aug2Dec August - December, Dec2Feb December - February, Feb2May February - May
Details

The northernmost plots were grouped together because they were similarly affected by shading from the sun in the north. For the remaining two blocks shelter effects, whether from the west or from the east, were thought more important.

Source


References


Examples

```r
print("Data Summary - Example 2.2.1")
attach(kiwishade)
kiwimeans <- aggregate(yield, by=list(block, shade), mean)
names(kiwimeans) <- c("block","shade","meanyield")

kiwimeans[1:4,]
pause()

print("Multilevel Design - Example 9.3")
kiwishade.aov <- aov(yield ~ shade+Error(block/shade),data=kiwishade)
summary(kiwishade.aov)
pause()

sapply(split(yield, shade), mean)
pause()

kiwi.table <- t(sapply(split(yield, plot), as.vector))
kiwi.means <- sapply(split(yield, plot), mean)
kiwi.means.table <- matrix(rep(kiwi.means,4), nrow=12, ncol=4)
kiwi.summary <- data.frame(kiwi.means, kiwi.table=kiwi.means.table)
names(kiwi.summary)<- c("Mean", "Vine 1", "Vine 2", "Vine 3", "Vine 4")
kiwi.summary

mean(kiwi.means) # the grand mean (only for balanced design)

require(nlme)
kiwishade.lme <- lme(fixed = yield ~ shade, random = ~ 1 | block/plot, data=kiwishade)
res <- residuals(kiwishade.lme) # By default fitted(kiwishade.lme, level=2)
coplot(res ~ hat | kiwishade$block, pch=16, columns=3, xlab= "Fitted", ylab="Residuals")

res <- residuals(kiwishade.lme, level=0) # shade effects only
unique(hat) # There are just four distinct values, one per treatment
coplot(res ~ hat | kiwishade$block, pch=16, columns=3, xlab="Fitted", ylab="Residuals")
```
n.omit <- 2
take <- rep(TRUE, 48)
take[sample(1:48, 2)] <- FALSE
kiwishade.lme <- lme(yield ~ shade, data = kiwishade,
                   random = ~1 | block/plot, subset=take)
VarCorr(kiwishade.lme)[4, 1] # Plot component of variance
VarCorr(kiwishade.lme)[4, 1] # Vine component of variance
detach(kiwishade)

---

leafshape

**Full Leaf Shape Data Set**

**Description**

Leaf length, width and petiole measurements taken at various sites in Australia.

**Usage**

leafshape

**Format**

This data frame contains the following columns:

- **bladelen** leaf length (in mm)
- **petiole** a numeric vector
- **bladewid** leaf width (in mm)
- **latitude** latitude
- **logwid** natural logarithm of width
- **logpet** logarithm of petiole
- **loglen** logarithm of length
- **arch** leaf architecture (0 = plagiotropic, 1 = orthotropic)
- **location** a factor with levels Sabah, Panama, Costa Rica, N Queensland, S Queensland, Tasmania

**Source**

leafshape17

Subset of Leaf Shape Data Set

Description

The leafshape17 data frame has 61 rows and 8 columns. These are leaf length, width and petiole measurements taken at several sites in Australia. This is a subset of the leafshape data frame.

Usage

leafshape17

Format

This data frame contains the following columns:

- **bladelen**: leaf length (in mm)
- **petiole**: a numeric vector
- **bladewid**: leaf width (in mm)
- **latitude**: latitude
- **logwid**: natural logarithm of width
- **logpet**: logarithm of petiole measurement
- **loglen**: logarithm of length
- **arch**: leaf architecture (0 = orthotropic, 1 = plagiotropic)

Source


Examples

```r
print("Discriminant Analysis - Example 11.2")
require(MASS)
leaf17.lda <- lda(arch ~ logwid+loglen, data=leafshape17)
leaf17.hat <- predict(leaf17.lda)
leaf17.lda
  table(leafshape17$arch, leaf17.hat$class)
pause()

  tab <- table(leafshape17$arch, leaf17.hat$class)
  sum(tab[row(tab)==col(tab)])/sum(tab)
leaf17cv.lda <- lda(arch ~ logwid+loglen, data=leafshape17, CV=TRUE)
  tab <- table(leafshape17$arch, leaf17cv.lda$class)
pause()

  leaf17(glm <- glm(arch ~ logwid + loglen, family=binomial, data=leafshape17)
  options(digits=3)
  summary(leaf17.glm)$coef
```
leaf17.one <- cv.binary(leaf17.glm)
table(leafshape17$arch, round(leaf17.one$internal)) # Resubstitution
pause()
table(leafshape17$arch, round(leaf17.one$cv)) # Cross-validation

### leaftemp

**Leaf and Air Temperature Data**

**Description**

These data consist of measurements of vapour pressure and of the difference between leaf and air temperature.

**Usage**

```r
data(leaf-temp)
```

**Format**

This data frame contains the following columns:

- **CO2level** Carbon Dioxide level: *low, medium, high*
- **vapPress** Vapour pressure
- **tempDiff** Difference between leaf and air temperature
- **BtempDiff** a numeric vector

**Source**

Katharina Siebke and Susan von Cammerer, Australian National University.

**Examples**

```r
print("Fitting Multiple Lines - Example 7.3")
leaf.lm1 <- lm(tempDiff ~ 1 , data = leaftemp)
leaf.lm2 <- lm(tempDiff ~ vapPress, data = leaftemp)
leaf.lm3 <- lm(tempDiff ~ CO2level + vapPress, data = leaftemp)
leaf.lm4 <- lm(tempDiff ~ CO2level + vapPress + vapPress:CO2level, 
               data = leaftemp)
anova(leaf.lm1, leaf.lm2, leaf.lm3, leaf.lm4)
summary(leaf.lm2)
plot(leaf.lm2)
```
**leaftemp.all**  
*Full Leaf and Air Temperature Data Set*

**Description**

The leaftemp.all data frame has 62 rows and 9 columns.

**Usage**

leaftemp.all

**Format**

This data frame contains the following columns:

- **glasshouse**  a factor with levels A, B, C
- **CO2level**  a factor with Carbon Dioxide Levels: high, low, medium
- **day**  a factor
- **light**  a numeric vector
- **CO2**  a numeric vector
- **tempDiff**  Difference between Leaf and Air Temperature
- **BtempDiff**  a numeric vector
- **airTemp**  Air Temperature
- **vapPress**  Vapour Pressure

**Source**

J.H. Maindonald

---

**litters**  
*Mouse Litters*

**Description**

Data on the body and brain weights of 20 mice, together with the size of the litter. Two mice were taken from each litter size.

**Usage**

litters

**Format**

This data frame contains the following columns:

- **lsize**  litter size
- **bodywt**  body weight
- **brainwt**  brain weight
Source


Examples

print("Multiple Regression - Example 6.2")

pairs(litters, labels=c("lsize\n(litter size)", "bodywt\n(Body Weight)", "brainwt\n(Brain Weight)") # pairs(litters) gives a scatterplot matrix with less adequate labeling

mice1.lm <- lm(brainwt ~ lsize, data = litters) # Regress on lsize
mice2.lm <- lm(brainwt ~ bodywt, data = litters) # Regress on bodywt
mice12.lm <- lm(brainwt ~ lsize + bodywt, data = litters) # Regress on lsize & bodywt

summary(mice1.lm)$coef # Similarly for other coefficients. 
# results are consistent with the biological concept of brain sparing

dropout()

hat(model.matrix(mice12.lm)) # hat diagonal

dropout()

plot(lm.influence(mice12.lm)$hat, residuals(mice12.lm))

print("Diagnostics - Example 6.3")

mice12.lm <- lm(brainwt ~ bodywt+lsize, data=litters)
oldpar <-par(mfrow = c(1,2))
bx <- mice12.lm$coef[2]; bz <- mice12.lm$coef[3]
res <- residuals(mice12.lm)
plot(litters$bodywt, bx*litters$bodywt+res, xlab="Body weight", ylab="Component + Residual")
panel.smooth(litters$bodywt, bx*litters$bodywt+res) # Overlay
plot(litters$lsize, bz*litters$lsize+res, xlab="Litter size", ylab="Component + Residual")
panel.smooth(litters$lsize, bz*litters$lsize+res)
par(oldpar)

logisticsim

Simple Logistic Regression Data Simulator

Description

This function simulates simple regression data from a logistic model.

Usage

logisticsim(x = seq(0, 1, length=101), a = 2, b = -4, seed=NULL)
measles

Arguments

- x: a numeric vector representing the explanatory variable
- a: the regression function intercept
- b: the regression function slope
- seed: numeric constant

Value

- a list consisting of
  - x: the explanatory variable vector
  - y: the Poisson response vector

Examples

logisticsim()

---

measles  

Deaths in London from measles

Description

Deaths in London from measles: 1629 – 1939, with gaps.

Usage

data(measles)

Format

The format is: Time-Series [1:311] from 1629 to 1939: 42 2 3 80 21 33 27 12 NA NA ...

Source


References

Description

The mifem data frame has 1295 rows and 10 columns.

Usage

mifem

Format

This data frame contains the following columns:

- **outcome**: mortality outcome, a factor with levels *live*, *dead*
- **age**: age at onset
- **yronset**: year of onset
- **premi**: previous myocardial infarction event, a factor with levels *y*, *n*, *nk not known*
- **smstat**: smoking status, a factor with levels *c current*, *x ex-smoker*, *n non-smoker*, *nk not known*
- **diabetes**: a factor with levels *y*, *n*, *nk not known*
- **highbp**: high blood pressure, a factor with levels *y*, *n*, *nk not known*
- **hichol**: high cholesterol, a factor with levels *y*, *n*, *nk not known*
- **angina**: a factor with levels *y*, *n*, *nk not known*
- **stroke**: a factor with levels *y*, *n*, *nk not known*

Source

Newcastle (Australia) centre of the Monica project; see the web site http://www.ktl.fi/monicaindex.html

Examples

```r
print("CART - Example 10.7")
summary(mifem)
pause()

require(rpart)
mifem.rpart <- rpart(outcome ~ ., data = mifem, cp = 0.0025)
plotcp(mifem.rpart)
printcp(mifem.rpart)
pause()

mifemb.rpart <- prune(mifem.rpart, cp=0.006)
print(mifemb.rpart)
```
**mignonette**  
*Darwin’s Wild Mignonette Data*

**Description**

Data which compare the heights of crossed plants with self-fertilized plants. Plants were paired within the pots in which they were grown, with one on one side and one on the other.

**Usage**

mignonette

**Format**

This data frame contains the following columns:

- **cross** heights of the crossed plants
- **self** heights of the self-fertilized plants

**Source**


**Examples**

```r
print("Is Pairing Helpful? - Example 4.3.1")
attach(mignonette)
plot(cross ~ self, pch=rep(c(4,1), c(3,12))); abline(0,1)
abline(mean(cross-self), 1, lty=2)
detach(mignonette)
```

---

**milk**  
*Milk Sweetness Study*

**Description**

The milk data frame has 17 rows and 2 columns. Each of 17 panelists compared two milk samples for sweetness.

**Usage**

milk

**Format**

This data frame contains the following columns:

- **four** a numeric vector consisting of the assessments for four units of additive
- **one** a numeric vector while the is the assessment for one unit of additive
modelcars

Source
??

References
??

Examples

```r
print("Rug Plot - Example 1.8.1")
xrange <- range(milk)
plot(four ~ one, data = milk, xlim = xrange, ylim = xrange, pch = 16)
rug(milk$one)
rug(milk$four, side = 2)
abline(0, 1)
```

---

### modelcars

#### Model Car Data

**Description**

The `modelcars` data frame has 12 rows and 2 columns. The data are for an experiment in which a model car was released three times at each of four different distances up a 20 degree ramp. The experimenter recorded distances traveled from the bottom of the ramp across a concrete floor.

**Usage**

`modelcars`

**Format**

This data frame contains the following columns:

- `distance.traveled` a numeric vector consisting of the lengths traveled (in cm)
- `starting.point` a numeric vector consisting of the distance of the starting point from the top of the ramp (in cm)

**Source**

J.H. Maindonald

**Examples**

```r
plot(modelcars)
modelcars.lm <- lm(distance.traveled ~ starting.point, data=modelcars)
aov(modelcars.lm)
pause()

print("Response Curves - Example 4.6")
attach(modelcars)
stripchart(distance.traveled ~ starting.point, vertical=TRUE, pch=15, xlab = "Distance up ramp")
detach(modelcars)
```
WHO Monica Data

Description

The monica data frame has 6357 rows and 12 columns. Note that mifem is the female subset of this data frame.

Usage

monica

Format

This data frame contains the following columns:

- **outcome**: mortality outcome, a factor with levels live, dead
- **age**: age at onset
- **sex**: m = male, f = female
- **hosp**: y = hospitalized, n = not hospitalized
- **yronset**: year of onset
- **premi**: previous myocardial infarction event, a factor with levels y, n, nk not known
- **smstat**: smoking status, a factor with levels c current, x ex-smoker, n non-smoker, nk not known
- **diabetes**: a factor with levels y, n, nk not known
- **highbp**: high blood pressure, a factor with levels y, n, nk not known
- **hichol**: high cholesterol, a factor with levels y, n nk not known
- **angina**: a factor with levels y, n, nk not known
- **stroke**: a factor with levels y, n, nk not known

Source

Newcastle (Australia) centre of the Monica project; see the web site http://www.ktl.fi

Examples

print("CART - Example 10.7")
summary(monica)
pause()

require(rpart)
monica.rpart <- rpart(outcome ~ ., data = monica, cp = 0.0025)
plotcp(monica.rpart)
printcp(monica.rpart)
pause()

monicab.rpart <- prune(monica.rpart, cp=0.006)
print(monicab.rpart)
motes       Moths Data

Description

The `motes` data frame has 41 rows and 4 columns. These data are from a study of the effect of habitat on the densities of two species of moth (A and P). Transects were set across the search area. Within transects, sections were identified according to habitat type.

Usage

`motes`

Format

This data frame contains the following columns:

- `meters` length of transect
- `A` number of type A moths found
- `P` number of type P moths found
- `habitat` a factor with levels `Bank`, `Disturbed`, `Lowerside`, `NEsoak`, `NWsoak`, `SEsoak`, `SWsoak`, `Upperside`

Source

Sharyn Wragg, formerly of Australian National University

Examples

```r
print("Quasi Poisson Regression - Example 8.3")
rbind(table(motes[,4]), sapply(split(motes[-4], motes$habitat), apply,2,
   sum))
A.glm <- glm(formula = A ~ log(meters) + factor(habitat), family =
quasipoisson, data = motes)
summary(A.glm)
moths$habitat <- relevel(moths$habitat, ref="Lowerside")
A.glm <- glm(A ~ habitat + log(meters), family=quasipoisson, data=moths)
summary(A.glm)$coef
```

multilap       Data Filtering Function

Description

A subset of data is selected for which the treatment to control ratio of non-binary covariates is never outside a specified range.

Usage

```r
multilap(df=nsw74psid1, maxf=20, colnames=c("educ", "age", "re74", "re75",
   "re78"))
```
Arguments

- df: a data frame
- maxf: filtering parameter
- colnames: columns to be compared for filtering

Author(s)

J.H. Maindonald

Description

This data frame contains 445 rows and 10 columns. These data are pertinent to an investigation of the way that earnings changed, between 1974-1975 and 1978, in the absence of training. Data for the experimental treatment group (NSW) were combined with control data results from the Panel Study of Income Dynamics (PSID) study.

Usage

nsw74demo

Format

This data frame contains the following columns:

- trt: a numeric vector identifying the study in which the subjects were enrolled (0 = PSID, 1 = NSW).
- age: age (in years).
- educ: years of education.
- black: (0 = not black, 1 = black).
- hisp: (0 = not hispanic, 1 = hispanic).
- marr: (0 = not married, 1 = married).
- nodeg: (0 = completed high school, 1 = dropout).
- re75: real earnings in 1975.
- re78: real earnings in 1978.

Source

http://www.columbia.edu/ rd247/nswdata.html

References

Labour Training Evaluation Data

Description

This data frame contains 2675 rows and 10 columns. These data are pertinent to an investigation of the way that earnings changed, between 1974-1975 and 1978, in the absence of training. Data for the experimental treatment group (NSW) were combined with control data results from the Panel Study of Income Dynamics (PSID) study.

Usage

nsw74psid1

Format

This data frame contains the following columns:

- **trt**: a numeric vector identifying the study in which the subjects were enrolled (0 = PSID, 1 = NSW).
- **age**: age (in years).
- **educ**: years of education.
- **black**: (0 = not black, 1 = black).
- **hisp**: (0 = not hispanic, 1 = hispanic).
- **marr**: (0 = not married, 1 = married).
- **nodeg**: (0 = completed high school, 1 = dropout).
- **re74**: real earnings in 1974.
- **re75**: real earnings in 1975.
- **re78**: real earnings in 1978.

Source

http://www.columbia.edu/ rd247/nswdata.html

References


Examples

```r
print("Interpretation of Regression Coefficients - Example 6.6")
nsw74psid1.lm <- lm(re78 ~ trt + (age + educ + re74 + re75) +
(black + hisp + marr + nodeg), data = nsw74psid1)
summary(nsw74psid1.lm)$coef
options(digits=4)
sapply(nsw74psid1[, c(2,3,8,9,10)], quantile, prob=c(.25,.5,.75,.95,1))
```
```
attach(nsw74psid1)
sapply(nsw74psid1[trt==1, c(2,3,8,9,10)], quantile,
prob=c(.25,.5,.75,.95,1))
pause()

here <- age <= 40 & re74<=5000 & re75 <= 5000 & re78 < 30000
nsw74psidA <- nsw74psid1[here, ]
detach(nsw74psid1)
table(nsw74psidA$trt)
pause()

A1.lm <- lm(re78 ~ trt + (age + educ + re74 + re75) + (black +
hisp + marr + nodeg), data = nsw74psidA)
summary(A1.lm)$coef
pause()

A2.lm <- lm(re78 ~ trt + (age + educ + re74 + re75) * (black +
hisp + marr + nodeg), data = nsw74psidA)
anova(A1.lm, A2.lm)
```

---

**nsw74psid3**  
*Labour Training Evaluation Data*

**Description**

These data are pertinent to an investigation of the way that earnings changed, between 1974-1975 and 1978, in the absence of training. The data frame combines data for the experimental treatment group (NSW, 185 observations), using as control data results from the PSID (Panel Study of Income Dynamics) study (128 observations). The latter were chosen to mimic the characteristics of the NSW training and control groups. These are a subset of the *nsw74psid1* data.

**Usage**

`nsw74psid3`

**Format**

This data frame contains the following columns:

- **trt** a numeric vector identifying the study in which the subjects were enrolled (0 = PSID, 1 = NSW)
- **age** age (in years)
- **educ** years of education
- **black** (0 = not black, 1 = black)
- **hisp** (0 = not hispanic, 1 = hispanic)
- **marr** (0 = not married, 1 = married)
- **nodeg** (0 = completed high school, 1 = dropout)
- **re74** real earnings in 1974
- **re75** real earnings in 1975
- **re78** real earnings in 1978
Source

http://www.columbia.edu/ rd247/nswdata.html

References


Examples

print("Contingency Tables - Example 4.4")
table(nsw74psid3$trt, nsw74psid3$nodeg)
chisq.test(table(nsw74psid3$trt,nsw74psid3$nodeg))

nsw74psidA  A Subset of the nsw74psid1 Data Set

Description

The nsw74psidA data frame has 252 rows and 10 columns. See nsw74psid1 for more information.

Usage

nsw74psidA

Format

This data frame contains the following columns:

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>trt</td>
<td>numeric</td>
</tr>
<tr>
<td>age</td>
<td>numeric</td>
</tr>
<tr>
<td>educ</td>
<td>numeric</td>
</tr>
<tr>
<td>black</td>
<td>numeric</td>
</tr>
<tr>
<td>hisp</td>
<td>numeric</td>
</tr>
<tr>
<td>marr</td>
<td>numeric</td>
</tr>
<tr>
<td>nodeg</td>
<td>numeric</td>
</tr>
<tr>
<td>re74</td>
<td>numeric</td>
</tr>
<tr>
<td>re75</td>
<td>numeric</td>
</tr>
<tr>
<td>re78</td>
<td>numeric</td>
</tr>
</tbody>
</table>

Details

This data set was obtained using:

here <- age <= 40 & re74<=5000 & re75 <= 5000 & re78 < 30000nsw74psidA
<- nsw74psid1[here, ]
Examples

```r
table(nsw74psidA$trt)
A1.lm <- lm(re78 ~ trt + (age + educ + re74 + re75) + (black + hisp + marr + nodeg), data = nsw74psidA)
summary(A1.lm)$coef

discA.glm <- glm(formula = trt ~ age + educ + black + hisp + marr + nodeg + re74 + re75, family = binomial, data = nsw74psidA)
A.scores <- predict(discA.glm)
options(digits=4)
overlap <- A.scores > -3.5 & A.scores < 3.8
A.lm <- lm(re78 ~ trt + A.scores, data=nsw74psidA, subset = overlap)
summary(A.lm)$coef
```

obounce  

*Bounce - obsolete*

**Description**

A utility function for oneway.plot

**Author(s)**

J.H. Maindonald

oddbooks  

*Measurements on 12 books*

**Description**

Data giving thickness (mm), height (cm), width (cm) and weight (g), of 12 books. Books were selected so that thickness decreased as page area increased

**Usage**

```r
data(oddbooks)
```

**Format**

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

- **thick** a numeric vector
- **height** a numeric vector
- **breadth** a numeric vector
- **weight** a numeric vector
Details

Source

JM took books from his library.

Examples
data(oddbooks)
str(oddbooks)
plot(oddbooks)

onesamp  Paired Sample t-test

Description

This function performs a t-test for the mean difference for paired data, and produces a scatterplot of one column against the other column, showing whether there was any benefit to using the paired design.

Usage

onesamp(dset=corn, x="unsprayed", y="sprayed", xlab=NULL, ylab=NULL, dubious=NULL, conv=NULL, dig=2)

Arguments

dset a matrix or dataframe having two columns
x name of column to play the role of the ‘predictor’
y name of column to play the role of the ‘response’
xlab horizontal axis label
ylab vertical axis label
dubious
conv
dig

Value

A scatterplot of y against x together with estimates of standard errors and standard errors of the difference (y-x).
Also produced is a confidence interval and p-value for the test.

Author(s)

J.H. Maindonald

Examples

onesamp(dset=pair65, x="ambient", y="heated", xlab ="Amount of stretch (ambient)", ylab ="Amount of stretch (heated)"
**onet.permutation**  
*One Sample Permutation t-test*

**Description**

This function computes the p-value for the one sample t-test using a permutation test. The permutation density can also be plotted.

**Usage**

```r
onet.permutation(x=pair65$heated - pair65$ambient, nsim=2000, plotit=TRUE)
```

**Arguments**

- `x`: a numeric vector containing the sample values (centered at the null hypothesis value)
- `nsim`: the number of permutations (randomly selected)
- `plotit`: if TRUE, the permutation density is plotted

**Value**

The p-value for the test of the hypothesis that the mean of `x` differs from 0

**Author(s)**

J.H. Maindonald

**References**


**Examples**

```r
onet.permutation()
```

---

**onetPermutation**  
*One Sample Permutation t-test*

**Description**

This function computes the p-value for the one sample t-test using a permutation test. The permutation density can also be plotted.

**Usage**

```r
onetPermutation(x=pair65$heated - pair65$ambient, nsim=2000, plotit=TRUE)
```
oneway.plot

Arguments

x a numeric vector containing the sample values (centered at the null hypothesis value)
nsim the number of permutations (randomly selected)
plotit if TRUE, the permutation density is plotted

Value
The p-value for the test of the hypothesis that the mean of x differs from 0

Author(s)
J.H. Maindonald

References

Examples
onetPermutation()

oneway.plot Display of One Way Analysis Results

Description
A line plot of means for unstructured comparison.

Usage
oneway.plot(obj = rice.aov, axisht = 6, xlim = NULL, xlab = NULL,
lsdht = 1.5, hsdht = 0.5, textht = axisht - 2.5, oma = rep(1, 4), angle = 80, alpha = 0.05)

Arguments
obj One way analysis of variance object (from aov)
axisht Axis height
xlim Range on horizontal axis
xlab Horizontal axis label
lsdht Height adjustment parameter for LSD comparison plot
hsdht Height adjustment parameter for Tukey’s HSD comparison plot
textht Height of text
oma Outer margin area
angle Text angle (in degrees)
alpha Test size
onewayPlot

Value

A line plot

Author(s)

J.H. Maindonald

Examples

rice.aov <- aov(ShootDryMass ~ trt, data=rice)
oneway.plot(obj=rice.aov)

onewayPlot  Display of One Way Analysis Results

Description

A line plot of estimates for unstructured comparison of factor levels

Usage

onewayPlot(obj = rice.aov, trtnam = "trt", axisht = 6, xlim = NULL, xlab = NULL, lsdht = 1.5, hsdht = 0.5, textht = axisht - 2.5, oma = rep(1, 4), angle = 80, alpha = 0.05)

Arguments

obj One way analysis of variance object (from aov)
trtnam name of factor for which line plot is required
axisht Axis height
xlim Range on horizontal axis
xlab Horizontal axis label
lsdht Height adjustment parameter for display of LSD
hsdht Height adjustment parameter for display of Tukey’s HSD
textht Height of text
oma Outer margin area
angle Text angle (in degrees)
alpha Test size

Value

 Estimates, labeled with level names, are set out along a line

Author(s)

J.H. Maindonald

Examples

rice.aov <- aov(ShootDryMass ~ trt, data=rice)
onewayPlot(obj=rice.aov)
**orings**  
*Challenger O-rings Data*

**Description**

Record of the number and type of O-ring failures prior to the tragic Challenger mission in January, 1986.

**Usage**

orings

**Format**

This data frame contains the following columns:

- **Temperature**  O-ring temperature for each test firing or actual launch of the shuttle rocket engine
- **Erosion**  Number of erosion incidents
- **Blowby**  Number of blowby incidents
- **Total**  Total number of incidents

**Source**


**References**


**Examples**

oldpar <- par(mfrow=c(1,2))
plot(Total~Temperature, data = orings[c(1,2,4,11,13,18),])  
# the observations included in the pre-launch charts
plot(Total~Temperature, data = orings)
par(oldpar)

**overlap.density**  
*Overlapping Density Plots - obsolete*

**Description**

Densities for two independent samples are estimated and plotted.

**Usage**

overlap.density(x0, x1, ratio=c(0.05, 20), compare.numbers=TRUE, plotit=TRUE, gpnames=c("Control", "Treatment"), xlab="Score")
overlapDensity

Arguments

x0  control group measurements
x1  treatment group measurements
ratio  the range within which the relative numbers of observations from the two groups are required to lie. [The relative numbers at any point are estimated from \( \frac{\text{density1} \times n1}{\text{density0} \times x0} \)]

compare.numbers  If TRUE (default), then density plots are scaled to have total area equal to the sample size; otherwise total area under each density is 1
plotit  If TRUE, a plot is produced
gpnames  Names of the two samples
xlab  Label for x-axis

Author(s)

J.H. Maindonald

See Also
t.test

Examples

attach(two65)
overlap.density(ambient, heated)
t.test(ambient, heated)

overlapsDensity  Overlapping Density Plots

Description

Densities for two independent samples are estimated and plotted.

Usage

\[
\text{overlapDensity}(x0, x1, \text{ratio} = c(0.05, 20), \text{compare.numbers} = \text{FALSE}, \text{plotit} = \text{TRUE}, \text{gpnames} = \text{c("Control", "Treatment")}, \text{cutoffs} = \text{c(lower=TRUE, upper=TRUE)}, \text{bw} = \text{FALSE}, \text{xlab} = \text{"Score"}, \text{col} = 1:2, \text{lt} = 1:2)
\]

Arguments

x0  control group measurements
x1  treatment group measurements
ratio  the range within which the relative numbers of observations from the two groups are required to lie. [The relative numbers at any point are estimated from \( \frac{\text{density1} \times n1}{\text{density0} \times x0} \)]
If TRUE (default), then density plots are scaled to have total area equal to the sample size; otherwise total area under each density is 1

If TRUE, a plot is produced

Names of the two samples

logical vector, indicating whether density estimates should be truncated below (lower=TRUE) or above (upper=TRUE)

logical, indicates whether to overwrite with a gray scale plot

Label for x-axis

standard color parameter

standard line type preference

J.H. Maindonald

See Also

t.test

Examples

attach(two65)
overlapDensity(ambient,heated)
t.test(ambient,heated)

Description

Monthly provisional mean total ozone (in Dobson units) at Halley Bay (approximately corrected to Bass-Paur).

Usage

ozone

Format

This data frame contains the following columns:

Year  the year
Aug  August mean total ozone
Sep  September mean total ozone
Oct  October mean total ozone
Nov  November mean total ozone
Dec  December mean total ozone
Jan  January mean total ozone
**pair65**

- **Feb** February mean total ozone
- **Mar** March mean total ozone
- **Apr** April mean total ozone
- **Annual** Yearly mean total ozone

**Source**

http://www.antarctica.ac.uk/met/jds/ozone/data/zoz5699.dat

**References**


**Examples**

```
AnnualOzone <- ts(ozone$Annual, start=1956)
plot(AnnualOzone)
```

---

**pair65**  
*Heated Elastic Bands*

**Description**

The *pair65* data frame has 9 rows and 2 columns. Eighteen elastic bands were divided into nine pairs, with bands of similar stretchiness placed in the same pair. One member of each pair was placed in hot water (60-65 degrees C) for four minutes, while the other was left at ambient temperature. After a wait of about ten minutes, the amounts of stretch, under a 1.35 kg weight, were recorded.

**Usage**

```r
pair65
```

**Format**

This data frame contains the following columns:

- **heated** a numeric vector giving the stretch lengths for the heated bands
- **ambient** a numeric vector giving the stretch lengths for the unheated bands

**Source**

J.H. Maindonald

**Examples**

```r
mean(pair65$heated - pair65$ambient)
sd(pair65$heated - pair65$ambient)
```
panel.corr  

Scatterplot Panel

Description

This function produces a bivariate scatterplot with the Pearson correlation. This is for use with the function `panelplot`.

Usage

panel.corr(data, ...)

Arguments

data  A data frame with columns x and y
...  Additional arguments

Author(s)

J.H. Maindonald

Examples

# correlation between body and brain weights for 20 mice:

weights <- litters[,-1]
names(weights) <- c("x","y")
weights <- list(weights)
weights[[1]]$xlim <- range(litters[,2])
weights[[1]]$ylim <- range(litters[,3])
panelplot(weights, panel.corr, totrows=1, totcols=1)

panelCorr  

Scatterplot Panel

Description

This function produces a bivariate scatterplot with the Pearson correlation. This is for use with the function `panelplot`.

Usage

panelCorr(data, ...)

Arguments

data  A data frame with columns x and y
...  Additional arguments
Author(s)
J.H. Maindonald

Examples

# correlation between body and brain weights for 20 mice:
weights <- litters[, -1]
names(weights) <- c("x", "y")
weights <- list(weights)
weights[[1]]$xlim <- range(litters[, 2])
weights[[1]]$ylim <- range(litters[, 3])
panelplot(weights, panelCorr, totrows=1, totcols=1)

Description
Panel plots of various types.

Usage
panelplot(data, panel=points, totrows=3, totcols=2, oma=rep(2.5, 4), par.strip.text=NULL)

Arguments
data A list consisting of elements, each of which consists of x, y, xlim and ylim vectors
panel The panel function to be plotted
totrows The number of rows in the plot layout
totcols The number of columns in the plot layout
oma Outer margin area
par.strip.text A data frame with column cex

Author(s)
J.H. Maindonald

Examples
x1 <- x2 <- x3 <- (11:30)/5
y1 <- x1 + rnorm(20)/2
y2 <- 2 - 0.05 * x1 + 0.1 * (x1 - 1.75)^4 + 1.25 * rnorm(20)
r <- round(cor(x1, y2), 3)
rho <- round(cor(rank(x1), rank(y2)), 3)
y3 <- (x1 - 3.85)^2 + 0.015 + rnorm(20)/4
theta <- ((2 * pi) * (1:20))/20
x4 <- 10 + 4 * cos(theta)
y4 <- 10 + 4 * sin(theta) + (0.5 * rnorm(20))
r1 <- cor(x1, y1)
y <- data.frame(x = c(rep(x1, 3), x4), y = c(y1, y2, y3, y4),
               gp = rep(1:4, rep(20, 4)))
xy <- split(xy, xy$gp)
xlimdf <- lapply(list(x1,x2,x3,x4), range)
ylimdf <- lapply(list(y1,y2,y3,y4), range)
xy <- lapply(1:4, function(i,u,v,w){list(xlim=v[[i]], ylim=w[[i]],
                                           x=u[[i]]$x, y=u[[i]]$y)),
            u=xy, v=xlimdf, w=ylimdf)

panel.corr <- function (data, ...) {
  x <- data$x
  y <- data$y
  points(x, y, pch = 16)
  chh <- par()$cxy[2]
  x1 <- min(x)
  y1 <- max(y) - chh/4
  r1 <- cor(x, y)
  text(x1, y1, paste(round(r1, 3)), cex = 0.8, adj = 0)
}

panelplot(xy, panel=panel.corr, totrows=2, totcols=2, oma=rep(1,4))

pause

Description
If a program produces several plots, insertion of pause() between two plots suspends execution until the <Enter> key is pressed, to allow inspection of the current plot.

Usage
pause()

Author(s)
From the ‘sm’ package of Bowman and Azzalini (1997)

poissonsim
Simple Poisson Regression Data Simulator

Description
This function simulates simple regression data from a Poisson model. It also has the option to create over-dispersed data of a particular type.

Usage
poissonsim(x = seq(0, 1, length=101), a = 2, b = -4, intcp.sd=NULL, slope.sd=NULL)
possum

Arguments

- **x**: a numeric vector representing the explanatory variable
- **a**: the regression function intercept
- **b**: the regression function slope
- **intcp.sd**: standard deviation of the (random) intercept
- **slope.sd**: standard deviation of the (random) slope
- **seed**: numeric constant

Value

A list consisting of

- **x**: the explanatory variable vector
- **y**: the Poisson response vector

Examples

```r
poissonSim()
```

---

possum  

*Possum Measurements*

Description

The *possum* data frame consists of nine morphometric measurements on each of 104 mountain brush-tail possums, trapped at seven sites from Southern Victoria to central Queensland.

Usage

```r
possum
```

Format

This data frame contains the following columns:

- **case**: observation number
- **site**: one of seven locations where possums were trapped
- **Pop**: a factor which classifies the sites as *Vic* Victoria, *other* New South Wales or Queensland
- **sex**: a factor with levels *f* female, *m* male
- **age**: age
- **hdlngth**: head length
- **skullw**: skull width
- **totlngth**: total length
- **taill**: tail length
- **footlgth**: foot length
- **earconch**: ear conch length
- **eye**: distance from medial canthus to lateral canthus of right eye
- **chest**: chest girth (in cm)
- **belly**: belly girth (in cm)
Source


Examples

```r
boxplot(earconch ~ sex, data=possum)
pause()

sex <- as.integer(possum$sex)
oldpar <- par(oma=c(2,4,5,4))
pairs(possum[, c(9:11)], pch=c(0,2:7), col=c("red","blue"),
      labels=c("tail\nlength","foot\nlength","ear conch\nlength"))
chh <- par()$cxy[2]; xleg <- 0.05; yleg <- 1.04
oldpar <- par(xpd=TRUE)
legend(xleg, yleg, c("Cambarville", "Bellbird", "Whian Whian ",
                 "Byrangery", "Conondale ", "Allyn River", "Bulburin"),
      pch=c(0,2:7), x.intersp=1, y.intersp=0.75, cex=0.8, xjust=0, bty="n", ncol=4)
text(x=0.2, y=yleg - 2.25*chh, "female", col="red", cex=0.8, bty="n")
text(x=0.75, y=yleg - 2.25*chh, "male", col="blue", cex=0.8, bty="n")
par(oldpar)
pause()

sapply(possum[,6:14], function(x)max(x,na.rm=TRUE)/min(x,na.rm=TRUE))
pause()

here <- na.omit(possum$footlgth)
possum.prc <- princomp(possum[here, 6:14])
pause()

plot(possum.prc$scores[,1] ~ possum.prc$scores[,2],
      col=c("red","blue")[as.numeric(possum$sex[here])],
      pch=c(0,2:7)[possum$site[here]], xlab = "PC1", ylab = "PC2")
# NB: We have abbreviated the axis titles
chh <- par()$cxy[2]; xleg <- -15; yleg <- 20.5
oldpar <- par(xpd=TRUE)
legend(xleg, yleg, c("Cambarville", "Bellbird", "Whian Whian ",
                 "Byrangery", "Conondale ", "Allyn River", "Bulburin"),
      pch=c(0,2:7), x.intersp=1, y.intersp=0.75, cex=0.8, xjust=0, bty="n", ncol=4)
text(x=-9, y=yleg - 2.25*chh, "female", col="red", cex=0.8, bty="n")
summary(possum.prc, loadings=TRUE, digits=2)
par(oldpar)
pause()

require(MASS)
here <- !is.na(possum$footlgth)
possum.lda <- lda(site ~ hdlngth+skullw+totlngth+ taill+footlgth+
                  earconch+eye+chest+belly, data=possum, subset=here)
options(digits=4)
possum.lda$svd # Examine the singular values
plot(possum.lda, dimen=3)
# Scatterplot matrix - scores on 1st 3 canonical variates (Figure 11.4)
possum.lda
```
possumsites  

**Description**

The `possumsites` data frame consists of latitudes, longitudes, and altitudes for the seven sites from Southern Victoria to central Queensland where the possum observations were made.

**Usage**

```r
possumsites
```

**Format**

This data frame contains the following columns:

- **latitude**: a numeric vector
- **longitude**: a numeric vector
- **altitude**: in meters

**Source**


**Examples**

```r
require(oz)
or(sections=c(3:5, 11:16))
attach(possumsites)
points(latitude, longitude, pch=16, col=2)
chw <- par()$cxy[1]
chh <- par()$cxy[2]
posval <- c(2,4,2,2,4,2,2)
text(latitude+(3-posval)*chw/4, longitude, row.names(possumsites), pos=posval)
```

---

**powerplot**  

**Plot of Power Functions**

**Description**

This function plots powers of a variable on the interval [0,10].

**Usage**

```r
powerplot(expr="x^2", xlab="x", ylab="y")
```
Arguments

expr  Functional form to be plotted
xlab  x-axis label
ylab  y-axis label

Details

Other expressions such as "sin(x)" and "cos(x)", etc. could also be plotted with this function, but results are not guaranteed.

Value

A plot of the given expression on the interval [0,10].

Author(s)

J.H. Maindonald

Examples

oldpar <- par(mfrow = c(2, 3), mar = par()$mar - c(1, 1, 1.0, 1), mgp = c(1.5, 0.5, 0), oma=c(0,1,0,1))
  # on.exit(par(oldpar))
  powerplot(expr="sqrt(x)", xlab="")
  powerplot(expr="x^0.25", xlab="", ylab="")
  powerplot(expr="log(x)", xlab="", ylab="")
  powerplot(expr="x^2")
  powerplot(expr="x^4", ylab="")
  powerplot(expr="exp(x)", ylab="")
par(oldpar)

poxetc  Deaths from various causes, in London from 1629 till 1881, with gaps

Description

Deaths from "flux" or smallpox, measles, all causes, and ratios of the the first two categories to total deaths.

Usage

data(poxetc)

Format

This is a multiple time series consisting of 5 series: fpox, measles, all, fpox2all, measles2all.

Details
Source

References

Examples
data(poxetc)
str(poxetc)
plot(poxetc)

press Predictive Error Sum of Squares

Description
Allen’s PRESS statistic is computed for a fitted model.

Usage
press(obj)

Arguments
obj A lm object

Value
A single numeric value.

Author(s)
W.J. Braun

See Also
lm

Examples
litters.lm <- lm(brainwt ~ bodywt + lsize, data = litters)
press(litters.lm)
litters.lm0 <- lm(brainwt ~ bodywt + lsize -1, data=litters)
press(litters.lm0) # no intercept
litters.lm1 <- lm(brainwt ~ bodywt, data=litters)
press(litters.lm1) # bodywt only
litters.lm2 <- lm(brainwt ~ bodywt + lsize + lsize:bodywt, data=litters)
press(litters.lm2) # include an interaction term
Primates

Description

A subset of Animals data frame from the MASS library. It contains the average body and brain measurements of five primates.

Usage

primates

Format

This data frame contains the following columns:

- **Bodywt**: A numeric vector consisting of the body weights (in kg) of five different primates
- **Brainwt**: A numeric vector consisting of the corresponding brain weights (in g)

Source


Examples

```r
attach(primates)
plot(x = Bodywt, y = Brainwt, pch = 16,
     xlab = "Body weight (kg)", ylab = "Brain weight (g)",
     xlim = c(5, 300), ylim = c(0, 1500))
chw <- par()$cxy[1]
chh <- par()$cxy[2]
text(x = Bodywt + chw,
     y = Brainwt + c(-.1, 0, 0, .1, 0) * chh,
     labels = row.names(primates), adj = 0)
detach(primates)
```

qreference

Normal QQ Reference Plot

Description

This function computes the normal QQ plot for given data and allows for comparison with normal QQ plots of simulated data.

Usage

```r
qreference(test = NULL, m = 50, nrep = 6, distribution = function(x) qnorm(x, mean = ifelse(is.null(test), 0, mean(test)), sd = ifelse(is.null(test), 1, sd(test))), seed = NULL, nrows = NULL, cex.strip = 0.75, xlab = NULL, ylab = NULL)
```
Arguments

test a vector containing a sample to be tested; if not supplied, all qq-plots are of the reference distribution
m the sample size for the reference samples; default is test sample size if test sample is supplied
nrep the total number of samples, including reference samples and test sample if any
distribution reference distribution; default is standard normal
seed the random number generator seed
nrows number of rows in the plot layout
cex.strip character expansion factor for labels
xlab label for x-axis
ylab label for y-axis

Value

QQ plots of the sample (if test is non-null) and all reference samples

Author(s)

J.H. Maindonald

Examples

# qreference(rt(180,1))
# qreference(rt(180,1), distribution=function(x) qt(x, df=1))
# qreference(rexp(180), nrep = 4)
# toyvars.lm <- lm(distance ~ angle + factor(car), data = toyvars)
# qreference(residuals(toyvars.lm), nrep = 9)

races2000 Scottish Hill Races Data - 2000

Description

The record times in 2000 for 77 Scottish hill races. We believe the data are, for the most part, trustworthy. However, the dist variable for Caerketton (record 58) seems to have been variously recorded as 1.5 mi and 2.5 mi.

Usage

races2000
Format

This data frame contains the following columns:

- **h**: male record time in hours
- **m**: plus minutes
- **s**: plus seconds
- **h0**: female record time in hours
- **m0**: plus minutes
- **s0**: plus seconds
- **dist**: distance, in miles (on the map)
- **climb**: total height gained during the route, in feet
- **time**: record time in hours
- **timef**: record time in hours for females
- **type**: a factor, with levels indicating type of race, i.e. hill, marathon, relay, uphill or other

Source

The Scottish Running Resource, http://www.hillrunning.co.uk

Examples

```r
pairs(races2000[, -11])
```

---

**rainforest**

**Rainforest Data**

Description

The rainforest data frame has 65 rows and 7 columns.

Usage

```r
rainforest
```

Format

This data frame contains the following columns:

- **dbh**: a numeric vector
- **wood**: a numeric vector
- **bark**: a numeric vector
- **root**: a numeric vector
- **rootsk**: a numeric vector
- **branch**: a numeric vector
- **species**: a factor with levels *Acacia mabellae*, *C. fraseri*, *Acmena smithii*, *B. myrtifolia*
rareplants

Source

J. Ash, Australian National University

References


Examples

table(rainforest$species)

---

rareplants Rare and Endangered Plant Species

Description

These data were taken from species lists for South Australia, Victoria and Tasmania. Species were classified as CC, CR, RC and RR, with C denoting common and R denoting rare. The first code relates to South Australia and Victoria, and the second to Tasmania. They were further classified by habitat according to the Victorian register, where D = dry only, W = wet only, and WD = wet or dry.

Usage

rareplants

Format

The format is: chr "rareplants"

Source

Jasmyn Lynch, Department of Botany and Zoology at Australian National University

Examples

chisq.test(rareplants)
**rice**  
*Genetically Modified and Wild Type Rice Data*

**Description**

The *rice* data frame has 72 rows and 7 columns. The data are from an experiment that compared wild type (wt) and genetically modified rice plants (ANU843), each with three different chemical treatments (F10, NH4Cl, and NH4NO3).

**Usage**

`rice`

**Format**

This data frame contains the following columns:

- **PlantNo** a numeric vector
- **Block** a numeric vector
- **RootDryMass** a numeric vector
- **ShootDryMass** a numeric vector
- **trt** a factor with levels F10, NH4Cl, NH4NO3, F10 +ANU843, NH4Cl +ANU843, NH4NO3 +ANU843
- **fert** a factor with levels F10 NH4Cl NH4NO3
- **variety** a factor with levels wt ANU843

**Source**


**Examples**

```r
print("One and Two-Way Comparisons - Example 4.5")
attach(rice)
oldpar <- par(las = 2)
stripchart(ShootDryMass ~ trt, pch=1, cex=1, xlab="Level of factor 1")
detach(rice)
pause()

rice.aov <- aov(ShootDryMass ~ trt, data=rice); anova(rice.aov)
anova(rice.aov)
pause()

summary.lm(rice.aov)$coef
pause()

rice$trt <- relevel(rice$trt, ref="NH4Cl")
# Set NH4Cl as the baseline
```
roller

Lawn Roller Data

Description

The roller data frame has 10 rows and 2 columns. Different weights of roller were rolled over different parts of a lawn, and the depression was recorded.

Usage

roller

Format

This data frame contains the following columns:

- weight: a numeric vector consisting of the roller weights
- depression: the depth of the depression made in the grass under the roller

Source


Examples

plot(roller)
roller.lm <- lm(depression ~ weight, data = roller)
plot(roller.lm, which = 4)

science

School Science Survey Data

Description

The science data frame has 1385 rows and 7 columns.

The data are on attitudes to science, from a survey where there were results from 20 classes in private schools and 46 classes in public schools.
Usage

science

Format

This data frame contains the following columns:

- **State**  a factor with levels ACT Australian Capital Territory, NSW New South Wales
- **PrivPub** a factor with levels private school, public school
- **school** a factor, coded to identify the school
- **class** a factor, coded to identify the class
- **sex** a factor with levels f, m
- **like** a summary score based on two of the questions, on a scale from 1 (dislike) to 12 (like)
- **Class** a factor with levels corresponding to each class

Source

Francine Adams, Rosemary Martin and Murali Nayadu, Australian National University

Examples

attach(science)
classmeans <- aggregate(like, by=list(PrivPub, Class), mean)
names(classmeans) <- c("PrivPub","Class","like")
dim(classmeans)

attach(classmeans)
boxplot(split(like, PrivPub), ylab = "Class average of attitude to science score", boxwex = 0.4)
rug(like[PrivPub == "private"], side = 2)
rug(like[PrivPub == "public"], side = 4)
detach(classmeans)

require(nlme)

science.lme <- lme(fixed = like ~ sex + PrivPub, 
                   data = science, random = ~ 1 | school/Class, na.action=na.omit)
summary(science.lme)$tTable # Print coefficients.

science1.lme <- lme(fixed = like ~ sex + PrivPub, data = science, 
                     random = ~ 1 | Class, na.action=na.exclude)
summary(science1.lme)$tTable # Table of coefficients

intervals(science1.lme, which="var-cov")[[1]]$Class^2
intervals(science1.lme, which="var-cov")[[2]]^2

science.lme <- lme(fixed = like ~ sex + PrivPub, 
                   data = science, random = ~ 1 | Class/school, na.action=na.exclude)
res <- residuals(science.lme)
hat <- fitted(science.lme)
coplot(res ~ hat|science$PrivPub[is.na(science$sex)!=TRUE], 
       xlab="Fitted values", ylab="Residuals")
detach(science)
Description
The `seedrates` data frame has 5 rows and 2 columns on the effect of seeding rate of barley on yield.

Usage
```
seedrates
```

Format
This data frame contains the following columns:

- `rate` the seeding rate
- `grain` the number of grain per head of barley

Source

References

Examples
```
plot(grain~rate,data=seedrates,xlim=c(50,180),ylim=c(15.5,22),axes=FALSE)
new.df<-data.frame(rate=(2:8)*25)
seedrates.lm1<-lm(grain~rate,data=seedrates)
seedrates.lm2<-lm(grain~rate+I(rate^2),data=seedrates)
hat1<-predict(seedrates.lm1,newdata=new.df,interval="confidence")
hat2<-predict(seedrates.lm2,newdata=new.df,interval="confidence")
axis(1,at=new.df$rate); axis(2); box()
z1<-spline(new.df$rate, hat1[,"fit"]); z2<-spline(new.df$rate, hat2[,"fit"]) rate<-new.df$rate; lines(z1$x,z1$y)
lines(spline(rate,hat1[,"lwr"]),lty=1,col=3)
lines(spline(rate,hat1[,"upr"]),lty=1,col=3)
lines(z2$x,z2$y,lty=4)
lines(spline(rate,hat2[,"lwr"]),lty=4,col=3)
lines(spline(rate,hat2[,"upr"]),lty=4,col=3)
```
show.colors  

*Show R's Colors*

Description

This function displays the built-in colors.

Usage

```r
show.colors(type=c("singles", "shades", "gray"), order.cols=TRUE)
```

Arguments

- **type**: type of display - single, multiple or gray shades
- **order.cols**: Arrange colors in order

Value

A plot of colors for which there is a single shade (type = "single"), multiple shades (type = "multiple"), or gray shades (type = "gray")

Author(s)

J.H. Maindonald

Examples

```r
require(MASS)
show.colors()
```

simulateLinear  

*Simulation of Linear Models for ANOVA vs. Regression Comparison*

Description

This function simulates a number of bivariate data sets in which there are replicates at each level of the predictor. The p-values for ANOVA and for the regression slope are compared.

Usage

```r
simulateLinear(sd=2, npoints=5, nrep=4, nsets=200, type="xy", seed=21)
```

Arguments

- **sd**: The error standard deviation
- **npoints**: Number of distinct predictor levels
- **nrep**: Number of replications at each level
- **nsets**: Number of simulation runs
- **type**: Type of data
- **seed**: Random Number generator seed
Value

The proportion of regression p-values that are less than the ANOVA p-values is printed

Author(s)

J.H. Maindonald

Examples

simulateLinear()

socsupport  Social Support Data

Description

Data from a survey on social and other kinds of support.

Usage

socsupport

Format

This data frame contains the following columns:

gender  a factor with levels female, male
age  age, in years, with levels 18–20, 21–24, 25–30, 31–40, 40+
country  a factor with levels australia, other
marital  a factor with levels married, other, single
livewith  a factor with levels alone, friends, other, parents, partner, residences
employment  a factor with levels employed fulltime, employed part-time, govt assistance, other, parental support
firstyr  a factor with levels first year, other
enrolment  a factor with levels full-time, part-time
emotional  summary of 5 questions on emotional support availability
emotionalsat  summary of 5 questions on emotional support satisfaction
tangible  summary of 4 questions on availability of tangible support
tangiblesat  summary of 4 questions on satisfaction with tangible support
affect  summary of 3 questions on availability of affectionate support sources
affectsat  summary of 3 questions on satisfaction with affectionate support sources
psi  summary of 3 questions on availability of positive social interaction
psisat  summary of 3 questions on satisfaction with positive social interaction
esupport  summary of 4 questions on extent of emotional support sources
psupport  summary of 4 questions on extent of practical support sources
supsources  summary of 4 questions on extent of social support sources (formerly, socsupport)
BDI  Score on the Beck depression index (summary of 21 questions)
Source

Melissa Manning, Psychology, Australian National University

Examples

attach(socsupport)

not.na <- apply(socsupport[,9:19], 1, function(x)!any(is.na(x)))
ss.pr1 <- princomp(as.matrix(socsupport[not.na, 9:19]), cor=TRUE)
pairs(ss.pr1$scores[,1:3])
sort(~ss.pr1$scores[,1])  # Minus the largest value appears first
pause()

not.na[36] <- FALSE
ss.pr <- princomp(as.matrix(socsupport[not.na, 9:19]), cor=TRUE)
summary(ss.pr)  # Examine the contribution of the components
pause()

# We now regress BDI on the first six principal components:
ss.lm <- lm(BDI[not.na] ~ ss.pr$scores[, 1:6], data=socsupport)
summary(ss.lm)$coef
pause()

ss.pr$loadings[,1]
plot(BDI[not.na] ~ ss.pr$scores[,1], col=as.numeric(gender),
pch=as.numeric(gender), xlab="1st principal component", ylab="BDI")
topleft <- par()$usr[c(1,4)]
legend(topleft[1], topleft[2], col=1:2, pch=1:2, legend=levels(gender))

softbacks  Measurements on a Selection of Paperback Books

Description

This is a subset of the allbacks data frame which gives measurements on the volume and weight of 8 paperback books.

Usage

softbacks

Format

This data frame contains the following columns:

volume  a numeric vector giving the book volumes in cubic centimeters
weight  a numeric vector giving the weights in grams

Source

The bookshelf of J. H. Maindonald.
Examples

print("Outliers in Simple Regression - Example 5.2")
paperback.lm <- lm(weight ~ volume, data=softbacks)
summary(paperback.lm)
plot(paperback.lm)

sorption

sorption data set

Description

Concentration-time measurements on different varieties of apples under methyl bromide injection.

Usage

data(sorption)

Format

A data frame with 192 observations on the following 14 variables.

- **m5** a numeric vector
- **m10** a numeric vector
- **m30** a numeric vector
- **m60** a numeric vector
- **m90** a numeric vector
- **m120** a numeric vector
- **ct** concentration-time
- **Cultivar** a factor with levels Pacific Rose BRAEBURN Fuji GRANNY Gala ROYAL Red Delicious Splendour
- **Dose** injected dose of methyl bromide
- **rep** replicate number, within Cultivar and year
- **year** a factor with levels 1988 1989 1998 1999
- **gp** a factor with levels BRAEBURN1 BRAEBURN2 Fuji1 Fuji10 Fuji2 Fuji6 Fuji7 Fuji8 Fuji9 GRANNY1 GRANNY2 Gala4 Gala5 Pacific Rose10 Pacific Rose6 Pacific Rose7 Pacific Rose8 Pacific Rose9 ROYAL1 ROYAL2 Red Del10 Red Del19 Red Delicious1 Red Delicious2 Red Delicious3 Red Delicious4 Red Delicious5 Red Delicious6 Red Delicious7 Red Delicious8 Splendour4 Splendour5 Splendour6 Splendour7 Splendour8 Splendour9 Splendour10
- **inyear** a factor with levels 1 2 3 4 5 6
**spam7**  
*Spam E-mail Data*

**Description**

The data consist of 4601 email items, of which 1813 items were identified as spam.

**Usage**

`spam7`

**Format**

This data frame contains the following columns:

- `crl.tot` total length of words in capitals
- `dollar` number of occurrences of the $ symbol
- `bang` number of occurrences of the ! symbol
- `money` number of occurrences of the word ‘money’
- `n000` number of occurrences of the string ‘000’
- `make` number of occurrences of the word ‘make’
- `yesno` outcome variable, a factor with levels `n` not spam, `y` spam

**Source**

George Forman, Hewlett-Packard Laboratories

These data are available from the University of California at Irvine Repository of Machine Learning Databases and Domain Theories. The address is: http://www.ics.uci.edu/ Here

**Examples**

```r
require(rpart)
spam.rpart <- rpart(formula = yesno ~ crl.tot + dollar + bang +
                    money + n000 + make, data=spam7)
plot(spam.rpart)
text(spam.rpart)
```

---

**stVincent**  
*Averages by block of yields for the St. Vincent Corn data*

**Description**

These data frames have yield averages by blocks (parcels).

**Usage**

`stVincent`
sugar

Format
A data frame with 324 observations on 8 variables.

code  a numeric vector
island a numeric vector
id    a numeric vector
site   a factor with 8 levels.
block  a factor with levels I II III IV
plot   a numeric vector
trt    a factor consisting of 12 levels
harvwt a numeric vector; the average yield

Source
Student and Research Worker. Springer-Verlag. (pp. 339-353)

sugar               Sugar Data

Description
The sugar data frame has 12 rows and 2 columns. They are from an experiment that compared an
unmodified wild type plant with three different genetically modified forms. The measurements are
weights of sugar that were obtained by breaking down the cellulose.

Usage
sugar

Format
This data frame contains the following columns:

weight weight, in mg
trt  a factor with levels Control i.e. unmodified Wild form, A Modified 1, B Modified 2, C
     Modified 3

Source
Anonymous

Examples
sugar.aov <- aov(weight ~ trt, data=sugar)
fitted.values(sugar.aov)
summary.lm(sugar.aov)
sugar.aov <- aov(formula = weight ~ trt, data = sugar)
summary.lm(sugar.aov)
tinting  Car Window Tinting Experiment Data

Description

These data are from an experiment that aimed to model the effects of the tinting of car windows on visual performance. The authors were mainly interested in effects on side window vision, and hence in visual recognition tasks that would be performed when looking through side windows.

Usage
tinting

Format

This data frame contains the following columns:

- **case**: observation number
- **id**: subject identifier code (1-26)
- **age**: age (in years)
- **sex**: a factor with levels f female, m male
- **tint**: an ordered factor with levels representing degree of tinting: no < lo < hi
- **target**: a factor with levels locon: low contrast, hicon: high contrast
- **it**: the inspection time, the time required to perform a simple discrimination task (in milliseconds)
- **csoa**: critical stimulus onset asynchrony, the time to recognize an alphanumeric target (in milliseconds)
- **agegp**: a factor with levels younger, 21-27, older, 70-78

Details

Visual light transmittance (VLT) levels were 100% (tint=none), 81.3% (tint=lo), and 35.1% (tint=hi). Based on these and other data, Burns et al. argue that road safety may be compromised if the front side windows of cars are tinted to 35

Source


Examples

```r
require(lattice)
levels(tinting$agegp) <- capstring(levels(tinting$agegp))
xyplot(csoa ~ it | sex * agegp, data=tinting) # Simple use of xyplot()
pause()

xyplot(csoa ~ it | sex * agegp, data=tinting, panel=panel.superpose, groups=target)
pause()

xyplot(csoa ~ it | sex * agegp, data=tinting, panel=panel.superpose, col=1:2,
```
Description

The `toycars` data frame has 27 rows and 3 columns. Observations are on the distance traveled by one of three different toy cars on a smooth surface, starting from rest at the top of a 16 inch long ramp tilted at varying angles.

Usage

toycars

Format

This data frame contains the following columns:

- `angle` tilt of ramp, in degrees
- `distance` distance traveled, in meters
- `car` a numeric code (1 = first car, 2 = second car, 3 = third car)

Examples

toycars.lm <- lm(distance ~ angle + factor(car), data=toycars)
summary(toycars.lm)

Description

Twenty-one elastic bands were divided into two groups.

One of the sets was placed in hot water (60-65 degrees C) for four minutes, while the other was left at ambient temperature. After a wait of about ten minutes, the amounts of stretch, under a 1.35 kg weight, were recorded.

Usage

pair65
Format

This list contains the following elements:

heated a numeric vector giving the stretch lengths for the heated bands
ambient a numeric vector giving the stretch lengths for the unheated bands

Source

J.H. Maindonald

Examples

twot.permutation(two65$ambient, two65$heated) # two sample permutation test

twot.permutation()  \textit{Two Sample Permutation Test - Obsolete}

Description

This function computes the p-value for the two sample t-test using a permutation test. The permutation density can also be plotted.

Usage

twot.permutation(x1=two65$ambient, x2=two65$heated, nsim=2000, plotit=TRUE)

Arguments

\begin{itemize}
\item \texttt{x1} Sample 1
\item \texttt{x2} Sample 2
\item \texttt{nsim} Number of simulations
\item \texttt{plotit} If TRUE, the permutation density will be plotted
\end{itemize}

Details

Suppose we have $n_1$ values in one group and $n_2$ in a second, with $n = n_1 + n_2$. The permutation distribution results from taking all possible samples of $n_2$ values from the total of $n$ values.

Value

The p-value for the test of the hypothesis that the mean of $x_1$ differs from $x_2$

Author(s)

J.H. Maindonald

References


Examples

twot.permutation()
**Two Sample Permutation Test**

**Description**

This function computes the p-value for the two sample t-test using a permutation test. The permutation density can also be plotted.

**Usage**

```r
twotPermutation(x1=two65$ambient, x2=two65$heated, nsim=2000, plotit=TRUE)
```

**Arguments**

- `x1`: Sample 1
- `x2`: Sample 2
- `nsim`: Number of simulations
- `plotit`: If TRUE, the permutation density will be plotted

**Details**

Suppose we have \( n_1 \) values in one group and \( n_2 \) in a second, with \( n = n_1 + n_2 \). The permutation distribution results from taking all possible samples of \( n_2 \) values from the total of \( n \) values.

**Value**

The p-value for the test of the hypothesis that the mean of \( x_1 \) differs from \( x_2 \)

**Author(s)**

J.H. Maindonald

**References**


**Examples**

```r
twotPermutation()
```
Variance Inflation Factors

Description
Variances inflation factors are computed for the standard errors of linear model coefficient estimates.

Usage
vif(obj, digits=5)

Arguments
- **obj**: A `lm` object
- **digits**: Number of digits

Value
A vector of variance inflation factors corresponding to the coefficient estimates given in the `lm` object.

Author(s)
J.H. Maindonald

See Also
lm

Examples
litters.lm <- lm(brainwt ~ bodywt + lsize, data = litters)
vif(litters.lm)

carprice1.lm <- lm(gpm100 ~ Type+Min.Price+Price+Max.Price+Range.Price, data=carprice)
vif(carprice1.lm)

carprice.lm <- lm(gpm100 ~ Type + Price, data = carprice)
vif(carprice1.lm)
Averages by block of corn yields, for treatment 111 only

**Description**

These data frames have averages by blocks (parcels) for the treatment 111.

**Usage**

`vincell1b`

**Format**

A data frame with 36 observations on 8 variables.

- **site** a factor with levels AGSV CASV CPSV LPSV MPSV OOSV OTSV SSSV UISV
- **parcel** a factor with levels I II III IV
- **code** a numeric vector
- **island** a numeric vector
- **id** a numeric vector
- **plot** a numeric vector
- **trt** a numeric vector
- **harvwt** a numeric vector

**Source**


---

**Video Lottery Terminal Data**

**Description**

Data on objects appearing in three windows on a video lottery terminal, together with the prize payout (usually 0). Observations were taken on two successive days in late 1994 at a hotel lounge north of Winnipeg, Manitoba. Each observation cost 25 cents (Canadian). The game played was ‘Double Diamond’.

**Usage**

`vlt`
Format

This data frame contains the following columns:

- window1 object appearing in the first window.
- window2 object appearing in the second window.
- window3 object appearing in the third window.
- prize cash prize awarded (in Canadian dollars).
- night 1, if observation was taken on day 1; 2, if observation was taken on day 2.

Details

At each play, each of three windows shows one of 7 possible objects. Apparently, the three windows are independent of each other, and the objects should appear with equal probability across the three windows. The objects are coded as follows: blank (0), single bar (1), double bar (2), triple bar (3), double diamond (5), cherries (6), and the numeral "7" (7).

Prizes (in quarters) are awarded according to the following scheme: 800 (5-5-5), 80 (7-7-7), 40 (3-3-3), 25 (2-2-2), 10 (1-1-1), 10 (6-6-6), 5 (2 "6"s), 2 (1 "6") and 5 (any combination of "1", "2" and "3"). In addition, a "5" doubles any winning combination, e.g. (5-3-3) pays 80 and (5-3-5) pays 160.

Source


Examples

```r
vlt.stk <- stack(vlt[,1:3])
table(vlt.stk)
```

wages1833

Wages of Lancashire Cotton Factory Workers in 1833

Description

The wages1833 data frame gives the wages of Lancashire cotton factory workers in 1833.

Usage

```r
wages1833
```

Format

This data frame contains the following columns:

- age age in years
- mnum number of male workers
- mwage average wage of male workers
- fnum number of female workers
- fwage average wage of female workers
Source


Examples

```r
attach(wages1833)
plot(mwage~age,ylim=range(c(mwage,fwage[fwage>0])))
points(fwage[fwage>0]~age[fwage>0],pch=15,col="red")
lines(lowess(age,mwage))
lines(lowess(age[fwage>0],fwage[fwage>0]),col="red")
```

**whoops**  
*Deaths from whooping cough, in London*

Description

Deaths from whooping cough, in London from 1740 to 1881.

Usage

```r
data(whoops)
```

Format

This is a multiple time series consisting of 3 series: wcough, ratio, and alldeaths.

Source


References


Examples

```r
data(whoops)
str(whoops)
plot(whoops)
```
Index

*Topic IO
  hardcopy, 43

*Topic datasets
  ACF1, 1
  ais, 7
  allbacks, 8
  anesthetic, 9
  ant111b, 10
  antigua, 11
  appletaste, 11
  austpop, 12
  biomass, 14
  bomsoi, 15
  bomsoi2001, 17
  bostonc, 20
  carprice, 22
  Cars93.summary, 4
  cfseal, 24
  cities, 25
  codling, 26
  cottonworkers, 29
  cuckoohosts, 30
  cuckoos, 31
  dengue, 34
  dewpoint, 35
  elastic1, 36
  elastic2, 37
  elasticband, 38
  fossilfuel, 39
  frogs, 40
  frostedflakes, 41
  fruitohms, 42
  geophones, 43
  head.injury, 44
  headInjury, 45
  hills, 46
  hills2000, 47
  houseprices, 48
  humanpower, 49
  ironslag, 50
  jobs, 51
  kiwishade, 52
  leafshape, 54
  leafshape17, 55
  leaftemp, 56
  leaftemp.all, 57
  litters, 57
  Lottario, 5
  Manitoba.lakes, 6
  measles, 59
  mifem, 60
  mignonette, 61
  milk, 61
  modelcars, 62
  monica, 63
  moths, 64
  nsw74demo, 65
  nsw74psid1, 66
  nsw74psid3, 67
  nsw74psidA, 68
  oddbooks, 69
  orings, 74
  ozone, 76
  pair65, 77
  possum, 81
  possumsites, 83
  poxetc, 84
  primates, 86
  races2000, 87
  rainforest, 88
  rareplants, 89
  rice, 90
  roller, 91
  science, 91
  seedrates, 93
  socsupport, 95
  softbacks, 96
  sorption, 97
  spam7, 98
  stVincent, 98
  sugar, 99
  tinting, 100
  toycars, 101
  two65, 101
  vince111b, 105
  vlt, 105
INDEX

lm, 14, 28
logisticsim, 58
Lottario, 5
Manitoba.lakes, 6
measles, 59
mifem, 60
mignonette, 61
milk, 61
modelcars, 62
monica, 63
moths, 64
multilap, 64
nsw74demo, 65
nsw74psid1, 66
nsw74psid3, 67
nsw74psidA, 68
obounce, 69
oddbooks, 69
onesamp, 70
onet.permutation, 71
onetPermutation, 71
oneway.plot, 72
onewayPlot, 72
orings, 74
overlap.density, 74
overlapDensity, 75
ozone, 76
pair65, 77
panel.corr, 78
panelCorr, 78
panelplot, 79
pause, 80
poissonsim, 80
possum, 81
possumsites, 83
powerplot, 83
poxetc, 84
press, 85
primates, 86
qreference, 86
races2000, 87
rainforest, 88
rareplants, 89
rice, 90
roller, 91
science, 91
seedrates, 93
show.colors, 94
simulateLinear, 94
soqsupport, 95
softbacks, 96
sorption, 97
spam7, 98
stVincent, 98
sugar, 99
summary.lm, 13
tinting, 100
toycars, 101
two65, 101
twot.permutation, 102
twotPermutation, 103
vif, 104
vincelllb, 105
vlt, 105
wages1833, 106
whoops, 107