The VR Package
October 4, 2007

Priority  recommended
Contains  MASS class nnet spatial
Version  7.2-37
Date  2007-10-03
Depends  R (>= 2.4.0), grDevices, graphics, stats, utils
Suggests  lattice, nlme, survival
Author  S original by Venables & Ripley. R port by Brian Ripley <ripley@stats.ox.ac.uk>, following earlier work by Kurt Hornik and Albrecht Gebhardt.
Maintainer  Brian Ripley <ripley@stats.ox.ac.uk>
License  GPL-2 | GPL-3
URL  http://www.stats.ox.ac.uk/pub/MASS4/
## Contents

### Package ‘MASS’

<table>
<thead>
<tr>
<th>Package Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aids2</td>
<td>1</td>
</tr>
<tr>
<td>Animals</td>
<td>2</td>
</tr>
<tr>
<td>Boston</td>
<td>3</td>
</tr>
<tr>
<td>Cars93</td>
<td>4</td>
</tr>
<tr>
<td>Cushing</td>
<td>5</td>
</tr>
<tr>
<td>DDT</td>
<td>6</td>
</tr>
<tr>
<td>GAurine</td>
<td>6</td>
</tr>
<tr>
<td>Insurance</td>
<td>7</td>
</tr>
<tr>
<td>MASS-internal</td>
<td>8</td>
</tr>
<tr>
<td>Melanoma</td>
<td>8</td>
</tr>
<tr>
<td>Null</td>
<td>9</td>
</tr>
<tr>
<td>OME</td>
<td>10</td>
</tr>
<tr>
<td>Pima.tr</td>
<td>13</td>
</tr>
<tr>
<td>Rabbit</td>
<td>14</td>
</tr>
<tr>
<td>Rubber</td>
<td>14</td>
</tr>
<tr>
<td>SP500</td>
<td>15</td>
</tr>
<tr>
<td>Sitka</td>
<td>16</td>
</tr>
<tr>
<td>Sitka89</td>
<td>16</td>
</tr>
<tr>
<td>Skye</td>
<td>17</td>
</tr>
<tr>
<td>Traffic</td>
<td>18</td>
</tr>
<tr>
<td>UScereal</td>
<td>19</td>
</tr>
<tr>
<td>UScrime</td>
<td>20</td>
</tr>
<tr>
<td>VA</td>
<td>21</td>
</tr>
<tr>
<td>abbey</td>
<td>22</td>
</tr>
<tr>
<td>accdeaths</td>
<td>22</td>
</tr>
<tr>
<td>addterm</td>
<td>23</td>
</tr>
<tr>
<td>anorexia</td>
<td>24</td>
</tr>
<tr>
<td>anova.negbin</td>
<td>25</td>
</tr>
<tr>
<td>area</td>
<td>26</td>
</tr>
<tr>
<td>bacteria</td>
<td>27</td>
</tr>
<tr>
<td>bandwidth.nrd</td>
<td>28</td>
</tr>
<tr>
<td>bcv</td>
<td>29</td>
</tr>
<tr>
<td>beav1</td>
<td>30</td>
</tr>
<tr>
<td>beav2</td>
<td>31</td>
</tr>
<tr>
<td>biopsy</td>
<td>33</td>
</tr>
<tr>
<td>birthwt</td>
<td>34</td>
</tr>
<tr>
<td>Contents</td>
<td>Page</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
</tr>
<tr>
<td>boxcox</td>
<td>35</td>
</tr>
<tr>
<td>cabbages</td>
<td>36</td>
</tr>
<tr>
<td>caith</td>
<td>37</td>
</tr>
<tr>
<td>cats</td>
<td>38</td>
</tr>
<tr>
<td>cement</td>
<td>39</td>
</tr>
<tr>
<td>chem</td>
<td>39</td>
</tr>
<tr>
<td>con2tr</td>
<td>40</td>
</tr>
<tr>
<td>confint-MASS</td>
<td>41</td>
</tr>
<tr>
<td>contr.sdif</td>
<td>42</td>
</tr>
<tr>
<td>coop</td>
<td>43</td>
</tr>
<tr>
<td>corresp</td>
<td>44</td>
</tr>
<tr>
<td>cov.rob</td>
<td>45</td>
</tr>
<tr>
<td>cov.trob</td>
<td>47</td>
</tr>
<tr>
<td>cpus</td>
<td>49</td>
</tr>
<tr>
<td>crabs</td>
<td>50</td>
</tr>
<tr>
<td>deaths</td>
<td>51</td>
</tr>
<tr>
<td>denumerate</td>
<td>51</td>
</tr>
<tr>
<td>dose.p</td>
<td>52</td>
</tr>
<tr>
<td>drivers</td>
<td>53</td>
</tr>
<tr>
<td>dropterm</td>
<td>54</td>
</tr>
<tr>
<td>eagles</td>
<td>55</td>
</tr>
<tr>
<td>epil</td>
<td>56</td>
</tr>
<tr>
<td>eqscplot</td>
<td>58</td>
</tr>
<tr>
<td>farms</td>
<td>59</td>
</tr>
<tr>
<td>fgl</td>
<td>60</td>
</tr>
<tr>
<td>fitdistr</td>
<td>61</td>
</tr>
<tr>
<td>forbes</td>
<td>62</td>
</tr>
<tr>
<td>fractions</td>
<td>63</td>
</tr>
<tr>
<td>galaxies</td>
<td>64</td>
</tr>
<tr>
<td>gamma.dispersion</td>
<td>65</td>
</tr>
<tr>
<td>gamma.shape</td>
<td>66</td>
</tr>
<tr>
<td>gehan</td>
<td>67</td>
</tr>
<tr>
<td>genotype</td>
<td>69</td>
</tr>
<tr>
<td>geysers</td>
<td>69</td>
</tr>
<tr>
<td>gilgais</td>
<td>70</td>
</tr>
<tr>
<td>ginv</td>
<td>71</td>
</tr>
<tr>
<td>glm.convert</td>
<td>72</td>
</tr>
<tr>
<td>glm.nb</td>
<td>73</td>
</tr>
<tr>
<td>glmnPQL</td>
<td>74</td>
</tr>
<tr>
<td>hills</td>
<td>75</td>
</tr>
<tr>
<td>hist.scott</td>
<td>76</td>
</tr>
<tr>
<td>housing</td>
<td>76</td>
</tr>
<tr>
<td>huber</td>
<td>78</td>
</tr>
<tr>
<td>hubers</td>
<td>79</td>
</tr>
<tr>
<td>immer</td>
<td>80</td>
</tr>
<tr>
<td>isoMDS</td>
<td>81</td>
</tr>
<tr>
<td>kde2d</td>
<td>82</td>
</tr>
<tr>
<td>lda</td>
<td>84</td>
</tr>
</tbody>
</table>
CONTENTS

ldahist ................................................................. 86
leuk ................................................................. 87
lm.gls ............................................................... 88
lm.ridge ............................................................ 89
loglm ................................................................. 91
loglm1 .............................................................. 94
logtrans ............................................................. 95
lqs ................................................................. 96
mammals ............................................................ 99
mca ................................................................. 100
mcycle .............................................................. 101
menarche ........................................................... 101
michelson .......................................................... 102
minn38 .............................................................. 103
motors ............................................................... 103
muscle ............................................................... 104
mvnorm ............................................................. 106
negative.binomial ................................................... 107
newcomb ............................................................ 108
nlschools ........................................................... 108
npk ................................................................. 109
npr1 ................................................................. 110
oats ................................................................. 111
painters ............................................................. 112
pairs.lda ........................................................... 113
parcoord ............................................................ 114
petrol ............................................................... 115
Belgian-phones ..................................................... 116
plot.lda ............................................................. 116
plot.mca ............................................................ 117
polr ................................................................. 118
predict.glmmPQL .................................................... 120
predict.lda .......................................................... 121
predict.lqs .......................................................... 123
predict.mca ........................................................ 124
predict.qda ........................................................ 125
qda ................................................................. 126
quine ............................................................... 128
rational ............................................................. 129
renumerate .......................................................... 130
rlm ................................................................. 131
rms.curv ............................................................ 133
rnegbin ............................................................. 134
road ................................................................. 135
rotifer ............................................................... 136
sammon ............................................................. 136
ships ............................................................... 138
shoes ............................................................... 138
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>shrimp</td>
<td>139</td>
</tr>
<tr>
<td>shuttle</td>
<td>139</td>
</tr>
<tr>
<td>snails</td>
<td>140</td>
</tr>
<tr>
<td>stdres</td>
<td>141</td>
</tr>
<tr>
<td>steam</td>
<td>142</td>
</tr>
<tr>
<td>stepAIC</td>
<td>142</td>
</tr>
<tr>
<td>stormer</td>
<td>145</td>
</tr>
<tr>
<td>studres</td>
<td>146</td>
</tr>
<tr>
<td>summary.loglm</td>
<td>146</td>
</tr>
<tr>
<td>summary.negbin</td>
<td>147</td>
</tr>
<tr>
<td>summary.rlm</td>
<td>148</td>
</tr>
<tr>
<td>survey</td>
<td>150</td>
</tr>
<tr>
<td>synth.tr</td>
<td>151</td>
</tr>
<tr>
<td>theta.md</td>
<td>151</td>
</tr>
<tr>
<td>topo</td>
<td>153</td>
</tr>
<tr>
<td>truehist</td>
<td>153</td>
</tr>
<tr>
<td>ucv</td>
<td>155</td>
</tr>
<tr>
<td>waders</td>
<td>155</td>
</tr>
<tr>
<td>whiteside</td>
<td>157</td>
</tr>
<tr>
<td>width.SJ</td>
<td>158</td>
</tr>
<tr>
<td>write.matrix</td>
<td>159</td>
</tr>
<tr>
<td>wtloss</td>
<td>160</td>
</tr>
</tbody>
</table>

Package ‘class’  

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOM</td>
<td>161</td>
</tr>
<tr>
<td>batchSOM</td>
<td>162</td>
</tr>
<tr>
<td>condense</td>
<td>164</td>
</tr>
<tr>
<td>knn</td>
<td>165</td>
</tr>
<tr>
<td>knn.cv</td>
<td>166</td>
</tr>
<tr>
<td>knn1</td>
<td>167</td>
</tr>
<tr>
<td>lvq1</td>
<td>168</td>
</tr>
<tr>
<td>lvq2</td>
<td>169</td>
</tr>
<tr>
<td>lvq3</td>
<td>170</td>
</tr>
<tr>
<td>lvqinit</td>
<td>171</td>
</tr>
<tr>
<td>lvqtest</td>
<td>172</td>
</tr>
<tr>
<td>multiedit</td>
<td>173</td>
</tr>
<tr>
<td>olvq1</td>
<td>174</td>
</tr>
<tr>
<td>reduce.nn</td>
<td>175</td>
</tr>
<tr>
<td>somgrid</td>
<td>176</td>
</tr>
</tbody>
</table>

Package ‘nnet’  

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>class.ind</td>
<td>179</td>
</tr>
<tr>
<td>multinom</td>
<td>180</td>
</tr>
<tr>
<td>nnetHess</td>
<td>182</td>
</tr>
<tr>
<td>nnet</td>
<td>183</td>
</tr>
<tr>
<td>predict.nnet</td>
<td>185</td>
</tr>
<tr>
<td>which.is.max</td>
<td>186</td>
</tr>
</tbody>
</table>
Package ‘spatial’

- Kaver .................................................. 189
- Kenvl .................................................. 190
- Kfn ...................................................... 191
- Psim ..................................................... 192
- SSI ....................................................... 193
- Strauss .................................................. 194
- anova.trls ............................................. 195
- correlogram .......................................... 196
- expcov ................................................. 197
- ppgetregion .......................................... 198
- ppinit ................................................. 198
- pplik ................................................. 199
- ppregion ............................................... 200
- predict.trls .......................................... 201
- prmat .................................................. 202
- semat ................................................... 203
- surf.gls ............................................... 204
- surf.ls ............................................... 205
- trls.influence ....................................... 206
- trmat .................................................. 207
- variogram ............................................ 208

Index .................................................... 211
Package ‘MASS’

Description The main library and the datasets

Title Main Package of Venables and Ripley’s MASS

LazyLoad yes

LazyData yes

Aids2 Australian AIDS Survival Data

Description

Data on patients diagnosed with AIDS in Australia before 1 July 1991.

Usage

Aids2

Format

This data frame contains 2843 rows and the following columns:

- **state** Grouped state of origin: NSW includes ACT and "other" is WA, SA, NT and TAS.
- **sex** Sex of patient
- **diag** (Julian) date of diagnosis
- **death** (Julian) date of death or end of observation
- **status** "A" (alive) or "D" (dead) at end of observation
- **T.categ** Reported transmission category
- **age** Age (years) at diagnosis

Note

This data set has been slightly jittered as a condition of its release, to ensure patient confidentiality.
Source

Dr P. J. Solomon and the Australian National Centre in HIV Epidemiology and Clinical Research.

References


---

**Animals**

*Brain and Body Weights for 28 Species*

---

**Description**

Average brain and body weights for 28 species of land animals.

**Usage**

Animals

**Format**

- **body**  body weight in kg
- **brain**  brain weight in g

**Note**

The name Animals avoids conflicts with a system dataset animals in S-PLUS 4.5 and later.

**Source**


**References**

Boston  

**Housing Values in Suburbs of Boston**

**Description**

The Boston data frame has 506 rows and 14 columns.

**Usage**

Boston

**Format**

This data frame contains the following columns:

- **crim** per capita crime rate by town
- **zn** proportion of residential land zoned for lots over 25,000 sq.ft.
- **indus** proportion of non-retail business acres per town
- **chas** Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- **nox** nitrogen oxides concentration (parts per 10 million)
- **rm** average number of rooms per dwelling
- **age** proportion of owner-occupied units built prior to 1940
- **dis** weighted mean of distances to five Boston employment centres
- **rad** index of accessibility to radial highways
- **tax** full-value property-tax rate per $10,000
- **ptratio** pupil-teacher ratio by town
- **black** $1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town
- **lstat** lower status of the population (percent)
- **medv** median value of owner-occupied homes in $1000

**Source**


Data from 93 Cars on Sale in the USA in 1993

Description

The Cars93 data frame has 93 rows and 27 columns.

Usage

Cars93

Format

This data frame contains the following columns:

- **Manufacturer**: Manufacturer
- **Model**: Model
- **Type**: Type: Small, Sporty, Compact, Midsize, Large, Van
- **Min.Price**: Minimum Price (in $1,000) - price for a basic version
- **Price**: Midrange Price (in $1,000) - average of Min.Price and Max.Price
- **Max.Price**: Maximum Price (in $1,000) - price for “a premium version”
- **MPG.city**: City MPG (miles per US gallon by EPA rating)
- **MPG.highway**: Highway MPG
- **AirBags**: Air Bags standard. Factor: none, driver only, or driver & passenger
- **DriveTrain**: Drive train type: rear wheel, front wheel or 4WD; (factor).
- **Cylinders**: Number of cylinders (missing for Mazda RX-7, which has a rotary engine).
- **EngineSize**: Engine size (litres)
- **Horsepower**: Horsepower (maximum)
- **RPM**: RPM (revs per minute at maximum horsepower)
- **Rev.per.mile**: Engine revolutions per mile (in highest gear)
- **Man.trans.avail**: Is a manual transmission version available? (yes or no, Factor).
- **Fuel.tank.capacity**: Fuel tank capacity (US gallons)
- **Passengers**: Passenger capacity (persons)
- **Length**: Length (inches)
- **Wheelbase**: Wheelbase (inches)
- **Width**: Width (inches)
- **Turn.circle**: U-turn space (feet)
- **Rear.seat.room**: Rear seat room (inches) (missing for 2-seater vehicles)
- **Luggage.room**: Luggage capacity (cubic feet) (missing for vans)
- **Weight**: Weight (pounds)
- **Origin**: Of non-USA or USA company origins? (factor)
- **Make**: Combination of Manufacturer and Model (character)
Details

Cars were selected at random from among 1993 passenger car models that were listed in both the Consumer Reports issue and the PACE Buying Guide. Pickup trucks and Sport/Utility vehicles were eliminated due to incomplete information in the Consumer Reports source. Duplicate models (e.g., Dodge Shadow and Plymouth Sundance) were listed at most once.

Further description can be found in Lock (1993). Use the URL http://www.amstat.org/publications/jse/v1n1/datasets.lock.html

Source


References


---

Cushings

Diagnostic Tests on Patients with Cushing’s Syndrome

Description

Cushing’s syndrome is a hypertensive disorder associated with over-secretion of cortisol by the adrenal gland. The observations are urinary excretion rates of two steroid metabolites.

Usage

Cushings

Format

The Cushings data frame has 27 rows and 3 columns:

**Tetrahydrocortisone** urinary excretion rate (mg/24hr) of Tetrahydrocortisone.

**Pregnanetriol** urinary excretion rate (mg/24hr) of Pregnanetriol.

**Type** underlying type of syndrome, coded a (adenoma), b (bilateral hyperplasia), c (carcinoma) or u for unknown.

Source


References

DDT

**Description**

A numeric vector of 15 measurements by different laboratories of the pesticide DDT in kale, in ppm (parts per million) using the multiple pesticide residue measurement.

**Usage**

DDT

**Source**


GAGurine

**Description**

Data were collected on the concentration of a chemical GAG in the urine of 314 children aged from zero to seventeen years. The aim of the study was to produce a chart to help a paediatrican to assess if a child’s GAG concentration is ‘normal’.

**Usage**

GAGurine

**Format**

This data frame contains the following columns:

- **Age**  age of child in years
- **GAG**  concentration of GAG (the units have been lost)

**Source**

Mrs Susan Prosser, Paediatrics Department, University of Oxford, via Department of Statistics Consulting Service.

**References**

Description

The data given in data frame `Insurance` consist of the numbers of policyholders of an insurance company who were exposed to risk, and the numbers of car insurance claims made by those policyholders in the third quarter of 1973.

Usage

`Insurance`

Format

This data frame contains the following columns:

- **District**: district of policyholder (1 to 4): 4 is major cities.
- **Group**: group of car (1 to 4), <1 litre, 1–1.5 litre, 1.5–2 litre, >2 litre.
- **Age**: of driver in 4 ordered groups, <25, 25–29, 30–35, >35.
- **Holders**: numbers of policyholders
- **Claims**: numbers of claims

Source


References


Examples

```r
## main-effects fit as Poisson GLM with offset
glm(Claims ~ District + Group + Age + offset(log(Holders)),
   data = Insurance, family = poisson)

# same via loglm
loglm(Claims ~ District + Group + Age + offset(log(Holders)),
   data = Insurance)
```
**MASS-internal  Internal MASS functions**

**Description**
Internal MASS functions

**Usage**

- `enlist(vec)`
- `fbeta(x, alpha, beta)`
- `frequency.polygon(x, nclass = nclass.freq(x), xlab="", ylab="", ...)`
- `nclass.freq(x)`
- `neg.bin(theta = stop("'theta' must be given"))`
- `negexp.SSival(mCall, data, LHS)`

**Details**
These are not intended to be called by the user. Some are for compatibility with earlier versions of MASS (the book).

---

**Melanoma  Survival from Malignant Melanoma**

**Description**
The **Melanoma** data frame has data on 205 patients in Denmark with malignant melanoma.

**Usage**

**Melanoma**

**Format**
This data frame contains the following columns:

- `time` survival time in days, possibly censored
- `status` 1 died from melanoma, 2 alive, 3 dead from other causes
- `sex` 1 = male, 0 = female
- `age` age in years
- `year` of operation
- `thickness` tumour thickness in mm
- `ulcer` 1 = presence, 0 = absence
Null

Null Spaces of Matrices

Description
Given a matrix, M, find a matrix N giving a basis for the null space. That is t(N) %*% M is the zero and N has the maximum number of linearly independent columns.

Usage
Null(M)

Arguments
M Input matrix. A vector is coerced to a 1-column matrix.

Value
The matrix N with the basis for the null space, or an empty vector if the matrix M is square and of maximal rank.

References

See Also
qr, qr.Q

Examples
# The function is currently defined as
function(M)
{
  tmp <- qr(M)
  set <- if(tmp$rank == 0) 1:ncol(M) else - (1:tmp$rank)
  qr.Q(tmp, complete = TRUE)[, set, drop = FALSE]
}
Tests of Auditory Perception in Children with OME

Description

Experiments were performed on children on their ability to differentiate a signal in broad-band noise. The noise was played from a pair of speakers and a signal was added to just one channel; the subject had to turn his/her head to the channel with the added signal. The signal was either coherent (the amplitude of the noise was increased for a period) or incoherent (independent noise was added for the same period to form the same increase in power).

The threshold used in the original analysis was the stimulus loudness needs to get 75% correct responses. Some of the children had suffered from otitis media with effusion (OME).

Usage

OME

Format

The OME data frame has 1129 rows and 7 columns:

- **ID** Subject ID (1 to 99, with some IDs missing). A few subjects were measured at different ages.
- **OME** "low" or "high" or "N/A" (at ages other than 30 and 60 months).
- **Age** Age of the subject (months).
- **Loud** Loudness of stimulus, in decibels.
- **Noise** Whether the signal in the stimulus was "coherent" or "incoherent".
- **Correct** Number of correct responses from Trials trials.
- **Trials** Number of trials performed.

Background

The experiment was to study otitis media with effusion (OME), a very common childhood condition where the middle ear space, which is normally air-filled, becomes congested by a fluid. There is a concomitant fluctuating, conductive hearing loss which can result in various language, cognitive and social deficits. The term ‘binaural hearing’ is used to describe the listening conditions in which the brain is processing information from both ears at the same time. The brain computes differences in the intensity and/or timing of signals arriving at each ear which contributes to sound localisation and also to our ability to hear in background noise.

Some years ago, it was found that children of 7-8 years with a history of significant OME had significantly worse binaural hearing than children without such a history, despite having equivalent sensitivity. The question remained as to whether it was the timing, the duration, or the degree of severity of the otitis media episodes during critical periods, which affected later binaural hearing. In an attempt to begin to answer this question, 95 children were monitored for the presence of effusion every month since birth. On the basis of OME experience in their first two years, the test population was split into one group of high OME prevalence and one of low prevalence.
Examples

# Fit logistic curve from p = 0.5 to p = 1.0
fp1 <- deriv(~ 0.5 + 0.5/(1 + exp(-(x-L75)/scal)),
    c("L75", "scal"),
    function(x, L75, scal) NULL)
nls(Correct/Trials ~ fp1(Loud, L75, scal), data = OME,
    start = c(L75=45, scal=3))

nls(Correct/Trials ~ fp1(Loud, L75, scal),
    data = OME[OME$Noise == "coherent",]
    start = c(L75=45, scal=3))

nls(Correct/Trials ~ fp1(Loud, L75, scal),
    data = OME[OME$Noise == "incoherent",]
    start = c(L75=45, scal=3))

# individual fits for each experiment
aa <- factor(OME$Age)
ab <- 10*OME$ID + unclass(aa)
ac <- unclass(factor(ab))
OME$UID <- as.vector(ac)
OME$UIDn <- OME$UID + 0.1*(OME$Noise == "incoherent")
rm(aa, ab, ac)
OMEi <- OME

library(nlme)
fp2 <- deriv(~ 0.5 + 0.5/(1 + exp(-(x-L75)/2)),
    "L75", function(x, L75) NULL)
dec <- getOption("OutDec")
options(show.error.messages = FALSE, OutDec=".")
OMEi.nls <- nlsList(Correct/Trials ~ fp2(Loud, L75) | UIDn,
    data = OMEi, start = list(L75=45), control = list(maxiter=100))
options(show.error.messages = TRUE, OutDec=dec)
tmp <- sapply(OMEi.nls, function(X)
    {if(is.null(X)) NA else as.vector(coef(X))})
OMEif <- data.frame(UID = round(as.numeric(names(tmp))),
    Noise = rep(c("coherent", "incoherent"), 110),
    L75 = as.vector(tmp), stringsAsFactors = TRUE)
OMEif$Age <- OME$Age[match(OMEif$UID, OME$UID)]
OMEif$OME <- OME$OME[match(OMEif$UID, OME$UID)]
OMEif <- OMEif[OMEif$L75 > 30,]
summary(lm(L75 ~ Noise/Age, data = OMEif, na.action = na.omit))
summary(lm(L75 ~ Noise/(Age + OME),
    data = OMEif, subset = (Age >= 30 & Age <= 60),
    na.action = na.omit), cor = FALSE)

# Or fit by weighted least squares
fpl75 <- deriv(~ sqrt(n)*(r/n - 0.5 - 0.5/(1 + exp(-(x-L75)/scal))),
    c("L75", "scal"),
    function(x, scal) NULL)
nls(0 ~ fp175(Correct, Trials, Loud, L75, scal),
data = OME[OME$Noise == "coherent",],
start = c(L75=45, scal=3))
nls(0 ~ fp175(Correct, Trials, Loud, L75, scal),
data = OME[OME$Noise == "incoherent",],
start = c(L75=45, scal=3))

# Test to see if the curves shift with age
fp175age <- deriv(~ sqrt(n) *(r/n - 0.5 - 0.5/(1 +
expt(-(x-L75-slope*age)/scal))),
c("L75", "slope", "scal"),
function(r, n, x, age, L75, slope, scal) NULL)
OME.nls1 <-
nls(0 ~ fp175age(Correct, Trials, Loud, Age, L75, slope, scal),
data = OME[OME$Noise == "coherent",],
start = c(L75=45, slope=0, scal=2))
sqrt(diag(vcov(OME.nls1)))

OME.nls2 <-
nls(0 ~ fp175age(Correct, Trials, Loud, Age, L75, slope, scal),
data = OME[OME$Noise == "incoherent",],
start = c(L75=45, slope=0, scal=2))
sqrt(diag(vcov(OME.nls2)))

# Now allow random effects by using NLME
OMEf <- OME[rep(1:nrow(OME), OME$Trials),]
attach(OME)
OMEf$Resp <- rep(rep(c(1,0), length(Trials)),
t(cbind(Correct, Trials-Correct)))
OMEf <- OMEf[, -match(c("Correct", "Trials"), names(OMEf))]
detach("OME")

## Not run: ## this fails in R on some platforms
fp2 <- deriv(~ 0.5 + 0.5/(1 + exp(-(x-L75)/exp(lsc))),
c("L75", "lsc"),
function(x, L75, lsc) NULL)
G1.nlme <- nlme(Resp ~ fp2(Loud, L75, lsc),
fixed = list(L75 ~ Age, lsc ~ 1),
random = L75 + lsc ~ 1 | UID,
data = OMEf[OMEf$Noise == "coherent",], method = "ML",
start = list(fixed=c(L75=48.7, -0.03), lsc=0.24)),
verbose = TRUE)
summary(G1.nlme)

G2.nlme <- nlme(Resp ~ fp2(Loud, L75, lsc),
fixed = list(L75 ~ Age, lsc ~ 1),
random = L75 + lsc ~ 1 | UID,
data = OMEf[OMEf$Noise == "incoherent",], method="ML",
start = list(fixed=c(L75=c(41.5, -0.1), lsc=0)),
verbose = TRUE)
summary(G2.nlme)

## End(Not run)
Description

A population of women who were at least 21 years old, of Pima Indian heritage and living near Phoenix, Arizona, was tested for diabetes according to World Health Organization criteria. The data were collected by the US National Institute of Diabetes and Digestive and Kidney Diseases. We used the 532 complete records after dropping the (mainly missing) data on serum insulin.

Usage

Pima.tr
Pima.tr2
Pima.te

Format

These data frames contains the following columns:

- **npreg** number of pregnancies
- **glu** plasma glucose concentration in an oral glucose tolerance test
- **bp** diastolic blood pressure (mm Hg)
- **skin** triceps skin fold thickness (mm)
- **bmi** body mass index (weight in kg/(height in m)^2)
- **ped** diabetes pedigree function
- **age** age in years
- **type** Yes or No, for diabetic according to WHO criteria

Details

The training set Pima.tr contains a randomly selected set of 200 subjects, and Pima.te contains the remaining 322 subjects. Pima.tr2 contains Pima.tr plus 100 subjects with missing values in the explanatory variables.

Source


Blood Pressure in Rabbits

Description

Five rabbits were studied on two occasions, after treatment with saline (control) and after treatment with the $5-HT_3$ antagonist MDL 72222. After each treatment ascending doses of phenylbiguanide were injected intravenously at 10 minute intervals and the responses of mean blood pressure measured. The goal was to test whether the cardiogenic chemoreflex elicited by phenylbiguanide depends on the activation of $5-HT_3$ receptors.

Usage

Rabbit

Format

This data frame contains 60 rows and the following variables:

- **BPchange**: change in blood pressure relative to the start of the experiment
- **Dose**: dose of Phenylbiguanide in micrograms
- **Run**: label of run ("C1" to "C5", then "M1" to "M5")
- **Treatment**: placebo or the $5-HT_3$ antagonist MDL 72222
- **Animal**: label of animal used ("R1" to "R5")

Source


[The numerical data are not in the paper but were supplied by Professor Ludbrook]

References


Accelerated Testing of Tyre Rubber

Description

Data frame from accelerated testing of tyre rubber.

Usage

Rubber
SP500

Format

- \textbf{loss} the abrasion loss in gm/hr.
- \textbf{hard} the hardness in Shore units.
- \textbf{tens} tensile strength in kg/sq m.

Source

O.L. Davies (1947) \textit{Statistical Methods in Research and Production}. Oliver and Boyd, Table 6.1 p. 119.


References


---

\textbf{SP500} \hspace{1cm} \textit{Returns of the Standard and Poors 500}

Description

Returns of the Standard and Poors 500 Index in the 1990's

Usage

\texttt{SP500}

Format


References

Description

The Sitka data frame has 395 rows and 4 columns. It gives repeated measurements on the log-size of 79 Sitka spruce trees, 54 of which were grown in ozone-enriched chambers and 25 were controls. The size was measured five times in 1988, at roughly monthly intervals.

Usage

Sitka

Format

This data frame contains the following columns:

- **size**: measured size (height times diameter squared) of tree, on log scale
- **Time**: time of measurement in days since 1 January 1988
- **tree**: number of tree
- **treat**: either "ozone" for an ozone-enriched chamber or "control"

Source


References


See Also

Sitka89

---

Description

The Sitka89 data frame has 632 rows and 4 columns. It gives repeated measurements on the log-size of 79 Sitka spruce trees, 54 of which were grown in ozone-enriched chambers and 25 were controls. The size was measured eight times in 1989, at roughly monthly intervals.
Skye

Usage

Sitka89

Format

This data frame contains the following columns:

- **size**: measured size (height times diameter squared) of tree, on log scale
- **Time**: time of measurement in days since 1 January 1988
- **tree**: number of tree
- **treat**: either "ozone" for an ozone-enriched chamber or "control"

Source


See Also

Sitka

---

Skye  

*AFM Compositions of Aphyric Skye Lavas*

Description

The Skye data frame has 23 rows and 3 columns.

Usage

Skye

Format

This data frame contains the following columns:

- **A**: Percentage of sodium and potassium oxides
- **F**: Percentage of iron oxide
- **M**: Percentage of magnesium oxide

Source

References


Examples

```r
# ternary() is from the on-line answers.
ternary <- function(X, pch = par("pch"), lceX = 1,
   add = FALSE, ord = 1:3, ...)
{
  X <- as.matrix(X)
  if(any(X) < 0) stop("X must be non-negative")
  s <- drop(X %*% rep(1, ncol(X)))
  if(any(s<=0)) stop("each row of X must have a positive sum")
  if(max(abs(s-1)) > 1e-6) {
    warning("row(s) of X will be rescaled")
    X <- X / s
  }
  X <- X[, ord]
  s3 <- sqrt(1/3)
  if(!add)
  {
    oldpty <- par("pty")
    on.exit(par(pty=oldpty))
    par(pch="s")
    plot(c(-s3, s3), c(0.5-s3, 0.5+s3), type="n", axes=FALSE,
      xlab="", ylab="")
    polygon(c(0, -s3, s3), c(1, 0, 0), density=0)
    lab <- NULL
    if(!is.null(dn <- dimnames(X))) lab <- dn[[2]]
    if(length(lab) < 3) lab <- as.character(1:3)
    eps <- 0.05 * lceX
    text(c(0, s3+eps*0.7, -s3-eps*0.7),
      c(1+eps, -0.1*eps, -0.1*eps), lab, cex=lceX)
  }
  points((X[,2] - X[,3]) *s3, X[,1], ...)
}

ternary(Skye/100, ord=c(1,3,2))
```

Traffic

Effect of Swedish Speed Limits on Accidents

Description

An experiment was performed in Sweden in 1961-2 to assess the effect of a speed limit on the motorway accident rate. The experiment was conducted on 92 days in each year, matched so that day $j$ in 1962 was comparable to day $j$ in 1961. On some days the speed limit was in effect and enforced, while on other days there was no speed limit and cars tended to be driven faster. The speed limit days tended to be in contiguous blocks.
UScereal

Usage

Traffic

Format

This data frame contains the following columns:

- **year**: 1961 or 1962
- **day**: of year
- **limit**: was there a speed limit?
- **y**: traffic accident count for that day

Source


References


---

UScereal

*Nutritional and Marketing Information on US Cereals*

Description

The UScereal data frame has 65 rows and 11 columns. The data come from the 1993 ASA Statistical Graphics Exposition, and are taken from the mandatory F&DA food label. The data have been normalized here to a portion of one American cup.

Usage

UScereal

Format

This data frame contains the following columns:

- **mfr**: Manufacturer, represented by its first initial: G=General Mills, K=Kelloggs, N=Nabisco, P=Post, Q=Quaker Oats, R=Ralston Purina.
- **calories**: number of calories in one portion
- **protein**: grams of protein in one portion
- **fat**: grams of fat in one portion
- **sodium**: milligrams of sodium in one portion
fibre grams of dietary fibre in one portion

carbo grams of complex carbohydrates in one portion

sugars grams of sugars in one portion

shelf display shelf (1, 2, or 3, counting from the floor)

potassium grams of potassium

vitamins vitamins and minerals (none, enriched, or 100%)

Source

The original data are available at http://lib.stat.cmu.edu/datasets/1993.expo/.

References


Description

Criminologists are interested in the effect of punishment regimes on crime rates. This has been studied using aggregate data on 47 states of the USA for 1960 given in this data frame. The variables seem to have been re-scaled to convenient numbers.

Usage

UScrime

Format

This data frame contains the following columns:

M percentage of males aged 14-24

So indicator variable for a southern state

Ed mean years of schooling

Po1 police expenditure in 1960

Po2 police expenditure in 1959

LF labour force participation rate

M.F number of males per 1000 females

Pop state population

NW number of nonwhites per 1000 people

U1 unemployment rate of urban males 14-24
U2 unemployment rate of urban males 35-39
GDP gross domestic product per head
Ineq income inequality
Prob probability of imprisonment
Time average time served in state prisons
y rate of crimes in a particular category per head of population

Source

References

VA Veteran’s Administration Lung Cancer Trial

Description
Veteran’s Administration lung cancer trial from Kalbfleisch & Prentice.

Usage
VA

Format
A data frame with columns:

sttime survival or follow-up time in days.
status dead or censored.
treat standard or test
age patient’s age in years
Karn Karnofsky score of patient’s performance on a scale of 0 to 100.
diag.time times since diagnosis in months at entry to trial.
cell one of four cell types.
prior prior therapy?
### accdeaths

**Source**


**References**


### abbey

**Description**

A numeric vector of 31 determinations of nickel content (ppm) in a Canadian syenite rock.

**Usage**

abbey

**Source**


**References**


### accdeaths

**Description**

A regular time series giving the monthly totals of accidental deaths in the USA. The values for first six months of 1979 (p. 326) are 7798 7406 8363 8460 9217 9316

**Usage**

accdeaths

**Source**


**References**

Try All One-Term Additions to a Model

Description

Try fitting all models that differ from the current model by adding a single term from those supplied, maintaining marginality.

This function is generic; there exist methods for classes lm and glm and the default method will work for many other classes.

Usage

addterm(object, ...)

## Default S3 method:
addterm(object, scope, scale = 0, test = c("none", "Chisq"),
  k = 2, sorted = FALSE, trace = FALSE, ...)

## S3 method for class 'lm':
addterm(object, scope, scale = 0, test = c("none", "Chisq", "F"),
  k = 2, sorted = FALSE, ...)

## S3 method for class 'glm':
addterm(object, scope, scale = 0, test = c("none", "Chisq", "F"),
  k = 2, sorted = FALSE, trace = FALSE, ...)

Arguments

object An object fitted by some model-fitting function.

scope a formula specifying a maximal model which should include the current one. All additional terms in the maximal model with all marginal terms in the original model are tried.

scale used in the definition of the AIC statistic for selecting the models, currently only for lm, aov and glm models. Specifying scale asserts that the residual standard error or dispersion is known.

test should the results include a test statistic relative to the original model? The F test is only appropriate for lm and aov models, and perhaps for some over-dispersed glm models. The Chisq test can be an exact test (lm models with known scale) or a likelihood-ratio test depending on the method.

k the multiple of the number of degrees of freedom used for the penalty. Only k=2 gives the genuine AIC: k = log(n) is sometimes referred to as BIC or SBC.

sorted should the results be sorted on the value of AIC?

trace if TRUE additional information may be given on the fits as they are tried.

... arguments passed to or from other methods.
Details

The definition of AIC is only up to an additive constant: when appropriate (lm models with specified scale) the constant is taken to be that used in Mallows’ Cp statistic and the results are labelled accordingly.

Value

A table of class "anova" containing at least columns for the change in degrees of freedom and AIC (or Cp) for the models. Some methods will give further information, for example sums of squares, deviances, log-likelihoods and test statistics.

References


See Also
dropterm, stepAIC

Examples

quine.hi <- aov(log(Days + 2.5) ~ .^4, quine)
quine.lo <- aov(log(Days+2.5) ~ 1, quine)
addterm(quine.lo, quine.hi, test="F")

house.glm0 <- glm(Freq ~ Infl*Type*Cont + Sat, family=poisson, data=housing)
addterm(house.glm0, ~. + Sat:(Infl+Type+Cont), test="Chisq")
house.glm1 <- update(house.glm0, . ~ . + Sat*(Infl+Type+Cont))
addterm(house.glm1, ~. + Sat:(Infl+Type+Cont)^2, test = "Chisq")

anorexia

Anorexia Data on Weight Change

Description

The anorexia data frame has 72 rows and 3 columns. Weight change data for young female anorexia patients.

Usage

anorexia
anovay.negbin

Format

This data frame contains the following columns:

**Treat** Factor of three levels: Cont (Control), CBT (Cognitive Behavioural Treatment) and FT (Family treatment).

**Prewt** Weight of patient before study period, in lbs.

**Postwt** Weight of patient after study period, in lbs.

Source


(Note that the original source mistakenly says that weights are in kg.)

References


---

anovay.negbin

Likelihood Ratio Tests for Negative Binomial GLMs

Description

Method function to perform sequential likelihood ratio tests for Negative Binomial generalized linear models.

Usage

```r
## S3 method for class 'negbin':
anova(object, ..., test = "Chisq")
```

Arguments

- **object** Fitted model object of class "negbin", inheriting from classes "glm" and "lm", specifying a Negative Binomial fitted GLM. Typically the output of `glm.nb()`.

- **...** Zero or more additional fitted model objects of class "negbin". They should form a nested sequence of models, but need not be specified in any particular order.

- **test** Argument to match the `test` argument of `anova.glm`. Ignored (with a warning if changed) if a sequence of two or more Negative Binomial fitted model objects is specified, but possibly used if only one object is specified.
Details

This function is a method for the generic function `anova()` for class "negbin". It can be invoked by calling `anova(x)` for an object `x` of the appropriate class, or directly by calling `anova.negbin(x)` regardless of the class of the object.

Note

If only one fitted model object is specified, a sequential analysis of deviance table is given for the fitted model. The `theta` parameter is kept fixed. If more than one fitted model object is specified they must all be of class "negbin" and likelihood ratio tests are done of each model within the next. In this case `theta` is assumed to have been re-estimated for each model.

References


See Also

glm.nb, negative.binomial, summary.negbin

Examples

```r
m1 <- glm.nb(Days ~ Eth*Age*Lrn*Sex, quine, link = log)
m2 <- update(m1, . ~ . - Eth:Age:Lrn:Sex)
anova(m2, m1)
anova(m2)
```

area

**Adaptive Numerical Integration**

Description

Integrate a function of one variable over a finite range using a recursive adaptive method. This function is mainly for demonstration purposes.

Usage

```r
area(f, a, b, ..., fa = f(a, ...), fb = f(b, ...),
     limit = 10, eps = 1e-05)
```

Arguments

- `f` The integrand as an S function object. The variable of integration must be the first argument.
- `a` Lower limit of integration.
- `b` Upper limit of integration.
- `...` Additional arguments needed by the integrand.
fa  Function value at the lower limit.
fb  Function value at the upper limit.
limit  Limit on the depth to which recursion is allowed to go.
eps  Error tolerance to control the process.

Details

The method divides the interval in two and compares the values given by Simpson’s rule and the trapezium rule. If these are within \( \text{eps} \) of each other the Simpson’s rule result is given, otherwise the process is applied separately to each half of the interval and the results added together.

Value

The integral from \( a \) to \( b \) of \( f(x) \).

References


Examples

\[
\text{area} (\sin, 0, \pi) \quad \# \text{ integrate the \( \sin \) function from 0 to \( \pi \).}
\]

---

**bacteria**

*Presence of Bacteria after Drug Treatments*

Description

Tests of the presence of the bacteria \( H. \text{influenzae} \) in children with otitis media in the Northern Territory of Australia.

Usage

`bacteria`

Format

This data frame has 220 rows and the following columns:

- **y**  presence or absence: a factor with levels `n` and `y`.
- **ap**  active/placebo: a factor with levels `a` and `p`.
- **hilo**  hi/low compliance: a factor with levels `hi` amd `lo`.
- **week** numeric: week of test.
- **ID**  subject ID: a factor.
- **trt**  a factor with levels `placebo`, `drug` and `drug+`, a re-coding of `ap` and `hilo`.
Details

Dr A. Leach tested the effects of a drug on 50 children with a history of otitis media in the Northern Territory of Australia. The children were randomized to the drug or the a placebo, and also to receive active encouragement to comply with taking the drug.

The presence of *H. influenzae* was checked at weeks 0, 2, 4, 6 and 11: 30 of the checks were missing and are not included in this data frame.

Source


References


Examples

```r
contrasts(bacteria$trt) <- structure(contr.sdif(3),
       dimnames = list(NULL, c("drug", "encourage")))
## fixed effects analyses
summary(glm(y ~ trt * week, binomial, data = bacteria))
summary(glm(y ~ trt + week, binomial, data = bacteria))
summary(glm(y ~ trt + I(week > 2), binomial, data = bacteria))

# conditional random-effects analysis
library(survival)
bacteria$Time <- rep(1, nrow(bacteria))
coxph(Surv(Time, unclass(y)) ~ week + strata(ID),
     data = bacteria, method = "exact")
coxph(Surv(Time, unclass(y)) ~ factor(week) + strata(ID),
     data = bacteria, method = "exact")
coxph(Surv(Time, unclass(y)) ~ I(week > 2) + strata(ID),
     data = bacteria, method = "exact")

# PQL glmm analysis
library(nlme)
summary(glmmPQL(y ~ trt + I(week > 2), random = ~ 1 | ID,
                family = binomial, data = bacteria))
```

bandwidth.nrd

*Bandwidth for density() via Normal Reference Distribution*

Description

A well-supported rule-of-thumb for choosing the bandwidth of a Gaussian kernel density estimator.
bcv

Usage

bandwidth.nrd(x)

Arguments

x A data vector.

Value

A bandwidth on a scale suitable for the width argument of density.

References


Examples

# The function is currently defined as
function(x)
{
  r <- quantile(x, c(0.25, 0.75))
  h <- (r[2] - r[1])/1.34
  4 * 1.06 * min(sqrt(var(x)), h) * length(x)^(-1/5)
}

bcv Biased Cross-Validation for Bandwidth Selection

Description

Uses biased cross-validation to select the bandwidth of a Gaussian kernel density estimator.

Usage

bcv(x, nb = 1000, lower, upper)

Arguments

x a numeric vector

nb number of bins to use.

lower, upper Range over which to minimize. The default is almost always satisfactory.

Value

a bandwidth
References

See Also
ucv, width.SJ, density

Examples
bcv(geyser$duration)

---

beav1  Body Temperature Series of Beaver 1

Description
Reynolds (1994) describes a small part of a study of the long-term temperature dynamics of beaver \textit{Castor canadensis} in north-central Wisconsin. Body temperature was measured by telemetry every 10 minutes for four females, but data from a one period of less than a day for each of two animals is used there.

Usage
beav1

Format
The beav1 data frame has 114 rows and 4 columns. This data frame contains the following columns:

- **day**  Day of observation (in days since the beginning of 1990), December 12-13.
- **time**  Time of observation, in the form 0330 for 3.30am
- **temp**  Measured body temperature in degrees Celcius
- **activ**  Indicator of activity outside the retreat

Note
The observation at 22:20 is missing.

Source
References


See Also

beav2

Examples

attach(beav1)
beav1$hours <- 24*(day - 346) + trunc(time/100) + (time%%100)/60
plot(beav1$hours, beav1$temp, type="l", xlab="time",
ylab="temperature", main="Beaver 1")
usr <- par("usr"); usr[3:4] <- c(-0.2, 8); par(usr=usr)
lines(beav1$hours, beav1$activ, type="s", lty=2)
temp <- ts(c(beav1$temp[1:82], NA, beav1$temp[83:114]),
start = 9.5, frequency = 6)
activ <- ts(c(beav1$activ[1:82], NA, beav1$activ[83:114]),
start = 9.5, frequency = 6)

acf(temp[1:53])
acf(temp[1:53], type = "partial")
ar(temp[1:53])
act <- c(rep(0, 10), activ)
X <- cbind(1, act = act[11:125], act1 = act[10:124],
act2 = act[9:123], act3 = act[8:122])
alpha <- 0.80
stemp <- as.vector(temp - alpha*lag(temp, -1))
sX <- X[-1, ] - alpha * X[-115, ]
beav1.ls <- lm(stemp ~ -1 + sX, na.action = na.omit)
summary(beav1.ls, cor = FALSE)
detach("beav1"); rm(temp, activ)

beav2

---

**Body Temperature Series of Beaver 2**

Description

Reynolds (1994) describes a small part of a study of the long-term temperature dynamics of beaver *Castor canadensis* in north-central Wisconsin. Body temperature was measured by telemetry every 10 minutes for four females, but data from a one period of less than a day for each of two animals is used there.

Usage

beav2
Format

The beav2 data frame has 100 rows and 4 columns. This data frame contains the following columns:

**day**  Day of observation (in days since the beginning of 1990), November 3-4.
**time**  Time of observation, in the form 0330 for 3.30am
**temp**  Measured body temperature in degrees Celcius
**activ** Indicator of activity outside the retreat

Source


References


See Also

beav1

Examples

```r
attach(beav2)
beav2$hours <- 24*(day-307) + trunc(time/100) + (time%%100)/60
plot(beav2$hours, beav2$temp, type = "l", xlab = "time",
     ylab = "temperature", main = "Beaver 2")
usr <- par("usr"); usr[3:4] <- c(-0.2, 8); par(usr = usr)
lines(beav2$hours, beav2$activ, type = "s", lty = 2)

temp <- ts(temp, start = 8+2/3, frequency = 6)
activ <- ts(activ, start = 8+2/3, frequency = 6)
acf(temp[activ == 0]); acf(temp[activ == 1]) # also look at PACFs
ar(temp[activ == 0]); ar(temp[activ == 1])

arima(temp, order = c(1,0,0), xreg = activ)
dreg <- cbind(sin = sin(2*pi*beav2$hours/24), cos = cos(2*pi*beav2$hours/24))
arima(temp, order = c(1,0,0), xreg = cbind(activ=activ, dreg))

library(nlme)
beav2.gls <- gls(temp ~ activ, data = beav2, corr = corAR1(0.8),
                 method = "ML")
summary(beav2.gls)
summary(update(beav2.gls, subset = 6:100))
detach("beav2"); rm(temp, activ)
```
Biopsy Data on Breast Cancer Patients

Description

This breast cancer database was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg. He assessed biopsies of breast tumours for 699 patients up to 15 July 1992; each of nine attributes has been scored on a scale of 1 to 10, and the outcome is also known. There are 699 rows and 11 columns.

Usage

biopsy

Format

This data frame contains the following columns:

ID Sample code number (not unique)
V1 Clump thickness
V2 Uniformity of cell size
V3 Uniformity of cell shape
V4 Marginal adhesion
V5 Single epithelial cell size
V6 Bare nuclei (16 values are missing)
V7 Bland chromatin
V8 Normal nucleoli
V9 Mitoses
class "benign" or "malignant"

Source


### References


---

**birthwt**

*Risk Factors Associated with Low Infant Birth Weight*

### Description

The *birthwt* data frame has 189 rows and 10 columns. The data were collected at Baystate Medical Center, Springfield, Mass during 1986.

### Usage

`birthwt`

### Format

This data frame contains the following columns:

- **low** indicator of birth weight less than 2.5kg
- **age** mother’s age in years
- **lwt** mother’s weight in pounds at last menstrual period
- **race** mother’s race (1 = white, 2 = black, 3 = other)
- **smoke** smoking status during pregnancy
- **ptl** number of previous premature labours
- **ht** history of hypertension
- **ui** presence of uterine irritability
- **ftv** number of physician visits during the first trimester
- **bwt** birth weight in grams

### Source


### References

Examples

attach(birthwt)
race <- factor(race, labels = c("white", "black", "other"))
ptd <- factor(ptl > 0)
ftv <- factor(ftv)
levels(ftv)[c(1:2)] <- "2+
bwt <- data.frame(low = factor(low), age, lwt, race,
  smoke = (smoke > 0), ptd, ht = (ht > 0), ui = (ui > 0), ftv)
detach("birthwt")
options(contrasts = c("contr.treatment", "contr.poly"))
glm(low ~ ., binomial, bwt)

detach("birthwt")

boxcox  Box-Cox Transformations for Linear Models

Description

Computes and optionally plots profile log-likelihoods for the parameter of the Box-Cox power transformation.

Usage

boxcox(object, lambda = seq(-2, 2, length.out = 10), plotit = TRUE,
  interp, eps = 1/50, xlab = expression(lambda),
  ylab = "log-Likelihood", ...)  

Arguments

object a formula or fitted model object. Currently only \texttt{lm} and \texttt{aov} objects are handled.
lambda vector of values of \texttt{lambda} – default \(-2, 2\) in steps of 0.1.
plotit logical which controls whether the result should be plotted.
interp logical which controls whether spline interpolation is used. Default to \texttt{TRUE} if plotting with \texttt{lambda} of length less than 100.
eps  
Tolerance for lambda = 0; defaults to 0.02.

xlab  
defaults to "lambda".

ylab  
defaults to "log-Likelihood".

...  
additional parameters to be used in the model fitting.

Value

A list of the lambda vector and the computed profile log-likelihood vector, invisibly if the result is plotted.

Side Effects

If plotit = TRUE plots loglik vs lambda and indicates a 95% confidence interval about the maximum observed value of lambda. If interp = TRUE, spline interpolation is used to give a smoother plot.

References


Examples

data(trees)
boxcox(Volume ~ log(Height) + log(Girth), data = trees,
      lambda = seq(-0.25, 0.25, length = 10))

boxcox(Days+1 ~ Eth*Sex*Age*Lrn, data = quine,
      lambda = seq(-0.05, 0.45, len = 20))

cabbages  
*Data from a cabbage field trial*

Description

The cabbages data set has 60 observations and 4 variables

Usage

cabbages
caith

Format

This data frame contains the following columns:

- Cult: Factor giving the cultivar of the cabbage, two levels: c39 and c52.
- Date: Factor specifying one of three planting dates: d16, d20 or d21.
- HeadWt: Weight of the cabbage head, presumably in kg.
- VitC: Ascorbic acid content, in undefined units.

Source


References


---

craith

Colours of Eyes and Hair of People in Caithness

Description

Data on the cross-classification of people in Caithness, Scotland, by eye and hair colour. The region of the UK is particularly interesting as there is a mixture of people of Nordic, Celtic and Anglo-Saxon origin.

Usage
craith

Format

A 4 by 5 table with rows the eye colours (blue, light, medium, dark) and columns the hair colours (fair, red, medium, dark, black).

Source


References

Examples

corresp(caith)
dimnames(caith)[[2]] <- c("F", "R", "M", "D", "B")
par(mfcol=c(1,3))
plot(corresp(caith, nf=2)); title("symmetric")
plot(corresp(caith, nf=2), type="rows"); title("rows")
plot(corresp(caith, nf=2), type="col"); title("columns")
par(mfrow=c(1,1))

Anatomical Data from Domestic Cats

Description

The heart and body weights of samples of male and female cats used for digitalis experiments. The cats were all adult, over 2 kg body weight.

Usage

cats

Format

This data frame contains the following columns:

Sex  Sex factor. Levels "F" and "M".
Bwt  Body weight in kg.
Hwt  Heart weight in g.

Source


References

**cement**

Heat Evolved by Setting Cements

**Description**

Experiment on the heat evolved in the setting of each of 13 cements.

**Usage**

`cement`

**Format**

`x1, x2, x3, x4` Proportions (%) of active ingredients  
`y` heat evolved in cals/gm

**Details**

13 samples of Portland cement were set. For each sample, the percentages of the four main chemical ingredients was accurately measured. While the cement was setting the amount of heat evolved was also measured.

**Source**


**References**


**Examples**

`lm(y ~ x1 + x2 + x3 + x4, cement)`

---

**chem**

Copper in Wholemeal Flour

**Description**

A numeric vector of 24 determinations of copper in wholemeal flour, in parts per million.

**Usage**

`chem`
Source


References


---

**con2tr**

*Convert Lists to Data Frames for use by Trellis*

**Description**

Convert lists to data frames for use by Trellis.

**Usage**

`con2tr(obj)`

**Arguments**

- `obj` A list of components `x`, `y` and `z` as passed to `contour`

**Details**

`con2tr` repeats the `x` and `y` components suitably to match the vector `z`.

**Value**

A data frame suitable for passing to Trellis functions.

**References**

Confidence Intervals for Model Parameters

Description
Computes confidence intervals for one or more parameters in a fitted model. Package MASS adds methods for glm and nls fits.

Usage
```r
## S3 method for class 'glm':
confint(object, parm, level = 0.95, trace = FALSE, ...)

## S3 method for class 'nls':
confint(object, parm, level = 0.95, ...)
```

Arguments
- `object` a fitted model object. Methods currently exist for the classes "glm", "nls" and for profile objects from these classes.
- `parm` a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
- `level` the confidence level required.
- `trace` logical. Should profiling be traced?
- `...` additional argument(s) for methods.

Details
`confint` is a generic function in package base. These `confint` methods calls the appropriate profile method, then finds the confidence intervals by interpolation in the profile traces. If the profile object is already available it should be used as the main argument rather than the fitted model object itself.

Value
A matrix (or vector) with columns giving lower and upper confidence limits for each parameter. These will be labelled as (1-level)/2 and 1 - (1-level)/2 in % (by default 2.5% and 97.5%).

References

See Also
profile
Examples

```r
expn1 <- deriv(y ~ b0 + b1 * 2^(-x/th), c("b0", "b1", "th"),
              function(b0, b1, th, x) { })

wtloss.gr <- nls(Weight ~ expn1(b0, b1, th, Days),
                 data = wtloss, start = c(b0=90, b1=95, th=120))

expn2 <- deriv(-b0 + b1*((w0 - b0)/b1)^(x/d0),
               c("b0","b1","d0"), function(b0, b1, d0, x, w0) { })

wtloss.init <- function(obj, w0) {
  p <- coef(obj)
  d0 <- - log((w0 - p["b0"]) / p["b1"]) / log(2) * p["th"]
  c(p[c("b0", "b1")], d0 = as.vector(d0))
}

out <- NULL
w0s <- c(110, 100, 90)
for(w0 in w0s) {
  fm <- nls(Weight ~ expn2(b0, b1, d0, Days, w0),
             wtloss, start = wtloss.init(wtloss.gr, w0))
  out <- rbind(out, c(coef(fm)["d0"], confint(fm, "d0")))
}

out <- NULL
w0s <- c(110, 100, 90)
for(w0 in w0s) {
  fm <- nls(Weight ~ expn2(b0, b1, d0, Days, w0),
             wtloss, start = wtloss.init(wtloss.gr, w0))
  out <- rbind(out, c(coef(fm)["d0"], confint(fm, "d0")))
}

dimnames(out) <- list(paste(w0s, "kg:"), c("d0", "low", "high"))
out
```

ldose <- rep(0:5, 2)
umdead <- c(1, 4, 9, 13, 18, 20, 0, 2, 6, 10, 12, 16)
sex <- factor(rep(c("M", "F"), c(6, 6)))
SF <- cbind(numdead, numalive = 20 - numdead)
budworm.lg0 <- glm(SF ~ sex + ldose - 1, family = binomial)
confint(budworm.lg0)
confint(budworm.lg0, "ldose")

---

**contr.sdif**

### Successive Differences contrast coding

#### Description
A coding for unordered factors based on successive differences.

#### Usage
```r
contr.sdif(n, contrasts = TRUE)
```

#### Arguments
- **n**: The number of levels required.
- **contrasts**: Should there be \(n - 1\) columns orthogonal to the mean (the default) or \(n\) columns spanning the space.
Details

The contrast coefficients are chosen so that the coded coefficients in a one-way layout are the differences between the means of the second and first levels, the third and second levels, and so on.

Value

If contrasts is TRUE, a matrix with n rows and n – 1 columns, and the n by n identity matrix if contrasts is FALSE.

References


See Also

contr.treatment, contr.sum, contr.helmert

Examples

contr.sdif(6)

## coop  Co-operative Trial in Analytical Chemistry

Description

Seven specimens were sent to 6 laboratories in 3 separate batches and each analysed for Analyte. Each analysis was duplicated.

Usage

coop

Format

This data frame contains the following columns:

- **Lab**: Laboratory, L1, L2, ..., L6.
- **Spc**: Specimen, S1, S2, ..., S7.
- **Bat**: Batch, B1, B2, B3 (nested within Spc/Lab).
- **Conc**: Concentration of Analyte in g/kg.

Source

References


See Also

chem, abbey.

corresp

Simple Correspondence Analysis

Description

Find the principal canonical correlation and corresponding row- and column-scores from a correspondence analysis of a two-way contingency table.

Usage

```r
corresp(x, ...)  
# S3 method for class 'matrix':
corresp(x, nf = 1, ...)
# S3 method for class 'factor':
corresp(x, y, ...)
# S3 method for class 'data.frame':
corresp(x, ...)
# S3 method for class 'xtabs':
corresp(x, ...)
# S3 method for class 'formula':
corresp(formula, data, ...)
```

Arguments

- `x, formula` The function is generic, accepting various forms of the principal argument for specifying a two-way frequency table. Currently accepted forms are matrices, data frames (coerced to frequency tables), objects of class "xtabs" and formulae of the form ~ F1 + F2, where F1 and F2 are factors.
- `nf` The number of factors to be computed. Note that although 1 is the most usual, one school of thought takes the first two singular vectors for a sort of biplot.
- `y` a second factor for a cross-classification
- `data` a data frame against which to preferentially resolve variables in the formula.
- `...` If the principal argument is a formula, a data frame may be specified as well from which variables in the formula are preferentially satisfied.
Details
See Venables & Ripley (2002). The plot method produces a graphical representation of the table if nf=1, with the areas of circles representing the numbers of points. If nf is two or more the biplot method is called, which plots the second and third columns of the matrices $A = D r^{-1/2} U L$ and $B = D c^{-1/2} V L$ where the singular value decomposition is $U L V$. Thus the x-axis is the canonical correlation times the row and column scores. Although this is called a biplot, it does not have any useful inner product relationship between the row and column scores. Think of this as an equally-scaled plot with two unrelated sets of labels. The origin is marked on the plot with a cross. (For other versions of this plot see the book.)

Value
An list object of class "correspondence" for which print, plot and biplot methods are supplied. The main components are the canonical correlation(s) and the row and column scores.

References

See Also
svd, princomp

Examples
(ct <- corresp(~ Age + Eth, data = quine))
## Not run: plot(ct)
corresp(caith)
biplot(corresp(caith, nf = 2))

cov.rob

Resistant Estimation of Multivariate Location and Scatter

Description
Compute a multivariate location and scale estimate with a high breakdown point – this can be thought of as estimating the mean and covariance of the good part of the data. cov.mve and cov.mcd are compatibility wrappers.

Usage
cov.rob(x, cor = FALSE, quantile.used = floor((n + p + 1)/2),
       method = c("mve", "mcd", "classical"),
       nsamp = "best", seed)
cov.mve(...)
cov.mcd(...)

Arguments

**x**
a matrix or data frame.

**cor**
should the returned result include a correlation matrix?

**quantile.used**
the minimum number of the data points regarded as good points.

**method**
the method to be used – minimum volume ellipsoid, minimum covariance determinant or classical product-moment. Using `cov.mve` or `cov.mcd` forces `mve` or `mcd` respectively.

**nsamp**
the number of samples or "best" or "exact" or "sample". If "sample" the number chosen is \(\min(5 \times p, 3000)\), taken from Rousseeuw and Hubert (1997). If "best" exhaustive enumeration is done up to 5000 samples: if "exact" exhaustive enumeration will be attempted however many samples are needed.

**seed**
the seed to be used for random sampling: see `RNGkind`. The current value of `.Random.seed` will be preserved if it is set.

**...**
arguments to `cov.rob` other than `method`.

Details

For method "mve", an approximate search is made of a subset of size `quantile.used` with an enclosing ellipsoid of smallest volume; in method "mcd" it is the volume of the Gaussian confidence ellipsoid, equivalently the determinant of the classical covariance matrix, that is minimized. The mean of the subset provides a first estimate of the location, and the rescaled covariance matrix a first estimate of scatter. The Mahalanobis distances of all the points from the location estimate for this covariance matrix are calculated, and those points within the 97.5% point under Gaussian assumptions are declared to be good. The final estimates are the mean and rescaled covariance of the good points.

The rescaling is by the appropriate percentile under Gaussian data; in addition the first covariance matrix has an ad hoc finite-sample correction given by Marazzi.

For method "mve" the search is made over ellipsoids determined by the covariance matrix of \(p\) of the data points. For method "mcd" an additional improvement step suggested by Rousseeuw and van Driessen (1999) is used, in which once a subset of size `quantile.used` is selected, an ellipsoid based on its covariance is tested (as this will have no larger a determinant, and may be smaller).

Value

A list with components

- **center**
  the final estimate of location.

- **cov**
  the final estimate of scatter.

- **cor**
  (only is `cor = TRUE`) the estimate of the correlation matrix.

- **sing**
  message giving number of singular samples out of total

- **crit**
  the value of the criterion on log scale. For MCD this is the determinant, and for MVE it is proportional to the volume.

- **best**
  the subset used. For MVE the best sample, for MCD the best set of size `quantile.used`.

- **n.obs**
  total number of observations.
Author(s)
B.D. Ripley

References

See Also
lqs

Examples
data(stackloss)
set.seed(123)
cov.rob(stackloss)
cov.rob(stack.x, method = "mcd", nsamp = "exact")

cov.trob

Covariance Estimation for Multivariate t Distribution

Description
Estimates a covariance or correlation matrix assuming the data came from a multivariate t distribution: this provides some degree of robustness to outlier without giving a high breakdown point.

Usage
cov.trob(x, wt = rep(1, n), cor = FALSE, center = TRUE, nu = 5,
maxit = 25, tol = 0.01)

Arguments
x data matrix. Missing values (NAs) are not allowed.
wt A vector of weights for each case: these are treated as if the case i actually occurred wt[i] times.
cor Flag to choose between returning the correlation (cor = TRUE) or covariance (cor = FALSE) matrix.
center a logical value or a numeric vector providing the location about which the covariance is to be taken. If `center = FALSE`, no centering is done; if `center = TRUE` the MLE of the location vector is used.

nu ‘degrees of freedom’ for the multivariate t distribution. Must exceed 2 (so that the covariance matrix is finite).

maxit Maximum number of iterations in fitting.

tol Convergence tolerance for fitting.

Value

A list with the following components

- `cov` the fitted covariance matrix.
- `center` the estimated or specified location vector.
- `wt` the specified weights: only returned if the `wt` argument was given.
- `n.obs` the number of cases used in the fitting.
- `cor` the fitted correlation matrix: only returned if `cor = TRUE`.
- `call` The matched call.
- `iter` The number of iterations used.

References


See Also

cov, cov.wt, cov.mve

Examples

data(stackloss)
cov.trob(stackloss)
**Description**

A relative performance measure and characteristics of 209 CPUs.

**Usage**

`cpus`

**Format**

The components are:

- `name` Manufacturer and model
- `cyct` cycle time in nanoseconds
- `mmin` minimum main memory in kilobytes
- `mmax` maximum main memory in kilobytes
- `cach` cache size in kilobytes
- `chmin` minimum number of channels
- `chmax` maximum number of channels
- `perf` published performance on a benchmark mix relative to an IBM 370/158-3
- `estperf` estimated performance (by Ein-Dor & Feldmesser)

**Source**


**References**

Description

The crabs data frame has 200 rows and 8 columns, describing 5 morphological measurements on 50 crabs each of two colour forms and both sexes, of the species *Leptograpsus variegatus* collected at Fremantle, W. Australia.

Usage

`crabs`

Format

This data frame contains the following columns:

- **sp** species - "B" or "O" for blue or orange
- **sex** as it says
- **index** index 1:50 within each of the four groups
- **FL** frontal lobe size (mm)
- **RW** rear width (mm)
- **CL** carapace length (mm)
- **CW** carapace width (mm)
- **BD** body depth (mm)

Source


References

Description

A time series giving the monthly deaths from bronchitis, emphysema and asthma in the UK, 1974-1979, both sexes (deaths).

Usage

deaths

Source


References


See Also

This the same as dataset ldeaths.

denumerator

Transform an Allowable Formula for 'loglm' into one for 'terms'

Description

loglm allows dimension numbers to be used in place of names in the formula. denumerator modifies such a formula into one that terms can process.

Usage

denumerator(x)

Arguments

x  A formula conforming to the conventions of loglm, that is, it may allow dimension numbers to stand in for names when specifying a log-linear model.
Details

The model fitting function `loglm` fits log-linear models to frequency data using iterative proportional scaling. To specify the model the user must nominate the margins in the data that remain fixed under the log-linear model. It is convenient to allow the user to use dimension numbers, 1, 2, 3, ... for the first, second, third, ..., margins in a similar way to variable names. As the model formula has to be parsed by `terms`, which treats 1 in a special way and requires parsable variable names, these formulae have to be modified by giving genuine names for these margin, or dimension numbers. `denumerate` replaces these numbers with names of a special form, namely n is replaced by .vn. This allows `terms` to parse the formula in the usual way.

Value

A linear model formula like that presented, except that where dimension numbers, say n, have been used to specify fixed margins these are replaced by names of the form .vn which may be processed by `terms`.

See Also

`renumerate`

Examples

```r
denumerate(~(1+2+3)^3 + a/b)
## Not run: ~ (.v1 + .v2 + .v3)^3 + a/b
```

---

dose.p \hspace{1cm} Predict Doses for Binomial Assay model

Description

Calibrate binomial assays, generalizing the calculation of LD50.

Usage

dose.p(obj, cf = 1:2, p = 0.5)

Arguments

- `obj`: A fitted model object of class inheriting from "glm".
- `cf`: The terms in the coefficient vector giving the intercept and coefficient of (log-)dose.
- `p`: Probabilities at which to predict the dose needed.

Value

An object of class "glm.dose" giving the prediction (attribute "p" and standard error (attribute "SE") at each response probability.)
drivers

References


Examples

```r
ldose <- rep(0:5, 2)
numdead <- c(1, 4, 9, 13, 18, 20, 0, 2, 6, 10, 12, 16)
sex <- factor(rep(c("M", "F"), c(6, 6)))
SF <- cbind(numdead, numalive = 20 - numdead)
budworm.lg0 <- glm(SF ~ sex + ldose - 1, family = binomial)
dose.p(budworm.lg0, cf = c(1,3), p = 1:3/4)
dose.p(update(budworm.lg0, family = binomial(link=probit)),
       cf = c(1,3), p = 1:3/4)
```

---

**drivers**  
*Deaths of Car Drivers in Great Britain 1969-84*

Description

A regular time series giving the monthly totals of car drivers in Great Britain killed or seriously injured Jan 1969 to Dec 1984. Compulsory wearing of seat belts was introduced on 31 Jan 1983.

Usage

`drivers`

Source


References

Try All One-Term Deletions from a Model

Description

Try fitting all models that differ from the current model by dropping a single term, maintaining marginality.

This function is generic; there exist methods for classes `lm` and `glm` and the default method will work for many other classes.

Usage

```
dropterm (object, ...)  
```

## Default S3 method:
dropterm(object, scope, scale = 0, test = c("none", "Chisq"),
   k = 2, sorted = FALSE, trace = FALSE, ...)

## S3 method for class 'lm':
dropterm(object, scope, scale = 0, test = c("none", "Chisq", "F"),
   k = 2, sorted = FALSE, ...)  

## S3 method for class 'glm':
dropterm(object, scope, scale = 0, test = c("none", "Chisq", "F"),
   k = 2, sorted = FALSE, trace = FALSE, ...)

Arguments

- **object**: A object fitted by some model-fitting function.
- **scope**: a formula giving terms which might be dropped. By default, the model formula. Only terms that can be dropped and maintain marginality are actually tried.
- **scale**: used in the definition of the AIC statistic for selecting the models, currently only for `lm`, `aov` and `glm` models. Specifying `scale` asserts that the residual standard error or dispersion is known.
- **test**: should the results include a test statistic relative to the original model? The F test is only appropriate for `lm` and `aov` models, and perhaps for some over-dispersed `glm` models. The Chisq test can be an exact test (lm models with known scale) or a likelihood-ratio test depending on the method.
- **k**: the multiple of the number of degrees of freedom used for the penalty. Only \( k = 2 \) gives the genuine AIC: \( k = \log(n) \) is sometimes referred to as BIC or SBC.
- **sorted**: should the results be sorted on the value of AIC?
- **trace**: if `TRUE` additional information may be given on the fits as they are tried.
- **...**: arguments passed to or from other methods.
Details

The definition of AIC is only up to an additive constant: when appropriate (lm models with specified scale) the constant is taken to be that used in Mallows’ Cp statistic and the results are labelled accordingly.

Value

A table of class "anova" containing at least columns for the change in degrees of freedom and AIC (or Cp) for the models. Some methods will give further information, for example sums of squares, deviances, log-likelihoods and test statistics.

References


See Also

addterm, stepAIC

Examples

quine.hi <- aov(log(Days + 2.5) ~ .^4, quine)
quine.nxt <- update(quine.hi, . ~ . - Eth:Sex:Age:Lrn)
dropterm(quine.nxt, test = "F")
quine.stp <- stepAIC(quine.nxt,
    scope = list(upper = ~Eth*Sex*Age*Lrn, lower = ~1),
    trace = FALSE)
dropterm(quine.stp, test = "F")
quine.3 <- update(quine.stp, . ~ . - Eth:Age:Lrn)
dropterm(quine.3, test = "F")
quine.4 <- update(quine.3, . ~ . - Eth:Age)
dropterm(quine.4, test = "F")
quine.5 <- update(quine.4, . ~ . - Age:Lrn)
dropterm(quine.5, test = "F")

house.glm0 <- glm(Freq ~ Infl*Type*Cont + Sat, family=poisson,
data = housing)
house.glm1 <- update(house.glm0, . ~ . + Sat*(Infl+Type+Cont))
dropterm(house.glm1, test = "Chisq")

Description

Knight and Skagen collected during a field study on the foraging behaviour of wintering Bald Eagles in Washington State, USA data concerning 160 attempts by one (pirating) Bald Eagle to steal a chum salmon from another (feeding) Bald Eagle.
Usage
eagles

Format

The eagles data frame has 8 rows and 5 columns.

y Number of successful attempts.
n Total number of attempts.
P Size of pirating eagle (L = large, S = small).
A Age of pirating eagle (I = immature, A = adult).
V Size of victim eagle (L = large, S = small).

Source


References


Examples

eagles.glm <- glm(cbind(y, n - y) ~ P*A + V, data = eagles, family = binomial)
dropterm(eagles.glm)
prof <- profile(eagles.glm)
plot(prof)
pairs(prof)

epil

Seizure Counts for Epileptics

Description

Thall and Vail (1990) give a data set on two-week seizure counts for 59 epileptics. The number of seizures was recorded for a baseline period of 8 weeks, and then patients were randomly assigned to a treatment group or a control group. Counts were then recorded for four successive two-week periods. The subject’s age is the only covariate.

Usage

epil
Format

This data frame has 236 rows and the following 9 columns:

- **y** The count for the 2-week period.
- **trt** The treatment, "placebo" or "progabide".
- **base** The counts in the baseline 8-week period.
- **age** The subject’s age, in years.
- **V4** 0/1 indicator variable of period 4.
- **subject** The subject number, 1 to 59.
- **period** The period, 1 to 4.
- **lbase** The log-counts for the baseline period, centred to have zero mean.
- **lage** The log-ages, centred to have zero mean.

Source


References


Examples

```r
summary(glm(y ~ lbase*trt + lage + V4, family = poisson, 
            data = epil), cor = FALSE)
epil2 <- epil[epil$period == 1, ]
epil2["period"] <- rep(0, 59); epil2["y"] <- epil2["base"]
epil["time"] <- 1; epil2["time"] <- 4
epil2 <- rbind(epil, epil2)
epil2$pred <- unclass(epil2$trt) * (epil2$period > 0)
epil2$subject <- factor(epil2$subject)
epil3 <- aggregate(epil2, list(epil2$subject, epil2$period > 0),
                    function(x) if(is.numeric(x)) sum(x) else x[1])
epil3$pred <- factor(epil3$pred, 
                    labels = c("base", "placebo", "drug"))
contrasts(epil3$pred) <- structure(contr.sdif(3),
                                    dimnames = list(NULL, c("placebo-base", "drug-placebo")))
summary(glm(y ~ pred + factor(subject) + offset(log(time)), 
            family = poisson, data = epil3), cor = FALSE)

summary(glmmPQL(y ~ lbase*trt + lage + V4, 
                random = ~ 1 | subject, 
                family = poisson, data = epil))
summary(glmmPQL(y ~ pred, random = ~1 | subject, 
                family = poisson, data = epil3))
```
Plots with Geometrically Equal Scales

Description

Version of a scatterplot with scales chosen to be equal on both axes, that is 1cm represents the same units on each.

Usage

eqscplot(x, y, ratio = 1, tol = 0.04, uin, ...)

Arguments

- **x** vector of x values, or a 2-column matrix, or a list with components x and y
- **y** vector of y values
- **ratio** desired ratio of units on the axes. Units on the y axis are drawn at ratio times the size of units on the x axis. Ignored if uin is specified and of length 2.
- **tol** proportion of white space at the margins of plot
- **uin** desired values for the units-per-inch parameter. If of length 1, the desired units per inch on the x axis.
- **...** further arguments for plot and graphical parameters. Note that par(xaxs="i", yaxs="i") is enforced, and xlim and ylim will be adjusted accordingly.

Details

Limits for the x and y axes are chosen so that they include the data. One of the sets of limits is then stretched from the midpoint to make the units in the ratio given by ratio. Finally both are stretched by 1 + tol to move points away from the axes, and the points plotted.

Value

invisibly, the values of uin used for the plot.

Side Effects

performs the plot.

Note

Arguments ratio and uin were suggested by Bill Dunlap.

References

The `farms` data frame has 20 rows and 4 columns. The rows are farms on the Dutch island of Terschelling and the columns are factors describing the management of grassland.

**Usage**

```r
farms
```

**Format**

This data frame contains the following columns:

- **Mois** Five levels of soil moisture – level 3 does not occur at these 20 farms.
- **Manag** Grassland management type (SF = standard, BF = biological, HF = hobby farming, NM = nature conservation).
- **Use** Grassland use (U1 = hay production, U2 = intermediate, U3 = grazing)
- **Manure** Manure usage – classes C0 to C4.

**Source**


Quoted as from:


**References**


**Examples**

```r
farms.mca <- mca(farms, abbrev = TRUE) # Use levels as names
eqscplot(farms.mca$cs, type = "n")
text(farms.mca$rs, cex = 0.7)
text(farms.mca$cs, labels = dimnames(farms.mca$cs)[[1]], cex = 0.7)
```
Measurements of Forensic Glass Fragments

Description

The fgl data frame has 214 rows and 10 columns. It was collected by B. German on fragments of glass collected in forensic work.

Usage

fgl

Format

This data frame contains the following columns:

- **RI** refractive index; more precisely the refractive index is 1.518xxxx.
  The remaining 8 measurements are percentages by weight of oxides.

- **Na** sodium
- **Mg** manganese
- **Al** aluminium
- **Si** silicon
- **K** potassium
- **Ca** calcium
- **Ba** barium
- **Fe** iron

**type** The fragments were originally classed into seven types, one of which was absent in this dataset. The categories which occur are window float glass (WinF: 70), window non-float glass (WinNF: 76), vehicle window glass (Veh: 17), containers (Con: 13), tableware (Tabl: 9) and vehicle headlamps (Head: 29).

References

Maximum-likelihood Fitting of Univariate Distributions

Description

Maximum-likelihood fitting of univariate distributions, allowing parameters to be held fixed if desired.

Usage

fitdistr(x, densfun, start, ...)

Arguments

x  A numeric vector.
densfun  Either a character string or a function returning a density evaluated at its first argument.

Distributions "beta", "cauchy", "chi-squared", "exponential", "f", "gamma", "geometric", "log-normal", "lognormal", "logistic", "negative binomial", "normal", "Poisson", "t" and "weibull" are recognised, case being ignored.

start  A named list giving the parameters to be optimized with initial values. This can be omitted for some of the named distributions and must be for others (see Details).

...  Additional parameters, either for densfun or for optim. In particular, it can be used to specify bounds via lower or upper or both. If arguments of densfun (or the density function corresponding to a character-string specification) are included they will be held fixed.

Details

For the Normal, log-Normal, exponential and Poisson distributions the closed-form MLEs (and exact standard errors) are used, and start should not be supplied.

For all other distributions, direct optimization of the log-likelihood is performed using optim. The estimated standard errors are taken from the observed information matrix, calculated by a numerical approximation. For one-dimensional problems the Nelder-Mead method is used and for multi-dimensional problems the BFGS method, unless arguments named lower or upper are supplied when L-BFGS-B is used or method is supplied explicitly.

For the "t" named distribution the density is taken to be the location-scale family with location \( m \) and scale \( s \).

For the following named distributions, reasonable starting values will be computed if start is omitted or only partially specified: "cauchy", "gamma", "logistic", "negative binomial" (parametrized by \( \mu \) and size), "t" and "weibull". Note that these starting values may not be good enough if the fit is poor: in particular they are not resistant to outliers unless the fitted distribution is long-tailed.

There are print, coef and logLik methods for class "fitdistr".
Value

An object of class "fitdistr", a list with three components,

- **estimate** the parameter estimates,
- **sd** the estimated standard errors, and
- **loglik** the log-likelihood.

References


Examples

```r
set.seed(123)
x <- rgamma(100, shape = 5, rate = 0.1)
fitdistr(x, "gamma")
## now do this directly with more control.
fitdistr(x, dgamma, list(shape = 1, rate = 0.1), lower = 0.01)

set.seed(123)
x2 <- rt(250, df = 9)
fitdistr(x2, "t", df = 9)
## allow df to vary: not a very good idea!
fitdistr(x2, "t")
## now do fixed-df fit directly with more control.
mydt <- function(x, m, s, df) dt((x-m)/s, df)/s
fitdistr(x2, mydt, list(m = 0, s = 1), df = 9, lower = c(-Inf, 0))

set.seed(123)
x3 <- rweibull(100, shape = 4, scale = 100)
fitdistr(x3, "weibull")

set.seed(123)
x4 <- rnegbin(500, mu = 5, theta = 4)
fitdistr(x4, "Negative Binomial") # R only
```

---

**forbes**

*Forbes’ Data on Boiling Points in the Alps*

Description

A data frame with 17 observations on boiling point (degrees F) and barometric pressure in inches of mercury.

Usage

`forbes`
fractions

Format

bp  boiling point (degrees F)
pres  barometric pressure in inches of mercury

Source


fractions  Rational Approximation

Description

Find rational approximations to the components of a real numeric object using a standard continued fraction method.

Usage

fractions(x, cycles = 10, max.denominator = 2000, ...)

Arguments

x  Any object of mode numeric. Missing values are now allowed.
cycles  The maximum number of steps to be used in the continued fraction approximation process.
max.denominator  An early termination criterion. If any partial denominator exceeds max.denominator the continued fraction stops at that point.
...  arguments passed to or from other methods.

Details

Each component is first expanded in a continued fraction of the form
x = floor(x) + 1/(p1 + 1/(p2 + ...)))
where p1, p2, ... are positive integers, terminating either at cycles terms or when a pj > max.denominator. The continued fraction is then re-arranged to retrieve the numerator and denominator as integers.
The numerators and denominators are then combined into a character vector that becomes the "fracs" attribute and used in printed representations.
Arithmetic operations on "fractions" objects have full floating point accuracy, but the character representation printed out may not.
Value

An object of class "fractions". A structure with .Data component the same as the input numeric x, but with the rational approximations held as a character vector attribute, "fracs". Arithmetic operations on "fractions" objects are possible.

References


See Also

rational

Examples

X <- matrix(runif(25), 5, 5)
solve(X, X/5)
## [1,] 2.0000e-01 3.7199e-17 1.2214e-16 5.7887e-17 -8.7841e-17
## [2,] -1.1473e-16 2.0000e-01 7.0955e-17 2.0300e-17 -1.0566e-16
## [3,] 2.7975e-16 1.3653e-17 2.0000e-01 -1.3397e-16 1.5577e-16
## [4,] -2.9196e-16 2.0412e-17 1.5618e-16 2.0000e-01 -2.1921e-16
## [5,] -3.6476e-17 -3.6430e-17 3.6432e-17 4.7690e-17 2.0000e-01

fractions(solve(X, X/5))
## [1,] 1/5 0 0 0 0
## [2,] 0 1/5 0 0 0
## [3,] 0 0 1/5 0 0
## [4,] 0 0 0 1/5 0
## [5,] 0 0 0 0 1/5

fractions(solve(X, X/5)) + 1
## [1,] 6/5 1 1 1 1
## [2,] 1 6/5 1 1 1
## [3,] 1 1 6/5 1 1
## [4,] 1 1 1 6/5 1
## [5,] 1 1 1 1 6/5

galaxies

Velocities for 82 Galaxies

description

A numeric vector of velocities in km/sec of 82 galaxies from 6 well-separated conic sections of an unfilled survey of the Corona Borealis region. Multimodality in such surveys is evidence for voids and superclusters in the far universe.
gamma.dispersion

Usage

galaxies

Note

There is an 83rd measurement of 5607 km/sec in the Postman et al. paper which is omitted in Roeder (1990) and from the dataset here.
There is also a typo: this dataset has 78th observation 26690 while should be 26960.

Source


References


Examples

gal <- galaxies/1000
c(width.SJ(gal, method = "dpi"), width.SJ(gal))
plot(x = c(0, 40), y = c(0, 0.3), type = "n", bty = "l",
     xlab = "velocity of galaxy (1000km/s)", ylab = "density")
rug(gal)
lines(density(gal, width = 3.25, n = 200), lty = 1)
lines(density(gal, width = 2.56, n = 200), lty = 3)

gamma.dispersion

Calculate the MLE of the Gamma Dispersion Parameter in a GLM Fit

Description

A front end to gamma.shape for convenience. Finds the reciprocal of the estimate of the shape parameter only.

Usage

gamma.dispersion(object, ...)

Arguments

object Fitted model object giving the gamma fit.
... Additional arguments passed on to gamma.shape.
Value

The MLE of the dispersion parameter of the gamma distribution.

References


See Also

gamma.shape.glm, including the example on its help page.

gamma.shape

Estimate the Shape Parameter of the Gamma Distribution in a GLM Fit

Description

Find the maximum likelihood estimate of the shape parameter of the gamma distribution after fitting a Gamma generalized linear model.

Usage

## S3 method for class 'glm':
gamma.shape(object, it.lim = 10,
        eps.max = .Machine$double.eps^0.25, verbose = FALSE, ...)

Arguments

- **object**: Fitted model object from a Gamma family or quasi family with variance = "mu^2".
- **it.lim**: Upper limit on the number of iterations.
- **eps.max**: Maximum discrepancy between approximations for the iteration process to continue.
- **verbose**: If TRUE, causes successive iterations to be printed out. The initial estimate is taken from the deviance.
- **...**: further arguments passed to or from other methods.

Details

A glm fit for a Gamma family correctly calculates the maximum likelihood estimate of the mean parameters but provides only a crude estimate of the dispersion parameter. This function takes the results of the glm fit and solves the maximum likelihood equation for the reciprocal of the dispersion parameter, which is usually called the shape (or exponent) parameter.
Value

List of two components

alpha the maximum likelihood estimate
SE the approximate standard error, the square-root of the reciprocal of the observed information.

References


See Also

gamma.dispersion

Examples

clotting <- data.frame(
  u = c(5,10,15,20,30,40,60,80,100),
  lot1 = c(118,58,42,35,27,25,21,19,18),
  lot2 = c(69,35,26,21,18,16,13,12,12))
clot1 <- glm(lot1 ~ log(u), data = clotting, family = Gamma)
gamma.shape(clot1)
## Not run:
Alpha: 538.13
SE: 253.60
## End(Not run)

gm <- glm(Days + 0.1 ~ Age*Eth*Sex*Lrn,
  quasi(link=log, variance="mu^2"), quine,
  start = c(3, rep(0,31)))
gamma.shape(gm, verbose = TRUE)
## Not run:
Initial estimate: 1.0603
Iter. 1  Alpha: 1.23840774338543
Iter. 2  Alpha: 1.27699745778205
Iter. 3  Alpha: 1.27834332265501
Iter. 4  Alpha: 1.27834485787226

Alpha: 1.27834
SE: 0.13452
## End(Not run)

summary(gm, dispersion = gamma.dispersion(gm))  # better summary

gehan

Remission Times of Leukaemia Patients
Description

A data frame from a trial of 42 leukaemia patients. Some were treated with the drug 6-mercaptopurine and the rest are controls. The trial was designed as matched pairs, both withdrawn from the trial when either came out of remission.

Usage

gehan

Format

This data frame contains the following columns:

- **pair**: label for pair
- **time**: remission time in weeks
- **cens**: censoring, 0/1
- **treat**: treatment, control or 6-MP

Source


References


Examples

```r
library(survival)
gehan.surv <- survfit(Surv(time, cens) ~ treat, data = gehan,
  conf.type = "log-log")
summary(gehan.surv)
survreg(Surv(time, cens) ~ factor(pair) + treat, gehan, dist = "exponential")
summary(survreg(Surv(time, cens) ~ treat, gehan, dist = "exponential"))
survreg(Surv(time, cens) ~ treat, gehan)
gehan.cox <- coxph(Surv(time, cens) ~ treat, gehan)
summary(gehan.cox)
```
**Rat Genotype Data**

**Description**

Data from a foster feeding experiment with rat mothers and litters of four different genotypes: A, B, I and J. Rice litters were separated from their natural mothers at birth and given to foster mothers to rear.

**Usage**

**genotype**

**Format**

The data frame has the following components:

- **Litter**  The genotype of the litter
- **Mother**  The genotype of the foster mother
- **Wt**  Litter average weight gain of the litter, in grams at age 28 days. (The source states that the within-litter variability is negligible.)

**Source**


**References**


---

**Old Faithful Geyser Data**

**Description**

A version of the eruptions data from the ‘Old Faithful’ geyser in Yellowstone National Park, Wyoming. This version comes from Azzalini and Bowman (1990) and is of continuous measurement from August 1 to August 15, 1985.

Some nocturnal duration measurements were coded as 2, 3 or 4 minutes, having originally been described as ‘short’, ‘medium’ or ‘long’.
Usage
geyser

Format
A data frame with 299 observations on 2 variables.

<table>
<thead>
<tr>
<th>duration</th>
<th>numeric</th>
<th>Eruption time in mins</th>
</tr>
</thead>
<tbody>
<tr>
<td>waiting</td>
<td>numeric</td>
<td>Waiting time to next eruption</td>
</tr>
</tbody>
</table>

References

See Also
faithful

---

gilgais  
*Line Transect of Soil in Gilgai Territory*

Description
This dataset was collected on a line transect survey in gilgai territory in New South Wales, Australia. Gilgais are natural gentle depressions in otherwise flat land, and sometimes seem to be regularly distributed. The data collection was stimulated by the question: are these patterns reflected in soil properties? At each of 365 sampling locations on a linear grid of 4 meters spacing, samples were taken at depths 0-10 cm, 30-40 cm and 80-90 cm below the surface. pH, electrical conductivity and chloride content were measured on a 1:5 soil:water extract from each sample.

Usage
gilgais

Format
This data frame contains the following columns:

<table>
<thead>
<tr>
<th>pH00</th>
<th>pH at depth 0-10cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>pH30</td>
<td>pH at depth 30-40cm</td>
</tr>
<tr>
<td>pH80</td>
<td>pH at depth 80-90cm</td>
</tr>
<tr>
<td>e00</td>
<td>electrical conductivity in mS/cm (0-10 cm)</td>
</tr>
<tr>
<td>e30</td>
<td>electrical conductivity in mS/cm (30-40 cm)</td>
</tr>
<tr>
<td>e80</td>
<td>electrical conductivity in mS/cm (80-90 cm)</td>
</tr>
</tbody>
</table>
ginv

### Description
Calculate the Moore-Penrose generalized inverse of a matrix \( X \).

### Usage
```r
ginv(X, tol = sqrt(.Machine$double.eps))
```

### Arguments
- **X**: Matrix for which the Moore-Penrose inverse is required.
- **tol**: A relative tolerance to detect zero singular values.

### Value
A MP generalized inverse matrix for \( X \).

### References

### See Also
`solve, svd, eigen`
Examples

```r
## Not run:
# The function is currently defined as
function(X, tol = sqrt(.Machine$double.eps))
{
## Generalized Inverse of a Matrix
dnx <- dimnames(X)
if(is.null(dnx)) dnx <- vector("list", 2)
s <- svd(X)
.nz <- s$d > tol * s$d[1]
structure(
  if(any(nz)) s$v[, nz] %*% (t(s$u[, nz])/(s$d[nz])) else X,
  dimnames = dnx[2:1])
}  ## End(Not run)
```

---

**glm.convert**  
*Change a Negative Binomial fit to a GLM fit*

**Description**

This function modifies an output object from `glm.nb()` to one that looks like the output from `glm()` with a negative binomial family. This allows it to be updated keeping the theta parameter fixed.

**Usage**

`glm.convert(object)`

**Arguments**

- `object`  
  An object of class "negbin", typically the output from `glm.nb()`.

**Details**

Convenience function needed to effect some low level changes to the structure of the fitted model object.

**Value**

An object of class "glm" with negative binomial family. The theta parameter is then fixed at its present estimate.

**See Also**

`glm.nb`, `negative.binomial`, `glm`
Examples

quine.nb1 <- glm.nb(Days ~ Sex/(Age + Eth*Lrn), data = quine)
quine.nbA <- glm.convert(quine.nb1)
quine.nbB <- update(quine.nb1, . ~ . + Sex:Age:Lrn)
anova(quine.nbA, quine.nbB)

glm.nb

Fit a Negative Binomial Generalized Linear Model

Description

A modification of the system function glm() to include estimation of the additional parameter, theta, for a Negative Binomial generalized linear model.

Usage

glm.nb(formula, data, weights, subset, na.action,
       start = NULL, etastart, mustart,
       control = glm.control(...), method = "glm.fit",
       model = TRUE, x = FALSE, y = TRUE, contrasts = NULL,
       init.theta, link = log)

Arguments

formula, data, weights, subset, na.action, start, etastart, mustart, control, method

arguments for the glm() function. Note that these exclude family and offset
(but offset() can be used).

init.theta Optional initial value for the theta parameter. If omitted a moment estimator
after an initial fit using a Poisson GLM is used.

link The link function. Currently must be one of log, sqrt or identity.

Details

An alternating iteration process is used. For given theta the GLM is fitted using the same process
as used by glm(). For fixed means the theta parameter is estimated using score and infor-
mation iterations. The two are alternated until convergence of both. (The number of alternations
and the number of iterations when estimating theta are controlled by the maxit parameter of
glm.control.)

Setting trace > 0 traces the alternating iteration process. Setting trace > 1 traces the glm
fit, and setting trace > 2 traces the estimation of theta.

Value

A fitted model object of class negbin inheriting from glm and lm. The object is like the output
of glm but contains three additional components, namely theta for the ML estimate of theta,
SE.theta for its approximate standard error (using observed rather than expected information),
and twologlik for twice the log-likelihood function.
References

See Also

`glm, negative.binomial, anova.negbin, summary.negbin, theta.md`

Examples

```r
quine.nb1 <- glm.nb(Days ~ Sex/(Age + Eth+Lrn), data = quine)
quine.nb2 <- update(quine.nb1, . ~ . + Sex:Age:Lrn)
quine.nb3 <- update(quine.nb2, Days ~ .^4)
anova(quine.nb1, quine.nb2, quine.nb3)
```

---

**glmmPQL**

*Fit Generalized Linear Mixed Models via PQL*

**Description**

Fit a GLMM model with multivariate normal random effects, using Penalized Quasi-Likelihood.

**Usage**

```r
glmmPQL(fixed, random, family, data, correlation, weights, control, niter = 10, verbose = TRUE, ...)
```

**Arguments**

- `fixed` a two-sided linear formula giving fixed-effects part of the model.
- `random` A formula or list of formulae describing the random effects.
- `family` a GLM family.
- `data` an optional data frame used as the first place to find variables in the formulae.
- `correlation` an optional correlation structure.
- `weights` optional case weights as in `glm`.
- `control` an optional argument to be passed to `lme`.
- `niter` maximum number of iterations.
- `verbose` logical: print out record of iterations?
- `...` Further arguments for `lme`.

**Details**

`glmmPQL` works by repeated calls to `lme`, so package `nlme` will be loaded at first use if necessary.

**Value**

A object of class "`lme`": see `lmeObject`.
hills

References

See Also
lme

Examples
```r
library(nlme) # will be loaded automatically if omitted
summary(glmmPQL(y ~ trt + I(week > 2), random = ~ 1 | ID,
    family = binomial, data = bacteria))
```

hills

Record Times in Scottish Hill Races

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

Usage
hills

Format
The components are:
- **dist** distance in miles (on the map)
- **climb** total height gained during the route, in feet.
- **time** record time in minutes.

Source
[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.]

References
hist.scott  

Plot a Histogram with Automatic Bin Width Selection

Description

Plot a histogram with automatic bin width selection, using the Scott or Freedman–Diaconis formulae.

Usage

```r
hist.scott(x, prob = TRUE, xlab = deparse(substitute(x)), ...)  
```

```r
hist.FD(x, prob = TRUE, xlab = deparse(substitute(x)), ...)  
```

Arguments

- `x`: A data vector
- `prob`: Should the plot have unit area, so be a density estimate?
- `xlab, ...`: Further arguments to `hist`.

Value

For the `nclass.*` functions, the suggested number of classes.

Side Effects

Plot a histogram.

References


See Also

`hist`

housing  

Frequency Table from a Copenhagen Housing Conditions Survey

Description

The `housing` data frame has 72 rows and 5 variables.

Usage

`housing`
Format

**Sat**  Satisfaction of householders with their present housing circumstances, (High, Medium or Low, ordered factor).

**Infl**  Perceived degree of influence householders have on the management of the property (High, Medium, Low).

**Type**  Type of rental accommodation, (Tower, Atrium, Apartment, Terrace).

**Cont**  Contact residents are afforded with other residents, (Low, High).

**Freq**  Frequencies: the numbers of residents in each class.

Source


References


Examples

```r
options(contrasts = c("contr.treatment", "contr.poly"))

# Surrogate Poisson models
house.glm0 <- glm(Freq ~ Infl*Type*Cont + Sat, family = poisson,
                   data = housing)
summary(house.glm0, cor = FALSE)
addterm(house.glm0, ~. + Sat:(Infl+Type+Cont), test = "Chisq")

house.glm1 <- update(house.glm0, . ~ . + Sat*(Infl+Type+Cont))
summary(house.glm1, cor = FALSE)

1 - pchisq(deviance(house.glm1), house.glm1$df.residual)

dropterm(house.glm1, test = "Chisq")
addterm(house.glm1, ~. + Sat:(Infl+Type+Cont)^2, test = "Chisq")

hnames <- lapply(housing[, -5], levels) # omit Freq
newData <- expand.grid(hnames)
newData$Sat <- ordered(newData$Sat)
house.pm <- predict(house.glm1, newData,
                    type = "response") # poisson means
house.pm <- matrix(house.pm, ncol = 3, byrow = TRUE,
                    dimnames = list(NULL, hnames[[1]]))
house.pr <- house.pm/drop(house.pm %*% rep(1, 3))
cbind(expand.grid(hnames[-1]), round(house.pr, 2))
```
# Iterative proportional scaling
loglm(Freq ~ Infl*Type*Cont + Sat*(Infl+Type+Cont), data = housing)

# multinomial model
library(nnet)
(house.mult <- multinom(Sat ~ Infl + Type + Cont, weights = Freq, 
data = housing))
house.mult2 <- multinom(Sat ~ Infl*Type*Cont, weights = Freq, 
data = housing)
anova(house.mult, house.mult2)

house.pm <- predict(house.mult, expand.grid(hnames[-1]), 
                      type = "probs")
cbind(expand.grid(hnames[-1]), round(house.pm, 2))

# proportional odds model
house.cpr <- apply(house.pr, 1, cumsum)
logit <- function(x) log(x/(1-x))
house.ld <- logit(house.cpr[2, ]) - logit(house.cpr[1, ])
(ratio <- sort(drop(house.ld)))
mean(ratio)

(house.plr <- polr(Sat ~ Infl + Type + Cont, 
                    data = housing, weights = Freq))

house.pr1 <- predict(house.plr, expand.grid(hnames[-1]), 
                      type = "probs")
cbind(expand.grid(hnames[-1]), round(house.pr1, 2))

Fr <- matrix(housing$Freq, ncol = 3, byrow = TRUE)
2*sum(Fr*log(house.pr/house.pr1))

house.plr2 <- stepAIC(house.plr, ~.^2)
house.plr2$anova

---

**huber**

*Huber M-estimator of Location with MAD Scale*

**Description**

Finds the Huber M-estimator of location with MAD scale.

**Usage**

huber(y, k = 1.5, tol = 1e-06)

**Arguments**

- **y** vector of data values
- **k** Winsorizes at \( k \) standard deviations
- **tol** convergence tolerance
Value

list of location and scale parameters

mu location estimate
s MAD scale estimate

References


See Also

hubers, mad

Examples

huber(chem)

hubers(y, k = 1.5, mu, s, initmu = median(y), tol = 1e-06)

Arguments

y vector y of data values
k Winsorizes at k standard deviations
mu specified location
s specified scale
initmu initial value of mu
tol convergence tolerance

Value

list of location and scale estimates

mu location estimate
s scale estimate
References


See Also

huber

Examples

hubers(chem)
hubers(chem, mu=3.68)

---

Yields from a Barley Field Trial

Description

The *immer* data frame has 30 rows and 4 columns. Five varieties of barley were grown in six locations in each of 1931 and 1932.

Usage

immer

Format

This data frame contains the following columns:

- **Loc** The location.
- **Var** The variety of barley ("manchuria", "svansota", "velvet", "trebi" and "peatland").
- **Y1** Yield in 1931
- **Y2** Yield in 1932

Source


References

Examples

```r
immer.aov <- aov(cbind(Y1,Y2) ~ Loc + Var, data = immer)
summary(immer.aov)

immer.aov <- aov((Y1+Y2)/2 ~ Var + Loc, data = immer)
summary(immer.aov)
model.tables(immer.aov, type = "means", se = TRUE, cterms = "Var")
```

isoMDS  

Kruskal’s Non-metric Multidimensional Scaling

Description

One form of non-metric multidimensional scaling

Usage

```r
isoMDS(d, y = cmdscale(d, k), k = 2, maxit = 50, trace = TRUE, 
      tol = 1e-3, p = 2)

Shepard(d, x, p = 2)
```

Arguments

- `d`  
  distance structure of the form returned by `dist`, or a full, symmetric matrix. Data are assumed to be dissimilarities or relative distances, but must be positive except for self-distance. Both missing and infinite values are allowed.

- `y`  
  An initial configuration. If none is supplied, `cmdscale` is used to provide the classical solution, unless there are missing or infinite dissimilarities.

- `k`  
  The desired dimension for the solution, passed to `cmdscale`.

- `maxit`  
  The maximum number of iterations.

- `trace`  
  Logical for tracing optimization. Default `TRUE`.

- `tol`  
  Convergence tolerance.

- `p`  
  Power for Minkowski distance in the configuration space.

- `x`  
  A final configuration.

Details

This chooses a k-dimensional (default k = 2) configuration to minimize the stress, the square root of the ratio of the sum of squared differences between the input distances and those of the configuration to the sum of configuration distances squared. However, the input distances are allowed a monotonic transformation.

An iterative algorithm is used, which will usually converge in around 10 iterations. As this is necessarily an $O(n^2)$ calculation, it is slow for large datasets. Further, since for the default $p = 2$ the configuration is only determined up to rotations and reflections (by convention the centroid is at the origin), the result can vary considerably from machine to machine.
Value

Two components:

- **points**: A k-column vector of the fitted configuration.
- **stress**: The final stress achieved (in percent).

Side Effects

If `trace` is true, the initial stress and the current stress are printed out every 5 iterations.

References


See Also

`cmdscale`, `sammon`

Examples

data(swiss)
swiss.x <- as.matrix(swiss[, -1])
swiss.dist <- dist(swiss.x)
swiss.mds <- isoMDS(swiss.dist)
plot(swiss.mds$points, type = "n")
text(swiss.mds$points, labels = as.character(1:nrow(swiss.x)))
swiss.sh <- Shepard(swiss.dist, swiss.mds$points)
plot(swiss.sh, pch = ".")
lines(swiss.sh$x, swiss.sh$yf, type = "S")

---

**kde2d**

*Two-Dimensional Kernel Density Estimation*

Description

Two-dimensional kernel density estimation with an axis-aligned bivariate normal kernel, evaluated on a square grid.

Usage

```r
kde2d(x, y, h, n = 25, lims = c(range(x), range(y)))
```
Arguments

\(x\)  
x coordinate of data
\(y\)  
y coordinate of data
\(h\)  
vector of bandwidths for x and y directions. Defaults to normal reference bandwidth (see \texttt{bandwidth.nrd}).
\(n\)  
Number of grid points in each direction.
\(\text{lims}\)  
The limits of the rectangle covered by the grid as \(c(x_l, x_u, y_l, y_u)\).

Value

A list of three components.

\(\mathbf{x}, \mathbf{y}\)  
The x and y coordinates of the grid points, vectors of length \(n\).
\(\mathbf{z}\)  
An \(n \times n\) matrix of the evaluated density.

References


Examples

```r
attach(geyser)
plot(duration, waiting, xlim = c(0.5, 6), ylim = c(40, 100))
f1 <- kde2d(duration, waiting, n = 50, lims = c(0.5, 6, 40, 100))
image(f1, zlim = c(0, 0.05))
f2 <- kde2d(duration, waiting, n = 50, lims = c(0.5, 6, 40, 100),
  h = c(width.SJ(duration), width.SJ(waiting)) )
image(f2, zlim = c(0, 0.05))
persp(f2, phi = 30, theta = 20, d = 5)

plot(duration[-272], duration[-1], xlim = c(0.5, 6),
  ylim = c(1, 6), xlab = "previous duration", ylab = "duration")
f1 <- kde2d(duration[-272], duration[-1],
  h = rep(1.5, 2), n = 50, lims = c(0.5, 6, 0.5, 6))
contour(f1, xlab = "previous duration",
  ylab = "duration", levels = c(0.05, 0.1, 0.2, 0.4))
f1 <- kde2d(duration[-272], duration[-1],
  h = rep(0.6, 2), n = 50, lims = c(0.5, 6, 0.5, 6))
contour(f1, xlab = "previous duration",
  ylab = "duration", levels = c(0.05, 0.1, 0.2, 0.4))
f1 <- kde2d(duration[-272], duration[-1],
  h = rep(0.4, 2), n = 50, lims = c(0.5, 6, 0.5, 6))
contour(f1, xlab = "previous duration",
  ylab = "duration", levels = c(0.05, 0.1, 0.2, 0.4))
detach("geyser")
```
# lda

**Linear Discriminant Analysis**

## Description

Linear discriminant analysis.

## Usage

```r
lda(x, ...)
```

### S3 method for class 'formula':

```r
lda(formula, data, ..., subset, na.action)
```

### Default S3 method:

```r
lda(x, grouping, prior = proportions, tol = 1.0e-4,
   method, CV = FALSE, nu, ...)
```

### S3 method for class 'data.frame':

```r
lda(x, ...)
```

### S3 method for class 'matrix':

```r
lda(x, grouping, ..., subset, na.action)
```

## Arguments

- **formula**: A formula of the form `groups ~ x1 + x2 + ...`. That is, the response is the grouping factor and the right hand side specifies the (non-factor) discriminators.
- **data**: Data frame from which variables specified in `formula` are preferentially to be taken.
- **x**: (required if no formula is given as the principal argument.) a matrix or data frame or Matrix containing the explanatory variables.
- **grouping**: (required if no formula principal argument is given.) a factor specifying the class for each observation.
- **prior**: the prior probabilities of class membership. If unspecified, the class proportions for the training set are used. If present, the probabilities should be specified in the order of the factor levels.
- **tol**: A tolerance to decide if a matrix is singular; it will reject variables and linear combinations of unit-variance variables whose variance is less than `tol^2`.
- **subset**: An index vector specifying the cases to be used in the training sample. (NOTE: If given, this argument must be named.)
- **na.action**: A function to specify the action to be taken if NAs are found. The default action is for the procedure to fail. An alternative is `na.omit`, which leads to rejection of cases with missing values on any required variable. (NOTE: If given, this argument must be named.)
method

"moment" for standard estimators of the mean and variance, "mle" for MLEs, "mve" to use cov.mve, or "t" for robust estimates based on a t distribution.

CV

If true, returns results (classes and posterior probabilities) for leave-one-out cross-validation. Note that if the prior is estimated, the proportions in the whole dataset are used.

nu

degrees of freedom for method = "t".

... arguments passed to or from other methods.

Details

The function tries hard to detect if the within-class covariance matrix is singular. If any variable has within-group variance less than tol^2 it will stop and report the variable as constant. This could result from poor scaling of the problem, but is more likely to result from constant variables.

Specifying the prior will affect the classification unless over-ridden in predict.lda. Unlike in most statistical packages, it will also affect the rotation of the linear discriminants within their space, as a weighted between-groups covariance matrix is used. Thus the first few linear discriminants emphasize the differences between groups with the weights given by the prior, which may differ from their prevalence in the dataset.

If one or more groups is missing in the supplied data, they are dropped with a warning, but the classifications produced are with respect to the original set of levels.

Value

If CV = TRUE the return value is a list with components class, the MAP classification (a factor), and posterior, posterior probabilities for the classes.

Otherwise it is an object of class "lda" containing the following components:

prior

the prior probabilities used.

means

the group means.

scaling

a matrix which transforms observations to discriminant functions, normalized so that within groups covariance matrix is spherical.

svd

the singular values, which give the ratio of the between- and within-group standard deviations on the linear discriminant variables. Their squares are the canonical F-statistics.

N

The number of observations used.

call

The (matched) function call.

Note

This function may be called giving either a formula and optional data frame, or a matrix and grouping factor as the first two arguments. All other arguments are optional, but subset= and na.action=, if required, must be fully named.

If a formula is given as the principal argument the object may be modified using update() in the usual way.
References


See Also

predict.lda, qda, predict.qda

Examples

data(iris3)
Iris <- data.frame(rbind(iris3[,1], iris3[,2], iris3[,3]),
  Sp = rep(c("s","c","v"), rep(50,3)))
train <- sample(1:150, 75)
table(Iris$Sp[train])
## your answer may differ
## c  s  v
##  22 23 30
z <- lda(Sp ~ ., Iris, prior = c(1,1,1)/3, subset = train)
predict(z, Iris[-train, ])$class
## [1] s s s s s s s s s s s s s s s s s s s s c c c
## [31] c c c c c c c c c c c c c c c c c c v v v v
## [61] v v v v v v v v v v v v v v v v
(z1 <- update(z, . ~ . - Petal.W.))

\texttt{ldahist}

\vspace{1em}

\textit{Histograms or Density Plots of Multiple Groups}

Description

Plot histograms or density plots of data on a single Fisher linear discriminant.

Usage

\texttt{ldahist(data, g, nbins = 25, h, x0 = - h/1000, breaks,}
\texttt{xlim = range(breaks), ymax = 0, width,}
\texttt{type = c("histogram", "density", "both"),}
\texttt{sep = (type !="density"),}
\texttt{col = 5, xlab = deparse(substitute(data)), bty = "n", ...)}

Arguments

data \texttt{vector of data. Missing values (NAs) are allowed and omitted.}
g \texttt{factor or vector giving groups, of the same length as data.}
nbins \texttt{Suggested number of bins to cover the whole range of the data.}
h \texttt{The bin width (takes precedence over nbins).}
x0 \texttt{Shift for the bins - the breaks are at x0 + h * (..., -1, 0, 1, ...)}}
leuk

breaks The set of breakpoints to be used. (Usually omitted, takes precedence over h and nbins).
xlim The limits for the x-axis.
ymax The upper limit for the y-axis.
width Bandwidth for density estimates. If missing, the Sheather-Jones selector is used for each group separately.
type Type of plot.
sep Whether there is a separate plot for each group, or one combined plot.
col The colour number for the bar fill.
xlab label for the plot x-axis. By default, this will be the name of data.
bty The box type for the plot - defaults to none.
... additional arguments to polygon.

Side Effects
Histogram and/or density plots are plotted on the current device.

References

See Also
plot.lda.

leuk Survival Times and White Blood Counts for Leukaemia Patients

Description
A data frame of data from 33 leukaemia patients.

Usage
leuk

Format
A data frame with columns:

wbc white blood count
ag a test result, "present" or "absent"
time survival time in weeks
Details

Survival times are given for 33 patients who died from acute myelogenous leukaemia. Also measured was the patient’s white blood cell count at the time of diagnosis. The patients were also factored into 2 groups according to the presence or absence of a morphologic characteristic of white blood cells. Patients termed AG positive were identified by the presence of Auer rods and/or significant granulation of the leukaemic cells in the bone marrow at the time of diagnosis.

Source

Taken from  

References


Examples

```r
library(survival)
plot(survfit(Surv(time) ~ ag, data = leuk), lty = 2:3, col = 2:3)

# now Cox models
leuk.cox <- coxph(Surv(time) ~ ag + log(wbc), leuk)
summary(leuk.cox)
```

---

**lm.gls**

*Fit Linear Models by Generalized Least Squares*

Description

Fit linear models by Generalized Least Squares

Usage

```r
lm.gls(formula, data, W, subset, na.action, inverse = FALSE, 
method = "qr", model = FALSE, x = FALSE, y = FALSE, 
contrasts = NULL, ...)
```

Arguments

- **formula**  
a formula expression as for regression models, of the form `response ~ predictors`. See the documentation of `formula` for other details.
- **data**  
an optional data frame in which to interpret the variables occurring in `formula`.
- **W**  
a weight matrix.
lm.ridge

Ridge Regression

Description

Fit a linear model by ridge regression.

Usage

lm.ridge(formula, data, subset, na.action, lambda = 0, model = FALSE,
x = FALSE, y = FALSE, contrasts = NULL, ...)
Arguments

- **formula**: a formula expression as for regression models, of the form `response ~ predictors`. See the documentation of `formula` for other details. `offset` terms are allowed.
- **data**: an optional data frame in which to interpret the variables occurring in `formula`.
- **subset**: expression saying which subset of the rows of the data should be used in the fit. All observations are included by default.
- **na.action**: a function to filter missing data.
- **lambda**: A scalar or vector of ridge constants.
- **model**: should the model frame be returned?
- **x**: should the design matrix be returned?
- **y**: should the response be returned?
- **contrasts**: a list of contrasts to be used for some or all of factor terms in the formula. See the `contrasts.arg` of `model.matrix.default`.
- **...**: additional arguments to `lm.fit`.

Details

If an intercept is present in the model, its coefficient is not penalized. (If you want to penalized an intercept, put in your own constant term and remove the intercept.)

Value

A list with components

- **coef**: matrix of coefficients, one row for each value of `lambda`. Note that these are not on the original scale and are for use by the `coef` method.
- **scales**: scalings used on the X matrix.
- **Inter**: was intercept included?
- **lambda**: vector of lambda values
- **ym**: mean of y
- **xm**: column means of x matrix
- **GCV**: vector of GCV values
- **kHKB**: HKB estimate of the ridge constant.
- **kLW**: L-W estimate of the ridge constant.

References


See Also

- `lm`
Examples

data(longley) # not the same as the S-PLUS dataset
names(longley)[1] <- "y"
lm.ridge(y ~ ., longley)
plot(lm.ridge(y ~ ., longley,
lambda = seq(0,0.1,0.001)))
select(lm.ridge(y ~ ., longley,
lambda = seq(0,0.1,0.0001)))

loglm

Fit Log-Linear Models by Iterative Proportional Scaling

Description

This function provides a front-end to the standard function, loglin, to allow log-linear models to be specified and fitted in a manner similar to that of other fitting functions, such as glm.

Usage

loglm(formula, data, subset, na.action, ...)

Arguments

formula A linear model formula specifying the log-linear model.

If the left-hand side is empty, the data argument is required and must be a (complete) array of frequencies. In this case the variables on the right-hand side may be the names of the dimnames attribute of the frequency array, or may be the positive integers: 1, 2, 3, ... used as alternative names for the 1st, 2nd, 3rd, ... dimension (classifying factor). If the left-hand side is not empty it specifies a vector of frequencies. In this case the data argument, if present, must be a data frame from which the left-hand side vector and the classifying factors on the right-hand side are (preferentially) obtained. The usual abbreviation of a . to stand for ‘all other variables in the data frame’ is allowed. Any non-factors on the right-hand side of the formula are coerced to factor.

data Numeric array or data frame. In the first case it specifies the array of frequencies; in the second it provides the data frame from which the variables occurring in the formula are preferentially obtained in the usual way.

This argument may be the result of a call to xtabs.

subset Specifies a subset of the rows in the data frame to be used. The default is to take all rows.

na.action Specifies a method for handling missing observations. The default is to fail if missing values are present.

... May supply other arguments to the function loglm1.
Details

If the left-hand side of the formula is empty the data argument supplies the frequency array and the right-hand side of the formula is used to construct the list of fixed faces as required by loglin. Structural zeros may be specified by giving a start argument with those entries set to zero, as described in the help information for loglin.

If the left-hand side is not empty, all variables on the right-hand side are regarded as classifying factors and an array of frequencies is constructed. If some cells in the complete array are not specified they are treated as structural zeros. The right-hand side of the formula is again used to construct the list of faces on which the observed and fitted totals must agree, as required by loglin. Hence terms such as a:b, a+b and a/b are all equivalent.

Value

An object of class "loglm" conveying the results of the fitted log-linear model. Methods exist for the generic functions print, summary, deviance, fitted, coef, resid, anova and update, which perform the expected tasks. Only log-likelihood ratio tests are allowed using anova.

The deviance is simply an alternative name for the log-likelihood ratio statistic for testing the current model within a saturated model, in accordance with standard usage in generalized linear models.

Warning

If structural zeros are present, the calculation of degrees of freedom may not be correct. loglin itself takes no action to allow for structural zeros. loglm deducts one degree of freedom for each structural zero, but cannot make allowance for gains in error degrees of freedom due to loss of dimension in the model space. (This would require checking the rank of the model matrix, but since iterative proportional scaling methods are developed largely to avoid constructing the model matrix explicitly, the computation is at least difficult.)

When structural zeros (or zero fitted values) are present the estimated coefficients will not be available due to infinite estimates. The deviances will normally continue to be correct, though.

References


See Also

loglm, loglin

Examples

# The data frames Cars93, minn38 and quine are available # in the MASS package.

# Case 1: frequencies specified as an array.
sapply(minn38[, -5], levels))
## hs phs fol sex f
## 3 4 7 2 0
minn38a <- array(0, c(3,4,7,2), 1apply(minn38[, -5], levels))
loglm

minn38a <- data.matrix(minn38[, -5])
fm <- loglm(~1 + 2 + 3 + 4, minn38a)  # numerals as names.
deviance(fm)
## [1] 3711.9
fml <- update(fm, .~.^2)
fml2 <- update(fm, .~.^3, print = TRUE)
## 5 iterations: deviation 0.0750732
anova(fm, fml, fm2)
## Not run: LR tests for hierarchical log-linear models

Model 1:
  ~ 1 + 2 + 3 + 4
Model 2:
  . ~ 1 + 2 + 3 + 4 + 1:2 + 1:3 + 1:4 + 2:3 + 2:4 + 3:4
Model 3:
  . ~ 1 + 2 + 3 + 4 + 1:2 + 1:3 + 1:4 + 2:3 + 2:4 + 3:4 +
   1:2:3 + 1:2:4 + 1:3:4 + 2:3:4

Deviance df Delta(Dev) Delta(df) P(> Delta(Dev)
Model 1  3711.915 155
Model 2  220.043 108  3491.873  47  0.00000
Model 3  47.745  36  172.298  72  0.00000
Saturated 0.000  0  47.745  36  0.09114

## End(Not run)
# Case 1. An array generated with xtabs.
loglm(~ Type + Origin, xtabs(~ Type + Origin, Cars93))
## Not run: Call:
loglm(formula = ~Type + Origin, data = xtabs(~Type + Origin,  
Cars93))

Statistics:
   X^2  df  P(> X^2)
Likelihood Ratio 18.362  5  0.0025255
   Pearson 14.080  5  0.0151101

## End(Not run)
# Case 2. Frequencies given as a vector in a data frame
names(quine)
## [1] "Eth" "Sex" "Age" "Lrn" "Days"
fm <- loglm(Days ~ .^2, quine)
gm <- glm(Days ~ .^2, poisson, quine)  # check glm.
c(deviance(fm), deviance(gm))  # deviances agree
## [1] 1368.7 1368.7
c(fm$dfl, gm$dfl)  # resid df do not!
c(fm$dfl, gm$dfl.residual)  # resid df do not!
## [1] 127 128
# The loglm residual degrees of freedom is wrong because of
# a non-detectable redundancy in the model matrix.
**loglm1**

*Fit Log-Linear Models by Iterative Proportional Scaling – Internal function*

**Description**

`loglm1` is an internal function used by `loglm`. It is a generic function dispatching on the `data` argument.

**Usage**

```r
loglm1(formula, data, ...)  

## S3 method for class 'xtabs':  
loglm1(formula, data, ...)  

## S3 method for class 'data.frame':  
loglm1(formula, data, ...)  

## Default S3 method:  
loglm1(formula, data, start = rep(1, length(data)), fitted = FALSE,  
keep.frequencies = fitted, param = TRUE, eps = 1/10,  
iter = 40, print = FALSE, ...)  
```

**Arguments**

- `formula` A linear model formula specifying the log-linear model. See `loglm` for its interpretation.
- `data` Numeric array or data frame. In the first case it specifies the array of frequencies; in the second it provides the data frame from which the variables occurring in the formula are preferentially obtained in the usual way. This argument may also be the result of a call to `xtabs`.
- `start`, `param`, `eps`, `iter`, `print` Arguments passed to `loglin`.
- `fitted` logical: should the fitted values be returned?
- `keep.frequencies` If `TRUE` specifies that the (possibly constructed) array of frequencies is to be retained as part of the fitted model object. The default action is to use the same value as that used for `fitted`.
- `...` arguments passed to the default method.

**Value**

An object of class "loglm".
logtrans

See Also

\texttt{loglm, loglin}

---

\textbf{logtrans} \hspace{1cm} \textit{Estimate log Transformation Parameter}

\section*{Description}

Find and optionally plot the marginal (profile) likelihood for alpha for a transformation model of the form $\log(y + \alpha) \sim x_1 + x_2 + \ldots$.

\section*{Usage}

\texttt{logtrans(object, ...)}

\begin{verbatim}
## Default S3 method: logtrans(object, ..., alpha = seq(0.5, 6, by = 0.25) - min(y),
   plotit = TRUE, interp =, xlab = "alpha",
   ylab = "log Likelihood")

## S3 method for class 'formula': logtrans(object, data, ...)

## S3 method for class 'lm': logtrans(object, ...)

## S3 method for class 'logtrans': logtrans(object, ...)  # not shown
\end{verbatim}

\section*{Arguments}

\begin{itemize}
  \item \texttt{object} \hspace{1cm} Fitted linear model object, or formula defining the untransformed model that is $y \sim x_1 + x_2 + \ldots$. The function is generic.
  \item \texttt{...} \hspace{1cm} If \texttt{object} is a formula, this argument may specify a data frame as for \texttt{lm}.
  \item \texttt{alpha} \hspace{1cm} Set of values for the transformation parameter, alpha.
  \item \texttt{plotit} \hspace{1cm} Should plotting be done?
  \item \texttt{interp} \hspace{1cm} Should the marginal log-likelihood be interpolated with a spline approximation? (Default is \texttt{TRUE} if plotting is to be done and the number of real points is less than 100.)
  \item \texttt{xlab} \hspace{1cm} as for \texttt{plot}.
  \item \texttt{ylab} \hspace{1cm} as for \texttt{plot}.
  \item \texttt{data} \hspace{1cm} optional data argument for \texttt{lm} fit.
\end{itemize}

\section*{Value}

List with components $x$ (for alpha) and $y$ (for the marginal log-likelihood values).
Side Effects

A plot of the marginal log-likelihood is produced, if requested, together with an approximate mle and 95% confidence interval.

References


See Also

`boxcox`

Examples

```r
logtrans(Days ~ Age*Sex*Eth*Lrn, data = quine,
    alpha = seq(0.75, 6.5, len=20))
```

---

### lqs

#### Resistant Regression

**Description**

Fit a regression to the good points in the dataset, thereby achieving a regression estimator with a high breakdown point. `lmsreg` and `ltsreg` are compatibility wrappers.

**Usage**

```r
lqs(x, ...)  
```

## S3 method for class 'formula':

```r
lqs(formula, data, ...,  
    method = c("lts", "lqs", "lms", "S", "model.frame"),  
    subset, na.action, model = TRUE,  
    x.ret = FALSE, y.ret = FALSE, contrasts = NULL)
```

## Default S3 method:

```r
lqs(x, y, intercept = TRUE, method = c("lts", "lqs", "lms", "S"),  
    quantile, control = lqs.control(...), k0 = 1.548, seed, ...)
```

lmsreg(...)

ltsreg(...)

Arguments

- **formula**: a formula of the form \( y \sim x_1 + x_2 + \ldots \).
- **data**: data frame from which variables specified in `formula` are preferentially to be taken.
- **subset**: an index vector specifying the cases to be used in fitting. (NOTE: If given, this argument must be named exactly.)
- **na.action**: function to specify the action to be taken if NAs are found. The default action is for the procedure to fail. Alternatives include `na.omit` and `na.exclude`, which lead to omission of cases with missing values on any required variable. (NOTE: If given, this argument must be named exactly.)
- **model, x.ret, y.ret**: logical. If TRUE the model frame, the model matrix and the response are returned, respectively.
- **contrasts**: an optional list. See the `contrasts.arg` of `model.matrix.default`.
- **x**: a matrix or data frame containing the explanatory variables.
- **y**: the response: a vector of length the number of rows of `x`.
- **intercept**: should the model include an intercept?
- **method**: the method to be used. `model.frame` returns the model frame: for the others see the Details section. Using `lmsreg` or `ltsreg` forces "lms" and "lts" respectively.
- **quantile**: the quantile to be used: see Details. This is over-ridden if `method = "lms"`.
- **control**: additional control items: see Details.
- **k0**: the cutoff / tuning constant used for \( \chi() \) and \( \psi() \) functions when `method = "S"`, currently corresponding to Tukey’s ‘biweight’.
- **seed**: the seed to be used for random sampling: see `.Random.seed`. The current value of `.Random.seed` will be preserved if it is set.
- **...**: arguments to be passed to `lqs.default` or `lqs.control`, see control above and Details.

Details

Suppose there are \( n \) data points and \( p \) regressors, including any intercept.

The first three methods minimize some function of the sorted squared residuals. For methods "lqs" and "lms" is the quantile squared residual, and for "lts" it is the sum of the quantile smallest squared residuals. "lqs" and "lms" differ in the defaults for quantile, which are floor\((n+p+1)/2\) and floor\((n+1)/2\) respectively. For "lts" the default is floor\((n/2) + floor((p+1)/2)\).

The "S" estimation method solves for the scale \( s \) such that the average of a function chi of the residuals divided by \( s \) is equal to a given constant.

The control argument is a list with components

- **psamp**: the size of each sample. Defaults to \( p \).
**nsamp:** the number of samples or "best" (the default) or "exact" or "sample". If "sample" the number chosen is \( \min(5+p, 3000) \), taken from Rousseeuw and Hubert (1997). If "best" exhaustive enumeration is done up to 5000 samples; if "exact" exhaustive enumeration will be attempted however many samples are needed.

**adjust:** should the intercept be optimized for each sample? Defaults to TRUE.

**Value**

An object of class "lqs". This is a list with components

- **crit** the value of the criterion for the best solution found, in the case of method == "S" before IWLS refinement.
- **sing** character. A message about the number of samples which resulted in singular fits.
- **coefficients** of the fitted linear model
- **bestone** the indices of those points fitted by the best sample found (prior to adjustment of the intercept, if requested).
- **fitted.values** the fitted values.
- **residuals** the residuals.
- **scale** estimate(s) of the scale of the error. The first is based on the fit criterion. The second (not present for method == "S") is based on the variance of those residuals whose absolute value is less than 2.5 times the initial estimate.

**Note**

There seems no reason other than historical to use the lms and lqs options. LMS estimation is of low efficiency (converging at rate \( n^{-1/3} \)) whereas LTS has the same asymptotic efficiency as an M estimator with trimming at the quartiles (Marazzi, 1993, p.201). LQS and LTS have the same maximal breakdown value of \( (\text{floor}(n-p)/2) + 1)/n \) attained if \( \text{floor}(n+p)/2 \) <= quantile <= \( \text{floor}(n+p+1)/2 \). The only drawback mentioned of LTS is greater computation, as a sort was thought to be required (Marazzi, 1993, p.201) but this is not true as a partial sort can be used (and is used in this implementation).

Adjusting the intercept for each trial fit does need the residuals to be sorted, and may be significant extra computation if \( n \) is large and \( p \) small.

Opinions differ over the choice of psamp. Rousseeuw and Hubert (1997) only consider \( p \); Marazzi (1993) recommends \( p+1 \) and suggests that more samples are better than adjustment for a given computational limit.

The computations are exact for a model with just an intercept and adjustment, and for LQS for a model with an intercept plus one regressor and exhaustive search with adjustment. For all other cases the minimization is only known to be approximate.

**Author(s)**

B. D. Ripley
mammals

References


See Also

predict.lqs

Examples

```r
data(stackloss)
set.seed(123)
lqs(stack.loss ~ ., data = stackloss)
lqs(stack.loss ~ ., data = stackloss, method = "S", nsamp = "exact")
```

mammals

Brain and Body Weights for 62 Species of Land Mammals

Description

A data frame with average brain and body weights for 62 species of land mammals.

Usage

mammals

Format

- **body**: body weight in kg
- **brain**: brain weight in g
- **name**: Common name of species. Rock hyrax-a = *Heterohyrax brucci*. Rock hyrax-b = *Procavia habessinica*.

Source


References

mca

Multiple Correspondence Analysis

Description

Computes a multiple correspondence analysis of a set of factors.

Usage

mca(df, nf = 2, abbrev = FALSE)

Arguments

df A data frame containing only factors

nf The number of dimensions for the MCA. Rarely 3 might be useful.

abbrev Should the vertex names be abbreviated? By default these are of the form 'factor.level' but if abbrev = TRUE they are just 'level' which will suffice if the factors have distinct levels.

Value

An object of class "mca", with components

rs The coordinates of the rows, in nf dimensions.

cs The coordinates of the column vertices, one for each level of each factor.

fs Weights for each row, used to interpolate additional factors in predict.mca.

p The number of factors

d The singular values for the nf dimensions.

call The matched call.

References


See Also

predict.mca, plot.mca, corresp

Examples

farms.mca <- mca(farms, abbrev=TRUE)
farms.mca
plot(farms.mca)
mcycle

Data from a Simulated Motorcycle Accident

Description
A data frame giving a series of measurements of head acceleration in a simulated motorcycle accident, used to test crash helmets.

Usage
mcycle

Format
- **times** in milliseconds after impact
- **accel** in g

Source

References

menarche

Age of Menarche data

Description
Proportions of female children at various ages during adolescence who have reached menarche.

Usage
menarche

Format
This data frame contains the following columns:

- **Age** Average age of the group. (The groups are reasonably age homogeneous.)
- **Total** Total number of children in the group.
- **Menarche** Number who have reached menarche.
Source


The data are also given in

References


Examples

```r
mprob <- glm(cbind(Menarche, Total - Menarche) ~ Age, binomial(link = probit), data = menarche)
```

---

**michelson**

*Michelson’s Speed of Light Data*

Description

Measurements of the speed of light in air, made between 5th June and 2nd July, 1879. The data consists of five experiments, each consisting of 20 consecutive runs. The response is the speed of light in km/s, less 299000. The currently accepted value, on this scale of measurement, is 734.5.

Usage

```r
michelson
```

Format

The data frame contains the following components:

- **Expt** The experiment number, from 1 to 5
- **Run** The run number within each experiment
- **Speed** Speed-of-light measurement

Source


References

**minn38**  
**Minnesota High School Graduates of 1938**

**Description**

The Minnesota high school graduates of 1938 were classified according to four factors, described below. The `minn38` data frame has 168 rows and 5 columns.

**Usage**

minn38

**Format**

This data frame contains the following columns:

- `hs` High school rank: "L", "M" and "U" for lower, middle and upper third.
- `phs` Post high school status: Enrolled in college, ("C"), enrolled in non-collegiate school, ("N"), employed full-time, ("E") and other, ("O").
- `fol` Father's occupational level, (seven levels, "F1", "F2", ... , "F7").
- `sex` Sex factor, "F" or "M".
- `f` Frequency.

**Source**


---

**motors**  
**Accelerated Life Testing of Motorettes**

**Description**

The `motors` data frame has 40 rows and 3 columns. It describes an accelerated life test at each of four temperatures of 10 motorettes, and has rather discrete times.

**Usage**

motors
Format

This data frame contains the following columns:

- **temp**: the temperature (degrees C) of the test
- **time**: the time in hours to failure or censoring at 8064 hours (= 336 days).
- **cens**: an indicator variable for death

Source


taken from


References


Examples

```r
library(survival)
plot(survfit(Surv(time, cens) ~ factor(temp), motors), conf.int = FALSE)
# fit Weibull model
motor.wei <- survreg(Surv(time, cens) ~ temp, motors)
summary(motor.wei)
# and predict at 130C
unlist(predict(motor.wei, data.frame(temp=130), se.fit = TRUE))

motor.cox <- coxph(Surv(time, cens) ~ temp, motors)
summary(motor.cox)
# predict at temperature 200
plot(survfit(motor.cox, newdata = data.frame(temp=200),
          conf.type = "log-log"))
summary( survfit(motor.cox, newdata = data.frame(temp=130)) )
```

---

**muscle**  

*Effect of Calcium Chloride on Muscle Contraction in Rat Hearts*

Description

The purpose of this experiment was to assess the influence of calcium in solution on the contraction of heart muscle in rats. The left auricle of 21 rat hearts was isolated and on several occasions a constant-length strip of tissue was electrically stimulated and dipped into various concentrations of calcium chloride solution, after which the shortening of the strip was accurately measured as the response.
Usage

muscle

Format

This data frame contains the following columns:

- **Strip** which heart muscle strip was used?
- **Conc** concentration of calcium chloride solution, in multiples of 2.2 mM.
- **Length** the change in length (shortening) of the strip, (allegedly) in mm.

Source


References


Examples

```r
A <- model.matrix(~ Strip - 1, data=muscle)
rats.nls1 <- nls(log(Length) ~ cbind(A, rho^Conc),
    data = muscle, start = c(rho=0.1), algorithm="plinear")
B <- coef(rats.nls1)
B

st <- list(alpha = B[2:22], beta = B[23], rho = B[1])
(rats.nls2 <- nls(log(Length) ~ alpha[Strip] + beta*rho^Conc,
    data = muscle, start = st))

attach(muscle)
Muscle <- expand.grid(Conc = sort(unique(Conc)),
    Strip = levels(Strip))
Muscle$Yhat <- predict(rats.nls2, Muscle)
Muscle <- cbind(Muscle, logLength = rep(as.numeric(NA), 126))
ind <- match(paste(Strip, Conc),
    paste(Muscle$Strip, Muscle$Conc))
Muscle$logLength[ind] <- log(Length)
detach()

require(lattice)
xyplot(Yhat ~ Conc | Strip, Muscle, as.table = TRUE,
    ylim = range(c(Muscle$Yhat, Muscle$logLength), na.rm = TRUE),
    subscripts = TRUE, xlab = "CalCloride concentration (mM)",
    ylab = "log(Length in mm)", panel =
    function(x, y, subscripts, ...) { lines(spline(x, y))
    panel.xyplot(x, Muscle$logLength[subscripts], ...)}
```
Simulate from a Multivariate Normal Distribution

Description

Produces one or more samples from the specified multivariate normal distribution.

Usage

mvrnorm(n = 1, mu, Sigma, tol = 1e-6, empirical = FALSE)

Arguments

- **n**: the number of samples required.
- **mu**: a vector giving the means of the variables.
- **Sigma**: a positive-definite symmetric matrix specifying the covariance matrix of the variables.
- **tol**: tolerance (relative to largest variance) for numerical lack of positive-definiteness in Sigma.
- **empirical**: logical. If true, mu and Sigma specify the empirical not population mean and covariance matrix.

Details

The matrix decomposition is done via `eigen`; although a Choleski decomposition might be faster, the eigendecomposition is stabler.

Value

If \( n = 1 \) a vector of the same length as \( \mu \), otherwise an \( n \) by \( \text{length}(\mu) \) matrix with one sample in each row.

Side Effects

Causes creation of the dataset `.Random.seed` if it does not already exist, otherwise its value is updated.

References


See Also

`rnorm`
negative.binomial

Family function for Negative Binomial GLMs

Description

Specifies the information required to fit a Negative Binomial generalized linear model, with known theta parameter, using glm().

Usage

negative.binomial(theta = stop("'theta' must be specified"), link = "log")

Arguments

  theta  The known value of the additional parameter, theta.
  link   The link function, as a character string, name or one-element character vector specifying one of log, sqrt or identity, or an object of class "link-glm".

Value

An object of class "family", a list of functions and expressions needed by glm() to fit a Negative Binomial generalized linear model.

References


See Also

glm.nb, anova.negbin, summary.negbin

Examples

# Fitting a Negative Binomial model to the quine data
# with theta = 2 assumed known.
#
# glm(Days ~ .^4, family = negative.binomial(2), data = quine)
**newcomb**  
*Newcomb’s Measurements of the Passage Time of Light*

**Description**

A numeric vector giving the Third Series of measurements of the passage time of light recorded by Newcomb in 1882. The given values divided by 1000 plus 24 give the time in millionths of a second for light to traverse a known distance. The 'true' value is now considered to be 33.02.

**Usage**

`newcomb`

**Source**


**nlschools**  
*Eighth-Grade Pupils in the Netherlands*

**Description**

Snijders and Bosker (1999) use as a running example a study of 2287 eighth-grade pupils (aged about 11) in 132 classes in 131 schools in the Netherlands. Only the variables used in our examples are supplied.

**Usage**

`nlschools`

**Format**

This data frame contains 2287 rows and the following columns:

- **lang**  language test score
- **IQ**  Verbal IQ
- **class**  class ID
- **GS**  Class size: number of eighth-grade pupils recorded in the class (there may be others: see **COMB**, and some may have been omitted with missing values).
- **SES**  Social-economic status of pupil’s family.

**COMB**  were the pupils taught in a multi-grade class (0/1)? Classes which contained pupils from grades 7 and 8 are coded 1, but only eighth-graders were tested.
源泉


参考文献


例子

```r
library(nlme)
nl1 <- nlschools
attach(nl1)
classMeans <- tapply(IQ, class, mean)
nl1$IQave <- classMeans[as.character(class)]
nl1$IQ <- nl1$IQ - nl1$IQave
detach()
cen <- c("IQ", "IQave", "SES")
nl1[cen] <- scale(nl1[cen], center = TRUE, scale = FALSE)

nl.lme <- lme(lang ~ IQ*COMB + IQave + SES,
              random = ~ IQ | class, data = nl1)
summary(nl.lme)
```

**npk**

*Classical N, P, K Factorial Experiment*

**描述**

一个经典的 N, P, K (氮, 磷, 钾) 因素实验, 用于豌豆的生长,在 6 块地中进行。每块田的一半是因子设计的, 用于 3 块田。每块田的 NPK 交互作用被保留在因子设计中。

**使用**

npk

**格式**

`npk` 数据框有 24 行和 5 列:

- **block** 哪一地块 (1 到 6)。
- **N** 氮的指示 (0/1)。
- **P** 磷的指示 (0/1)。
- **K** 钾的指示 (0/1)。
- **yield** 豌豆的产量, 每块田的磅 (由于每块田是 1/70 英亩)。
Source


References


Examples

```r
options(contrasts = c("contr.sum", "contr.poly"))
npk.aov <- aov(yield ~ block + N*P*K, npk)
npk.aov
summary(npk.aov)
alias(npk.aov)
coef(npk.aov)
options(contrasts = c("contr.treatment", "contr.poly"))
npk.aov1 <- aov(yield ~ block + N + K, data = npk)
summary.lm(npk.aov1)
se.contrast(npk.aov1, list(N=="0", N=="1"), data = npk)
model.tables(npk.aov1, type = "means", se = TRUE)
```

npr1

US Naval Petroleum Reserve No. 1 data

Description

Data on the locations, porosity and permeability (a measure of oil flow) on 104 oil wells in the US Naval Petroleum Reserve No. 1 in California.

Usage

npr1

Format

This data frame contains the following columns:

- **x**, **y** x and y coordinates, in miles from an arbitrary origin
- **perm** permeability in milli-Darcies
- **por** porosity (%)

Source


References

oats

Data from an Oats Field Trial

Description

The yield of oats from a split-plot field trial using three varieties and four levels of manurial treatment. The experiment was laid out in 6 blocks of 3 main plots, each split into 4 sub-plots. The varieties were applied to the main plots and the manurial treatments to the sub-plots.

Usage

oats

Format

This data frame contains the following columns:

B Blocks, levels I, II, III, IV, V and VI

V Varieties, 3 levels.

N Nitrogen (manurial) treatment, levels 0.0cwt, 0.2cwt, 0.4cwt and 0.6cwt, showing the application in cwt/acre.

Y Yields in 1/4lbs per sub-plot, each of area 1/80 acre.

Source


References


Examples

```r
oats$Nf <- ordered(oats$N, levels = sort(levels(oats$N)))
oats.aov <- aov(Y ~ Nf*V + Error(B/V), data = oats, qr = TRUE)
summary(oats.aov)
summary(oats.aov, split = list(Nf=list(L=1, Dev=2:3)))
par(mfrow = c(1,2), pty = "s")
plot(fitted(oats.aov[[4]]), studres(oats.aov[[4]]))
abline(h = 0, lty = 2)
oats.pr <- proj(oats.aov)
qqnorm(oats.pr[[4]][,"Residuals"], ylab = "Stratum 4 residuals")
qqline(oats.pr[[4]][,"Residuals"])

par(mfrow = c(1,1), pty = "m")
oats.aov2 <- aov(Y ~ N + V + Error(B/V), data = oats, qr = TRUE)
model.tables(oats.aov2, type = "means", se = TRUE)
```
The Painter’s Data of de Piles

Description

The subjective assessment, on a 0 to 20 integer scale, of 54 classical painters. The painters were assessed on four characteristics: composition, drawing, colour and expression. The data is due to the Eighteenth century art critic, de Piles.

Usage


format

The row names of the data frame are the painters. The components are:

- **Composition**: Composition score
- **Drawing**: Drawing score
- **Colour**: Colour score
- **Expression**: Expression score
- **School**: The school to which a painter belongs, as indicated by a factor level code as follows:
  - "A": Renaissance;
  - "B": Mannerist;
  - "C": Seicento;
  - "D": Venetian;
  - "E": Lombard;
  - "F": Sixteenth Century;
  - "G": Seventeenth Century;
  - "H": French.

Source


References

pairs.lda

Produce Pairwise Scatterplots from an 'lda' Fit

Description

Pairwise scatterplot of the data on the linear discriminants.

Usage

```r
## S3 method for class 'lda':
pairs(x, labels = colnames(x), panel = panel.lda,
    dimen, abbrev = FALSE, ..., cex=0.7, type = c("std", "trellis"))
```

Arguments

- `x`: Object of class "lda".
- `labels`: vector of character strings for labelling the variables.
- `panel`: panel function to plot the data in each panel.
- `dimen`: The number of linear discriminants to be used for the plot; if this exceeds the number determined by `x` the smaller value is used.
- `abbrev`: whether the group labels are abbreviated on the plots. If `abbrev > 0` this gives `minlength` in the call to `abbreviate`.
- `...`: additional arguments for `pairs.default`.
- `cex`: graphics parameter `cex` for labels on plots.
- `type`: type of plot. The default is in the style of `pairs.default`; the style "trellis" uses the Trellis function `splom`.

Details

This function is a method for the generic function `pairs()` for class "lda". It can be invoked by calling `pairs(x)` for an object `x` of the appropriate class, or directly by calling `pairs.lda(x)` regardless of the class of the object.

References


See Also

- `pairs`
parcoord  

Parallel Coordinates Plot

Description

Parallel coordinates plot

Usage

parcoord(x, col = 1, lty = 1, var.label = FALSE, ...)

Arguments

x  
a matrix or data frame who columns represent variables. Missing values are allowed.

col  
A vector of colours, recycled as necessary for each observation.

lty  
A vector of line types, recycled as necessary for each observation.

var.label  
If TRUE, each variable’s axis is labelled with maximum and minimum values.

...  
Further graphics parameters which are passed to matplot.

Side Effects

a parallel coordinates plots is drawn.

Author(s)

B. D. Ripley. Enhancements based on ideas and code by Fabian Scheipl.

References


Examples

data(state)
parcoord(state.x77[, c(7, 4, 6, 2, 5, 3)])

data(iris3)
ir <- rbind(iris3[,1], iris3[,2], iris3[,3])
parcoord(log(ir)[, c(3, 4, 2, 1)], col = 1 + (0:149)%/%50)
Description

The yield of a petroleum refining process with four covariates. The crude oil appears to come from only 10 distinct samples.

These data were originally used by Prater (1956) to build an estimation equation for the yield of the refining process of crude oil to gasoline.

Usage

petrol

Format

The variables are as follows

- **No** Crude oil sample identification label. (factor)
- **SG** Specific gravity, degrees API. (Constant within sample.)
- **VP** Vapour pressure in psi. (Constant within sample.)
- **V10** Volatility of crude; ASTM 10% point. (Constant within sample.)
- **EP** Desired volatility of gasoline. (The end point. Varies within sample.)
- **Y** Yield as a percentage of crude.

Source


This dataset is also given in D. J. Hand, F. Daly, K. McConway, D. Lunn and E. Ostrowski (eds) (1994) *A Handbook of Small Data Sets*. Chapman & Hall.

References


Examples

```r
library(nlme)
Petrol <- petrol
Petrol[, 2:5] <- scale(as.matrix(Petrol[, 2:5]), scale = FALSE)
pet3.lme <- lme(Y ~ SG + VP + V10 + EP,
               random = ~ 1 | No, data = Petrol)
pet3.lme <- update(pet3.lme, method = "ML")
pet4.lme <- update(pet3.lme, fixed = Y ~ V10 + EP)
anova(pet4.lme, pet3.lme)
```
Belgian-phones  

**Belgium Phone Calls 1950-1973**

**Description**

A list object with the annual numbers of telephone calls, in Belgium. The components are:

- **year** The last two digits of the year.
- **calls** The number of telephone calls made (in millions of calls).

**Usage**

```r
data(phones)
```

**Source**


**References**


---

**plot.lda**

**Plot Method for Class 'lda'**

**Description**

Plots a set of data on one, two or more linear discriminants.

**Usage**

```r
## S3 method for class 'lda':
plot(x, panel = panel.lda, ..., cex = 0.7, dimen,
     abbrev = FALSE, xlab = "LD1", ylab = "LD2")
```

**Arguments**

- **x** An object of class "lda".
- **panel** the panel function used to plot the data.
- **...** additional arguments to pairs, ldahist or eqscplot.
- **cex** graphics parameter cex for labels on plots.
- **dimen** The number of linear discriminants to be used for the plot; if this exceeds the number determined by x the smaller value is used.
- **abbrev** whether the group labels are abbreviated on the plots. If abbrev > 0 this gives minlength in the call to abbreviate.
- **xlab** label for the x axis
- **ylab** label for the y axis
Details

This function is a method for the generic function plot() for class "lda". It can be invoked by calling plot(x) for an object x of the appropriate class, or directly by calling plot.lda(x) regardless of the class of the object.

The behaviour is determined by the value of dimen. For dimen > 2, a pairs plot is used. For dimen = 2, an equiscaled scatter plot is drawn. For dimen = 1, a set of histograms or density plots are drawn. Use argument type to match "histogram" or "density" or "both".

References


See Also

pairs.lda, ldahist, lda, predict.lda

plot.mca

Plot Method for Objects of Class 'mca'

Description

Plot a multiple correspondence analysis.

Usage

## S3 method for class 'mca':
plot(x, rows = TRUE, col, cex = par("cex"), ...)  

Arguments

x : An object of class "mca".
rows : Should the coordinates for the rows be plotted, or just the vertices for the levels?
col, cex : The colours and cex to be used for the row points and level vertices respectively.
... : Additional parameters to plot.

References


See Also

mca, predict.mca

Examples

plot(mca(farms, abbrev = TRUE))
Ordered Logistic or Probit Regression

Description

Fits a logistic or probit regression model to an ordered factor response. The default logistic case is *proportional odds logistic regression*, after which the function is named.

Usage

```r
polr(formula, data, weights, start, ..., subset, na.action,
    contrasts = NULL, Hess = FALSE, model = TRUE,
    method = c("logistic", "probit", "cloglog", "cauchit"))
```

Arguments

- `formula`: a formula expression as for regression models, of the form `response ~ predictors`. The response should be a factor (preferably an ordered factor), which will be interpreted as an ordinal response, with levels ordered as in the factor. A proportional odds model will be fitted. The model must have an intercept: attempts to remove one will lead to a warning and be ignored. An offset may be used. See the documentation of `formula` for other details.
- `data`: an optional data frame in which to interpret the variables occurring in `formula`.
- `weights`: optional case weights in fitting. Default to 1.
- `start`: initial values for the parameters. This is in the format `c(coefficients, zeta)`: see the Values section.
- `...`: additional arguments to be passed to `optim`, most often a `control` argument.
- `subset`: expression saying which subset of the rows of the data should be used in the fit. All observations are included by default.
- `na.action`: a function to filter missing data.
- `contrasts`: a list of contrasts to be used for some or all of the factors appearing as variables in the model formula.
- `Hess`: logical for whether the Hessian (the observed information matrix) should be returned.
- `model`: logical for whether the model matrix should be returned.
- `method`: logistic or probit or complementary log-log or cauchit (corresponding to a Cauchy latent variable and only available in R >= 2.1.0).

Details

This model is what Agresti (2002) calls a *cumulative link* model. The basic interpretation is as a *coarsened* version of a latent variable $Y_i$ which has a logistic or normal or extreme-value or Cauchy
distribution with scale parameter one and a linear model for the mean. The ordered factor which is observed is which bin \( Y_i \) falls into with breakpoints

\[
\zeta_0 = -\infty < \zeta_1 < \cdots < \zeta_K = \infty
\]

This leads to the model

\[
\logit P(Y \leq k \mid x) = \zeta_k - \eta
\]

with \( \logit \) replaced by \( \text{probit} \) for a normal latent variable, and \( \eta \) being the linear predictor, a linear function of the explanatory variables (with no intercept). Note that it is quite common for other software to use the opposite sign for \( \eta \).

In the logistic case, the left-hand side of the last display is the log odds of category \( k \) or less, and since these are log odds which differ only by a constant for different \( k \), the odds are proportional. Hence the term \textit{proportional odds logistic regression}.

In the complementary log-log case, we have a \textit{proportional hazards} model for grouped survival times.

There are methods for the standard model-fitting functions, including \texttt{predict}, \texttt{summary}, \texttt{vcov}, \texttt{anova}, \texttt{model.frame} and an \texttt{extractAIC} method for use with \texttt{stepAIC}. There are also \texttt{profile} and \texttt{confint} methods.

**Value**

A object of class "polr". This has components

- \texttt{coefficients} the coefficients of the linear predictor, which has no intercept.
- \texttt{zeta} the intercepts for the class boundaries.
- \texttt{deviance} the residual deviance.
- \texttt{fitted.values} a matrix, with a column for each level of the response.
- \texttt{lev} the names of the response levels.
- \texttt{terms} the terms structure describing the model.
- \texttt{df.residual} the number of residual degrees of freedoms, calculated using the weights.
- \texttt{edf} the (effective) number of degrees of freedom used by the model
- \texttt{n, nobs} the (effective) number of observations, calculated using the weights. (\texttt{nobs} is for use by \texttt{stepAIC}.
- \texttt{call} the matched call.
- \texttt{method} the matched method used.
- \texttt{convergence} the convergence code returned by \texttt{optim}.
- \texttt{niter} the number of function and gradient evaluations used by \texttt{optim}.
- \texttt{Hessian} (if \texttt{Hess} is true).
- \texttt{model} (if \texttt{model} is true).

**References**


predict.glmmPQL

Predict Method for glmmPQL Fits

Description

Obtains predictions from a fitted generalized linear model with random effects.

Usage

## S3 method for class 'glmmPQL':
predict(object, newdata = NULL, type = c("link", "response"),
level, na.action = na.pass, ...)

Arguments

object a fitted object of class inheriting from "glmmPQL".
newdata optionally, a data frame in which to look for variables with which to predict.
type the type of prediction required. The default is on the scale of the linear predictors; the alternative "response" is on the scale of the response variable. Thus for a default binomial model the default predictions are of log-odds (probabilities on logit scale) and type = "response" gives the predicted probabilities.
predict.lda

level  an optional integer vector giving the level(s) of grouping to be used in obtaining the predictions. Level values increase from outermost to innermost grouping, with level zero corresponding to the population predictions. Defaults to the highest or innermost level of grouping.

na.action function determining what should be done with missing values in newdata. The default is to predict NA.

... further arguments passed to or from other methods.

Value

If level is a single integer, a vector otherwise a data frame.

See Also
glmmPQL, predict.lme.

Examples

fit <- glmmPQL(y ~ trt + I(week > 2), random = ~1 | ID, family = binomial, data = bacteria)
predict(fit, bacteria, level = 0, type="response")
predict(fit, bacteria, level = 1, type="response")

---

predict.lda  Classify Multivariate Observations by Linear Discrimination

Description

Classify multivariate observations in conjunction with lda, and also project data onto the linear discriminants.

Usage

## S3 method for class 'lda':
predict(object, newdata, prior = object$prior, dimen,
method = c("plug-in", "predictive", "debiased"), ...)

Arguments

object  object of class "lda"

newdata  data frame of cases to be classified, or, if object has a formula, a data frame with columns of the same names as the variables used. A vector will be interpreted as a row vector. If newdata is missing, an attempt will be made to retrieve the data used to fit the lda object.

prior  The prior probabilities of the classes, by default the proportions in the training set or what was set in the call to lda.
dimen the dimension of the space to be used. If this is less than \(\min(p, n_g-1)\), only the first \(\text{dimen}\) discriminant components are used (except for method="predictive"), and only those dimensions are returned in \(x\).

method This determines how the parameter estimation is handled. With "plug-in" (the default) the usual unbiased parameter estimates are used and assumed to be correct. With "debiased" an unbiased estimator of the log posterior probabilities is used, and with "predictive" the parameter estimates are integrated out using a vague prior.

... arguments based from or to other methods

Details

This function is a method for the generic function predict() for class "lda". It can be invoked by calling predict\((x)\) for an object \(x\) of the appropriate class, or directly by calling predict.lda\((x)\) regardless of the class of the object.

Missing values in newdata are handled by returning NA if the linear discriminants cannot be evaluated. If newdata is omitted and the na.action of the fit omitted cases, these will be omitted on the prediction.

This version centres the linear discriminants so that the weighted mean (weighted by prior) of the group centroids is at the origin.

Value

a list with components

- class The MAP classification (a factor)
- posterior posterior probabilities for the classes
- \(x\) the scores of test cases on up to \(\text{dimen}\) discriminant variables

References


See Also

lda, qda, predict.qda

Examples

data(iris3)
tr <- sample(1:50, 25)
train <- rbind(iris3[,1], iris3[,2], iris3[,3])
test <- rbind(iris3[-tr,1], iris3[-tr,2], iris3[-tr,3])
c1 <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
z <- lda(train, c1)
predict(z, test)$class
predict.lqs

Predict from an lqs Fit

Description

Predict from an resistant regression fitted by lqs.

Usage

## S3 method for class 'lqs':
predict(object, newdata, na.action = na.pass, ...)

Arguments

- **object**: object inheriting from class "lqs"
- **newdata**: matrix or data frame of cases to be predicted or, if object has a formula, a data frame with columns of the same names as the variables used. A vector will be interpreted as a row vector. If newdata is missing, an attempt will be made to retrieve the data used to fit the lqs object.
- **na.action**: function determining what should be done with missing values in newdata. The default is to predict NA.
- **...**: arguments to be passed from or to other methods.

Details

This function is a method for the generic function predict() for class lqs. It can be invoked by calling predict(x) for an object x of the appropriate class, or directly by calling predict.lqs(x) regardless of the class of the object.

Missing values in newdata are handled by returning NA if the linear fit cannot be evaluated. If newdata is omitted and the na.action of the fit omitted cases, these will be omitted on the prediction.

Value

A vector of predictions.

Author(s)

B.D. Ripley

See Also

lqs
Examples

data(stackloss)
set.seed(123)
fm <- lqs(stack.loss ~ ., data = stackloss, method = "S", nsamp = "exact")
predict(fm, stackloss)

predict.mca  

Predict Method for Class 'mca'

Description

Used to compute coordinates for additional rows or additional factors in a multiple correspondence analysis.

Usage

## S3 method for class 'mca':
predict(object, newdata, type = c("row", "factor"), ...)

Arguments

object  
An object of class "mca", usually the result of a call to mca.

newdata  
A data frame containing either additional rows of the factors used to fit object or additional factors for the cases used in the original fit.

type  
Are predictions required for further rows or for new factors?

Value

If type = "row", the coordinates for the additional rows.
If type = "factor", the coordinates of the column vertices for the levels of the new factors.

References


See Also

mca, plot.mca
**predict.qda**  
*Classify from Quadratic Discriminant Analysis*

**Description**

Classify multivariate observations in conjunction with qda

**Usage**

```r
## S3 method for class 'qda':
predict(object, newdata, prior = object$prior,
       method = c("plug-in", "predictive", "debiased", "looCV"), ...)
```

**Arguments**

- `object`: object of class "qda"
- `newdata`: data frame of cases to be classified or, if `object` has a formula, a data frame with columns of the same names as the variables used. A vector will be interpreted as a row vector. If `newdata` is missing, an attempt will be made to retrieve the data used to fit the `qda` object.
- `prior`: The prior probabilities of the classes, by default the proportions in the training set or what was set in the call to `qda`.
- `method`: This determines how the parameter estimation is handled. With "plug-in" (the default) the usual unbiased parameter estimates are used and assumed to be correct. With "debiased" an unbiased estimator of the log posterior probabilities is used, and with "predictive" the parameter estimates are integrated out using a vague prior. With "looCV" the leave-one-out cross-validation fits to the original dataset are computed and returned.

**Details**

This function is a method for the generic function `predict()` for class "qda". It can be invoked by calling `predict(x)` for an object `x` of the appropriate class, or directly by calling `predict.qda(x)` regardless of the class of the object.

Missing values in `newdata` are handled by returning `NA` if the quadratic discriminants cannot be evaluated. If `newdata` is omitted and the `na.action` of the fit omitted cases, these will be omitted on the prediction.

**Value**

A list with components

- `class`: The MAP classification (a factor)
- `posterior`: posterior probabilities for the classes
qda

Quadratic Discriminant Analysis

Description

Quadratic discriminant analysis.

Usage

qda(x, ...)

## S3 method for class 'formula':
qda(formula, data, ..., subset, na.action)

## Default S3 method:
qda(x, grouping, prior = proportions,
method, CV = FALSE, nu, ...)

## S3 method for class 'data.frame':
qda(x, ...)

## S3 method for class 'matrix':
qda(x, grouping, ..., subset, na.action)

Arguments

formula A formula of the form groups ~ x1 + x2 + ... That is, the response is
the grouping factor and the right hand side specifies the (non-factor) discriminators.

Examples

data(iris3)
tr <- sample(1:50, 25)
train <- rbind(iris3[tr,,1], iris3[tr,,2], iris3[tr,,3])
test <- rbind(iris3[-tr,,1], iris3[-tr,,2], iris3[-tr,,3])
cl <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
zq <- qda(train, cl)
predict(zq, test)$class
qda

data Data frame from which variables specified in formula are preferentially to be taken.
x (required if no formula is given as the principal argument.) a matrix or data frame or Matrix containing the explanatory variables.
grouping (required if no formula principal argument is given.) a factor specifying the class for each observation.
prior the prior probabilities of class membership. If unspecified, the class proportions for the training set are used. If specified, the probabilities should be specified in the order of the factor levels.
subset An index vector specifying the cases to be used in the training sample. (NOTE: If given, this argument must be named.)
na.action A function to specify the action to be taken if NAs are found. The default action is for the procedure to fail. An alternative is na.omit, which leads to rejection of cases with missing values on any required variable. (NOTE: If given, this argument must be named.)
method "moment" for standard estimators of the mean and variance, "mle" for MLEs, "mve" to use cov.mve, or "t" for robust estimates based on a t distribution.
CV If true, returns results (classes and posterior probabilities) for leave-out-out cross-validation. Note that if the prior is estimated, the proportions in the whole dataset are used.
nu degrees of freedom for method = "t".
... arguments passed to or from other methods.

Details

Uses a QR decomposition which will give an error message if the within-group variance is singular for any group.

Value

an object of class "qda" containing the following components:

prior the prior probabilities used.
means the group means.
scaling for each group i, scaling[,i] is an array which transforms observations so that within-groups covariance matrix is spherical.
ldet a vector of half log determinants of the dispersion matrix.
lev the levels of the grouping factor.
terms (if formula is a formula) an object of mode expression and class term summarizing the formula.
call the (matched) function call.
class The MAP classification (a factor)
posterior posterior probabilities for the classes
References


See Also

predict.qda, lda

Examples

data(iris3)
tr <- sample(1:50, 25)
train <- rbind(iris3[tr,,1], iris3[tr,,2], iris3[tr,,3])
test <- rbind(iris3[-tr,,1], iris3[-tr,,2], iris3[-tr,,3])
c1 <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
z <- qda(train, cl)
predict(z,test)$class

---

quine

Absenteeism from School in Rural New South Wales

Description

The *quine* data frame has 146 rows and 5 columns. Children from Walgett, New South Wales, Australia, were classified by Culture, Age, Sex and Learner status and the number of days absent from school in a particular school year was recorded.

Usage

quine

Format

This data frame contains the following columns:

- **Eth** Ethnic background: Aboriginal or Not, ("A" or "N").
- **Sex** Sex factor: ("F" or "M").
- **Age** Age group: Primary ("F0"), or forms "F1," "F2" or "F3".
- **Lrn** Learner status factor: Average or Slow learner, ("AL" or "SL").
- **Days** Days absent from school in the year.

Source


References

**rational**  

**Rational Approximation**

**Description**
Find rational approximations to the components of a real numeric object using a standard continued fraction method.

**Usage**
rational(x, cycles = 10, max.denominator = 2000, ...)

**Arguments**
x  
Any object of mode numeric. Missing values are now allowed.

cycles  
The maximum number of steps to be used in the continued fraction approximation process.

max.denominator  
An early termination criterion. If any partial denominator exceeds max.denominator the continued fraction stops at that point.

...  
arguments passed to or from other methods.

**Details**
Each component is first expanded in a continued fraction of the form
\[ x = \text{floor}(x) + \frac{1}{(p_1 + \frac{1}{(p_2 + \ldots)})} \]
where \( p_1, p_2, \ldots \) are positive integers, terminating either at cycles terms or when a \( p_j \) > max.denominator. The continued fraction is then re-arranged to retrieve the numerator and denominator as integers and the ratio returned as the value.

**Value**
A numeric object with the same attributes as \( x \) but with entries rational approximations to the values. This effectively rounds relative to the size of the object and replaces very small entries by zero.

**See Also**
fractions

**Examples**
X <- matrix(runif(25), 5, 5)
solve(X, X/5)
## [1,] 2.0000e-01 3.7199e-17 1.2214e-16 5.7887e-17 -8.7841e-17
## [2,] -1.1473e-16 2.0000e-01 7.0955e-17 2.0300e-17 -1.0566e-16
## [3,] 2.7975e-16 1.3653e-17 2.0000e-01 -1.3397e-16 1.5577e-16
renumerate

Convert a Formula Transformed by 'denumerate'

Description

`denumerate` converts a formula written using the conventions of `loglm` into one that `terms` is able to process. `renumerate` converts it back again to a form like the original.

Usage

`renumerate(x)`

Arguments

x A formula, normally as modified by `denumerate`.

Details

This is an inverse function to `denumerate`. It is only needed since `terms` returns an expanded form of the original formula where the non-marginal terms are exposed. This expanded form is mapped back into a form corresponding to the one that the user originally supplied.

Value

A formula where all variables with names of the form `.vn`, where `n` is an integer, converted to numbers, `n`, as allowed by the formula conventions of `loglm`.

See Also

`denumerate`

Examples

denumerate(~(1+2+3)^3 + a/b)
## ~(1 + 2 + 3)^3 + a/b
renumerate(.Last.value)
## ~ (1 + 2 + 3)^3 + a/b
**Description**

Fit a linear model by robust regression using an M estimator.

**Usage**

```r
rlm(x, ...)```

## S3 method for class 'formula':
```
rlm(formula, data, weights, ..., subset, na.action,
    method = c("M", "MM", "model.frame"),
    wt.method = c("inv.var", "case"),
    model = TRUE, x.ret = TRUE, y.ret = FALSE, contrasts = NULL)
```

## Default S3 method:  
```
rlm(x, y, weights, ..., w = rep(1, nrow(x)),
    init, psi = psi.huber, scale.est, k2 = 1.345,
    method = c("M", "MM"), wt.method = c("inv.var", "case"),
    maxit = 20, acc = 1e-4, test.vec = "resid", lqs.control = NULL)
```

```r
psi.huber(u, k = 1.345, deriv = 0)
psi.hampel(u, a = 2, b = 4, c = 8, deriv = 0)
psi.bisquare(u, c = 4.685, deriv = 0)
```

**Arguments**

- `formula` a formula of the form `y ~ x1 + x2 + ....`
- `data` data frame from which variables specified in `formula` are preferentially to be taken.
- `weights` a vector of prior weights for each case.
- `subset` An index vector specifying the cases to be used in fitting.
- `na.action` A function to specify the action to be taken if NAs are found. The default action is for the procedure to fail. An alternative is `na.omit`, which leads to omission of cases with missing values on any required variable.
- `x` a matrix or data frame containing the explanatory variables.
- `y` the response: a vector of length the number of rows of `x`.
- `method` currently either M-estimation or find the model frame. MM estimation is M-estimation with Tukey’s biweight initialized by a specific S-estimator. See the details section.
- `wt.method` are the weights case weights (giving the relative importance of case, so a weight of 2 means there are two of these) or the inverse of the variances, so a weight of two means this error is half as variable?
model should the model frame be returned in the object?
x.ret should the model matrix be returned in the object?
y.ret should the response be returned in the object?
contrasts optional contrast specifications: see `lm`.
w (optional) initial down-weighting for each case.
init (optional) initial values for the coefficients OR a method to find initial values OR the result of a fit with a `coef` component. Known methods are "ls" (the default) for an initial least-squares fit using weights $w \cdot \text{weights}$, and "lts" for an unweighted least-trimmed squares fit with 200 samples.
psi the psi function is specified by this argument. It must give (possibly by name) a function $g(x, \ldots, \text{deriv})$ that for deriv=0 returns $\psi(x)/x$ and for deriv=1 returns $\psi'(x)$. Tuning constants will be passed in via ....
scale.est method of scale estimation: re-scaled MAD of the residuals or Huber’s proposal 2 (which can be selected by either "Huber" or "proposal 2").
k2 tuning constant used for Huber proposal 2 scale estimation.
maxit the limit on the number of IWLS iterations.
acc the accuracy for the stopping criterion.
test.vec the stopping criterion is based on changes in this vector.
... additional arguments to be passed to `rlm.default` or to the `psi` function.
lqs.control An optional list of control values for `lqs`.
u numeric vector of evaluation points.
k,a,b,c tuning constants
deriv 0 or 1: compute values of the psi function or of its first derivative.

Details

Fitting is done by iterated re-weighted least squares (IWLS).
Psi functions are supplied for the Huber, Hampel and Tukey bisquare proposals as `psi.huber`, `psi.hampel` and `psi.bisquare`. Huber’s corresponds to a convex optimization problem and gives a unique solution (up to collinearity). The other two will have multiple local minima, and a good starting point is desirable.

Selecting `method = "MM"` selects a specific set of options which ensures that the estimator has a high breakdown point. The initial set of coefficients and the final scale are selected by an S-estimator with $k0 = 1.548$; this gives (for $n \gg p$) breakdown point 0.5. The final estimator is an M-estimator with Tukey’s biweight and fixed scale that will inherit this breakdown point provided $c > k0$; this is true for the default value of `c` that corresponds to 95% relative efficiency at the normal. Case weights are not supported for `method = "MM"`.

Value

An object of class "rlm" inheriting from "lm". The additional components not in an `lm` object are

s the robust scale estimate used
rms.curv

w
the weights used in the IWLS process

psi
the psi function with parameters substituted

conv
the convergence criteria at each iteration

converged
did the IWLS converge?

wresid
a working residual, weighted for "inv.var" weights only.

References


See Also

lm, lqs.

Examples

data(stackloss)
summary(rlm(stack.loss ~ ., stackloss))
rlm(stack.loss ~ ., stackloss, psi = psi.hampel, init = "lts")
rlm(stack.loss ~ ., stackloss, psi = psi.bisquare)

rms.curv

Relative Curvature Measures for Non-Linear Regression

Description

Calculates the root mean square parameter effects and intrinsic relative curvatures, $c^0$ and $c^i$, for a fitted nonlinear regression, as defined in Bates & Watts, section 7.3, p. 253 et seq.

Usage

rms.curv(obj)

Arguments

obj
Fitted model object of class "nls". The model must be fitted using the default algorithm.
Details

The method of section 7.3.1 of Bates & Watts is implemented. The function deriv3 should be used to generate a model function with first derivative (gradient) matrix and second derivative (Hessian) array attributes. This function should then be used to fit the nonlinear regression model.

A print method, print.rms.curv, prints the pc and ic components only, suitably annotated. If either pc or ic exceeds some threshold (0.3 has been suggested) the curvature is unacceptably high for the planar assumption.

Value

A list of class rms.curv with components pc and ic for parameter effects and intrinsic relative curvatures multiplied by sqrt(F), ct and ci for c^\theta and c^\iota (unmultiplied), and C the C-array as used in section 7.3.1 of Bates & Watts.

References


See Also

deriv3

Examples

# The treated sample from the Puromycin data
data(Puromycin)
mmcurve <- deriv3(~ Vm * conc/(K + conc), c("Vm", "K"),
    function(Vm, K, conc) NULL)
Treated <- Puromycin[Puromycin$state == "treated", ]
(Purfit1 <- nls(rate ~ mmcurve(Vm, K, conc), data = Treated,
    start = list(Vm=200, K=0.1)))
rms.curv(Purfit1)
##Parameter effects: c^\theta x sqrt(F) = 0.2121
## Intrinsic: c^iota x sqrt(F) = 0.092

rnegbin

Simulate Negative Binomial Variates

Description

Function to generate random outcomes from a Negative Binomial distribution, with mean mu and variance mu + mu^2/theta.

Usage

rnegbin(n, mu = n, theta = stop("'theta' must be specified"))
Arguments

- `n`: If a scalar, the number of sample values required. If a vector, `length(n)` is the number required and `n` is used as the mean vector if `mu` is not specified.
- `mu`: The vector of means. Short vectors are recycled.
- `theta`: Vector of values of the `theta` parameter. Short vectors are recycled.

Details

The function uses the representation of the Negative Binomial distribution as a continuous mixture of Poisson distributions with Gamma distributed means. Unlike `rnbinom` the index can be arbitrary.

Value

Vector of random Negative Binomial variate values.

Side Effects

Changes `.Random.seed` in the usual way.

Examples

```r
# Negative Binomials with means fitted(fm) and theta = 4.5
fm <- glm.nb(Days ~ ., data = quine)
dummy <- rnegbin(fitted(fm), theta = 4.5)
```

---

Road Accident Deaths in US States

Description

A data frame with the annual deaths in road accidents for half the US states. Components are:

Usage

`road`

Format

- `state` name
- `deaths` number of deaths
- `drivers` number of drivers (in 10,000's)
- `popden` population density in people per square mile
- `rural` length of rural roads, in 1000's of miles
- `temp` average daily maximum temperature in January
- `fuel` fuel consumption in 10,000,000 US gallons per year
Source

Imperial College, London M.Sc. exercise

---

rotifer  
*Numbers of Rotifers by Fluid Density*

Description

The data give the numbers of rotifers falling out of suspension for different fluid densities. There are two species, \( Pm \) *Polyarthra major* and \( Kc \), *Keratella cochlearis* and for each species the number falling out and the total number are given.

Usage

`rotifer`

Format

- **density**: specific density of fluid
- **pm.y**: number falling out for *P. major*
- **pm.total**: total number of *P. major*
- **kc.y**: number falling out for *K. cochlearis*
- **kc.tot**: total number of *K. cochlearis*

Source


---

sammon  
*Sammon’s Non-Linear Mapping*

Description

One form of non-metric multidimensional scaling.

Usage

```r
sammon(d, y = cmdscale(d, k), k = 2, niter = 100, trace = TRUE,
magic = 0.2, tol = 1e-4)
```
Arguments

d  distance structure of the form returned by \texttt{dist}, or a full, symmetric matrix. Data are assumed to be dissimilarities or relative distances, but must be positive except for self-distance. This can contain missing values.

y  An initial configuration. If none is supplied, \texttt{cmdscale} is used to provide the classical solution. (If there are missing values in \texttt{d}, an initial configuration must be provided.) This must not have duplicates.

k  The dimension of the configuration.

niter  The maximum number of iterations.

trace  Logical for tracing optimization. Default \texttt{TRUE}.

magic  initial value of the step size constant in diagonal Newton method.

tol  Tolerance for stopping, in units of stress.

Details

This chooses a two-dimensional configuration to minimize the stress, the sum of squared differences between the input distances and those of the configuration, weighted by the distances, the whole sum being divided by the sum of input distances to make the stress scale-free.

An iterative algorithm is used, which will usually converge in around 50 iterations. As this is necessarily an \(O(n^2)\) calculation, it is slow for large datasets. Further, since the configuration is only determined up to rotations and reflections (by convention the centroid is at the origin), the result can vary considerably from machine to machine. In this release the algorithm has been modified by adding a step-length search (\texttt{magic}) to ensure that it always goes downhill.

Value

Two components:

points  A two-column vector of the fitted configuration.

stress  The final stress achieved.

Side Effects

If trace is true, the initial stress and the current stress are printed out every 10 iterations.

References


See Also

\texttt{cmdscale}, \texttt{isoMDS}
Examples

data(swiss)
swiss.x <- as.matrix(swiss[, -1])
swiss.sam <- sammon(dist(swiss.x))
plot(swiss.sam$points, type = "n")
text(swiss.sam$points, labels = as.character(1:nrow(swiss.x)))

ships

Ships Damage Data

Description

Data frame giving the number of damage incidents and aggregate months of service by ship type, year of construction, and period of operation.

Usage

ships

Format

type type: "A" to "E"
year year of construction: 1960-64, 65-69, 70-74, 75-79 (coded as "60", "65", "70", "75")
period period of operation: 1960-74, 75-79
service aggregate months of service
incidents number of damage incidents

Source


shoes

Shoe wear data of Box, Hunter and Hunter

Description

A list of two vectors, giving the wear of shoes of materials A and B for one foot each of ten boys.

Usage

shoes
**shrimp**

**Source**


**References**


---

<table>
<thead>
<tr>
<th>shrimp</th>
<th>Percentage of Shrimp in Shrimp Cocktail</th>
</tr>
</thead>
</table>

**Description**

A numeric vector with 18 determinations by different laboratories of the amount (percentage of the declared total weight) of shrimp in shrimp cocktail.

**Usage**

```
shrimp
```

**Source**


---

<table>
<thead>
<tr>
<th>shuttle</th>
<th>Space Shuttle Autolander Problem</th>
</tr>
</thead>
</table>

**Description**

The `shuttle` data frame has 256 rows and 7 columns. The first six columns are categorical variables giving example conditions; the seventh is the decision. The first 253 rows are the training set, the last 3 the test conditions.

**Usage**

```
shuttle
```
Format

This data frame contains the following columns:

- **stability** Stable positioning or not (stab / xstab)
- **error** Size of error (MM / SS / LX / XL)
- **sign** Sign of error, positive or negative (pp / nn)
- **wind** Wind sign (head / tail)
- **magn** Wind strength (Light / Medium / Strong / Out of Range)
- **vis** Visibility (yes / no)
- **use** Use the autolander or not

Source


References


---

**snails**  
*Snail Mortality Data*

Description

Groups of 20 snails were held for periods of 1, 2, 3 or 4 weeks in carefully controlled conditions of temperature and relative humidity. There were two species of snail, A and B, and the experiment was designed as a 4 by 3 by 4 by 2 completely randomized design. At the end of the exposure time the snails were tested to see if they had survived; the process itself is fatal for the animals. The object of the exercise was to model the probability of survival in terms of the stimulus variables, and in particular to test for differences between species.

The data are unusual in that in most cases fatalities during the experiment were fairly small.

Usage

snails

Format

The data frame contains the following components:

- **Species** Snail species A (1) or B (2)
- **Exposure** Exposure in weeks
- **Rel.Hum** Relative humidity (4 levels)
- **Temp** Temperature, in degrees Celsius (3 levels)
- **Deaths** Number of deaths
- **N** Number of snails exposed
**stdres**  

Extract Standardized Residuals from a Linear Model

**Description**

The standardized residuals. These are normalized to unit variance, fitted including the current data point.

**Usage**

```r
stdres(object)
```

**Arguments**

- `object` any object representing a linear model.

**Value**

The vector of appropriately transformed residuals.

**References**

steam  
*The Saturated Steam Pressure Data*

**Description**

Temperature and pressure in a saturated steam driven experimental device.

**Usage**

```r
steam
```

**Format**

The data frame contains the following components:

- **Temp**  Temperature, in degrees Celsius
- **Press** Pressure, in Pascals

**Source**


**References**


stepAIC  
*Choose a model by AIC in a Stepwise Algorithm*

**Description**

Performs stepwise model selection by AIC.

**Usage**

```r
stepAIC(object, scope, scale = 0,
        direction = c("both", "backward", "forward"),
        trace = 1, keep = NULL, steps = 1000, use.start = FALSE,
        k = 2, ...)
```
**Arguments**

- **object**
  An object representing a model of an appropriate class. This is used as the initial model in the stepwise search.

- **scope**
  Defines the range of models examined in the stepwise search. This should be either a single formula, or a list containing components `upper` and `lower`, both formulae. See the details for how to specify the formulae and how they are used.

- **scale**
  Used in the definition of the AIC statistic for selecting the models, currently only for `lm` and `aov` models (see `extractAIC` for details).

- **direction**
  The mode of stepwise search, can be one of "both", "backward", or "forward", with a default of "both". If the `scope` argument is missing the default for `direction` is "backward".

- **trace**
  If positive, information is printed during the running of `stepAIC`. Larger values may give more information on the fitting process.

- **keep**
  A filter function whose input is a fitted model object and the associated AIC statistic, and whose output is arbitrary. Typically `keep` will select a subset of the components of the object and return them. The default is not to keep anything.

- **steps**
  The maximum number of steps to be considered. The default is 1000 (essentially as many as required). It is typically used to stop the process early.

- **use.start**
  If true the updated fits are done starting at the linear predictor for the currently selected model. This may speed up the iterative calculations for `glm` (and other fits), but it can also slow them down. **Not used in R.**

- **k**
  The multiple of the number of degrees of freedom used for the penalty. Only `k = 2` gives the genuine AIC: `k = log(n)` is sometimes referred to as BIC or SBC.

- ... any additional arguments to `extractAIC`. (None are currently used.)

**Details**

The set of models searched is determined by the `scope` argument. The right-hand-side of its `lower` component is always included in the model, and right-hand-side of the model is included in the `upper` component. If `scope` is a single formula, it specifies the `upper` component, and the `lower` model is empty. If `scope` is missing, the initial model is used as the `upper` model.

Models specified by `scope` can be templates to update `object` as used by `update.formula`.

There is a potential problem in using `glm` fits with a variable `scale`, as in that case the deviance is not simply related to the maximized log-likelihood. The `glm` method for `extractAIC` makes the appropriate adjustment for a gaussian family, but may need to be amended for other cases. (**The binomial and poisson families have fixed `scale` by default and do not correspond to a particular maximum-likelihood problem for variable `scale`.**)

Where a conventional deviance exists (e.g. for `lm`, `aov` and `glm` fits) this is quoted in the analysis of variance table: it is the **unscaled** deviance.
Value

the stepwise-selected model is returned, with up to two additional components. There is an "anova" component corresponding to the steps taken in the search, as well as a "keep" component if the keep= argument was supplied in the call. The "Resid. Dev" column of the analysis of deviance table refers to a constant minus twice the maximized log likelihood: it will be a deviance only in cases where a saturated model is well-defined (thus excluding \texttt{lm}, \texttt{aov} and \texttt{survreg} fits, for example).

Note

The model fitting must apply the models to the same dataset. This may be a problem if there are missing values and an \texttt{na.action} other than \texttt{na.fail} is used (as is the default in \texttt{R}). We suggest you remove the missing values first.

References


See Also

\texttt{addterm}, \texttt{dropterm}, \texttt{step}

Examples

\begin{verbatim}
quine.hi <- aov(log(Days + 2.5) ~ .^4, quine)
quine.nxt <- update(quine.hi, . ~ . - Eth:Sex:Age:Lrn)
quine.stp <- stepAIC(quine.nxt,
    scope = list(upper = ~Eth*Sex*Age*Lrn, lower = ~1),
    trace = FALSE)
quine.stp$anova

cpus1 <- cpus
attach(cpus)
for(v in names(cpus)[2:7])
    cpus1[[v]] <- cut(cpus[[v]], unique(quantile(cpus[[v]]))
    include.lowest = TRUE)
detach()
cpus0 <- cpus1[,2:8] # excludes names, authors' predictions
cpus.samp <- sample(1:209, 100)
cpus.lm <- lm(log10(perf) ~ ., data = cpus1[cpus.samp,2:8])
cpus.lm2 <- stepAIC(cpus.lm, trace = FALSE)
cpus.lm2$anova

example(birthwt)
birthwt.glm <- glm(low ~ ., family = binomial, data = bwt)
birthwt.step <- stepAIC(birthwt.glm, trace = FALSE)
birthwt.step$anova

birthwt.step2 <- stepAIC(birthwt.glm, ~ .^2 + I(scale(age)^2)
    + I(scale(lwt)^2), trace = FALSE)
birthwt.step2$anova
\end{verbatim}
quine.nb <- glm.nb(Days ~ .^4, data = quine)
quine.nb2 <- stepAIC(quine.nb)
quine.nb2$anova

---

The Stormer Viscometer Data

Description

The stormer viscometer measures the viscosity of a fluid by measuring the time taken for an inner cylinder in the mechanism to perform a fixed number of revolutions in response to an actuating weight. The viscometer is calibrated by measuring the time taken with varying weights while the mechanism is suspended in fluids of accurately known viscosity. The data comes from such a calibration, and theoretical considerations suggest a nonlinear relationship between time, weight and viscosity, of the form \(\text{Time} = \frac{(B_1 \times \text{Viscosity})}{(\text{Weight} - B_2)} + E\) where \(B_1\) and \(B_2\) are unknown parameters to be estimated, and \(E\) is error.

Usage

stormer

Format

The data frame contains the following components:

- **Viscosity** Viscosity of fluid
- **Wt** Actuating weight
- **Time** Time taken

Source


References

studres

Extract Studentized Residuals from a Linear Model

Description

The Studentized residuals. Like standardized residuals, these are normalized to unit variance, but the Studentized version is fitted ignoring the current data point. (They are sometimes called jackknifed residuals).

Usage

studres(object)

Arguments

object any object representing a linear model.

Value

The vector of appropriately transformed residuals.

References


See Also

residuals, stdres

summary.loglm

Summary Method Function for Objects of Class 'loglm'

Description

Returns a summary list for log-linear models fitted by iterative proportional scaling using loglm.

Usage

## S3 method for class 'loglm':
summary(object, fitted = FALSE, ...)

Arguments

object a fitted loglm model object.
fitted if TRUE return observed and expected frequencies in the result. Using fitted = TRUE may necessitate re-fitting the object.
... arguments to be passed to or from other methods.
summary.negbin

Details

This function is a method for the generic function summary() for class "loglm". It can be invoked by calling summary(x) for an object x of the appropriate class, or directly by calling summary.loglm(x) regardless of the class of the object.

Value

A list is returned for use by print.summary.loglm. This has components

- formula: the formula used to produce object
- tests: the table of test statistics (likelihood ratio, Pearson) for the fit.
- oe: if fitted = TRUE, an array of the observed and expected frequencies, otherwise NULL.

References


See Also

loglm, summary

summary.negbin  Summary Method Function for Objects of Class 'negbin'

Description

Identical to summary.glm, but with three lines of additional output: the ML estimate of theta, its standard error, and twice the log-likelihood function.

Usage

## S3 method for class 'negbin':
summary(object, dispersion = 1, correlation = FALSE, ...)

Arguments

- object: Fitted model object of class negbin inheriting from glm and lm. Typically the output of glm.nb.
- dispersion: as for summary.glm, with a default of 1.
- correlation: as for summary.glm.
- ...: arguments passed to or from other methods.
Details

`summary.glm` is used to produce the majority of the output and supply the result. This function is a method for the generic function `summary()` for class "negbin". It can be invoked by calling `summary(x)` for an object `x` of the appropriate class, or directly by calling `summary.negbin(x)` regardless of the class of the object.

Value

As for `summary.glm`; the additional lines of output are not included in the resultant object.

Side Effects

A summary table is produced as for `summary.glm`, with the additional information described above.

References


See Also

`summary, glm.nb, negative.binomial, anova.negbin`

Examples

```r
summary(glm.nb(Days ~ Eth*Age*Lrn*Sex, quine, link = log))
```

summary.rlm  

**Summary Method for Robust Linear Models**

Description

`summary` method for objects of class "rlm"

Usage

```r
## S3 method for class 'rlm':
summary(object, method = c("XtX", "XtWX"), correlation = FALSE, ...)
```

Arguments

- `object`: the fitted model. This is assumed to be the result of some fit that produces an object inheriting from the class `rlm`, in the sense that the components returned by the `rlm` function will be available.
- `method`: Should the weighted (by the IWLS weights) or unweighted cross-products matrix be used?
- `correlation`: logical. Should correlations be computed (and printed)?
- `...`: arguments passed to or from other methods.
Details

This function is a method for the generic function `summary()` for class "rlm". It can be invoked by calling `summary(x)` for an object `x` of the appropriate class, or directly by calling `summary.rlm(x)` regardless of the class of the object.

Value

If printing takes place, only a null value is returned. Otherwise, a list is returned with the following components. Printing always takes place if this function is invoked automatically as a method for the `summary` function.

- `correlation`: The computed correlation coefficient matrix for the coefficients in the model.
- `cov.unscaled`: The unscaled covariance matrix; i.e., a matrix such that multiplying it by an estimate of the error variance produces an estimated covariance matrix for the coefficients.
- `sigma`: The scale estimate.
- `stddev`: A scale estimate used for the standard errors.
- `df`: The number of degrees of freedom for the model and for residuals.
- `coefficients`: A matrix with three columns, containing the coefficients, their standard errors and the corresponding t statistic.
- `terms`: The terms object used in fitting this model.

References


See Also

`summary`

Examples

```r
summary(rlm(calls ~ year, data = phones, maxit = 50))
```

```
## Not run:
Call:
rlm(formula = calls ~ year, data = phones, maxit = 50)

Residuals:
  Min 1Q Median 3Q Max
-18.31 -5.95 -1.68  26.46 173.77

Coefficients:
              Value Std. Error t value
(Intercept) -102.622  26.553   -3.86
year       2.041   0.429    4.76

Residual standard error: 9.03 on 22 degrees of freedom
Correlation of Coefficients:
```

survey

Student Survey Data

Description
This data frame contains the responses of 237 Statistics I students at the University of Adelaide to a number of questions.

Usage

Format
The components of the data frame are:

- **Sex** The sex of the student. (Factor with levels "Male" and "Female".)
- **Wr.Hnd** The span (distance from tip of thumb to tip of little finger of spread hand) of writing hand, in centimetres.
- **NW.Hnd** Span of non-writing hand
- **W.Hnd** Writing hand of student. (Factor, with levels "Left" and "Right".)
- **Fold** “Fold your arms! Which is on top” (Factor, with levels "R on L", "L on R", "Neither").
- **Pulse** Pulse rate of student (beats per minute.)
- **Clap** ‘Clap your hands! Which hand is on top?’ (Factor, with levels "Right", "Left", "Neither").
- **Exer** How often the student exercises. (Factor, with levels "Freq" (frequently), "Some", "None")
- **Smoke** How much the student smokes. (Factor, levels "Heavy", "Regul" (regularly), "Occas" (occasionally), "Never").
- **Height** The height of the student, in centimetres.
- **M.I** Indicates whether the student expressed height in imperial (feet/inches) or metric (centimetres/metres) units. (Factor, levels "Metric", "Imperial").
- **Age** Age of the student, in years.

References

Description

The `synth.tr` data frame has 250 rows and 3 columns. The `synth.te` data frame has 100 rows and 3 columns. It is intended that `synth.tr` be used from training and `synth.te` for testing.

Usage

```
synth.tr
synth.te
```

Format

These data frames contains the following columns:

- `xs` x-coordinate
- `ys` y-coordinate
- `yc` class, coded as 0 or 1.

Source


---

`theta.md` Estimate theta of the Negative Binomial

Description

Given the estimated mean vector, estimate `theta` of the Negative Binomial Distribution.

Usage

```
theta.md(y, mu, dfr, weights, limit = 20, eps = .Machine$double.eps^0.25)
theta.ml(y, mu, n, weights, limit = 10, eps = .Machine$double.eps^0.25, trace = FALSE)
theta.mm(y, mu, dfr, weights, limit = 10, eps = .Machine$double.eps^0.25)
```
Arguments

- **y**: Vector of observed values from the Negative Binomial.
- **mu**: Estimated mean vector.
- **n**: Number of data points (defaults to the sum of weights).
- **dfr**: Residual degrees of freedom (assuming theta known). For a weighted fit this is the sum of the weights minus the number of fitted parameters.
- **weights**: Case weights. If missing, taken as 1.
- **limit**: Limit on the number of iterations.
- **eps**: Tolerance to determine convergence.
- **trace**: logical: should iteration progress be printed?

Details

theta.md estimates by equating the deviance to the residual degrees of freedom, an analogue of a moment estimator.

theta.ml uses maximum likelihood.

theta.mm calculates the moment estimator of theta by equating the Pearson chi-square \( \sum ((y - \mu)^2 / (\mu + \mu^2 / \theta)) \) to the residual degrees of freedom.

Value

The required estimate of theta, as a scalar. For theta.ml, the standard error is given as attribute "SE".

See Also

glm.nb

Examples

```r
quine.nb <- glm.nb(Days ~ .^2, data = quine)
theta.md(quine$Days, fitted(quine.nb), dfr = df.residual(quine.nb))
theta.ml(quine$Days, fitted(quine.nb))
theta.mm(quine$Days, fitted(quine.nb), dfr = df.residual(quine.nb))

# weighted example
yeast <- data.frame(cbind(numbers = 0:5, fr = c(213, 128, 37, 18, 3, 1)))
fit <- glm.nb(numbers ~ 1, weights = fr, data = yeast)
summary(fit)
attach(yeast)
mu <- fitted(fit)
theta.md(numbers, mu, dfr = 399, weights = fr)
theta.ml(numbers, mu, weights = fr)
theta.mm(numbers, mu, dfr = 399, weights = fr)
detach()
```
**topo**  
*Spatial Topographic Data*

**Description**

The **topo** data frame has 52 rows and 3 columns, of topographic heights within a 310 feet square.

**Usage**

```r
topo
```

**Format**

This data frame contains the following columns:

- `x` x coordinates (units of 50 feet)
- `y` y coordinates (units of 50 feet)
- `z` heights (feet)

**Source**

Davis, J.C. (1973) *Statistics and Data Analysis in Geology*. Wiley.

**References**


---

**truehist**  
*Plot a Histogram*

**Description**

Creates a histogram on the current graphics device.

**Usage**

```r
truehist(data, nbins = "Scott", h, x0 = -h/1000, 
breaks, prob = TRUE, xlim = range(breaks), 
ymax = max(est), col, 
xlab = deparse(substitute(data)), bty = "n", ...)
```
Arguments

data  numeric vector of data for histogram. Missing values (NAs) are allowed and omitted.
nbins The suggested number of bins. Either a positive integer, or a character string naming a rule: "Scott" or "Freedman-Diaconis" or "FD". (Case is ignored.)
h  The bin width, a strictly positive number (takes precedence over nbins).
x0  Shift for the bins - the breaks are at x0 + h * (... -1, 0, 1, ...)
breaks  The set of breakpoints to be used. (Usually omitted, takes precedence over h and nbins).
prob  If true (the default) plot a true histogram. The vertical axis has a relative frequency density scale, so the product of the dimensions of any panel gives the relative frequency. Hence the total area under the histogram is 1 and it is directly comparable with most other estimates of the probability density function. If false plot the counts in the bins.
xlim  The limits for the x-axis.
ymax  The upper limit for the y-axis.
col  The colour number for the bar fill.
xlab  label for the plot x-axis. By default, this will be the name of data.
bty  The box type for the plot - defaults to none.
...  additional arguments to rect or plot.

Details

This plots a true histogram, a density estimate of total area 1. If breaks is specified, those breakpoints are used. Otherwise if h is specified, a regular grid of bins is used with width h. If neither breaks nor h is specified, nbins is used to select a suitable h.

Side Effects

A histogram is plotted on the current device.

References


See Also

hist
Description

Uses unbiased cross-validation to select the bandwidth of a Gaussian kernel density estimator.

Usage

ucv(x, nb = 1000, lower, upper)

Arguments

x a numeric vector
nb number of bins to use.
lower, upper Range over which to minimize. The default is almost always satisfactory.

Value

a bandwidth.

References


See Also

bcv, width.SJ, density

Examples

ucv(geyser$duration)

waders

Counts of Waders at 15 Sites in South Africa

Description

The waders data frame has 15 rows and 19 columns. The entries are counts of waders in summer.

Usage

waders
Format

This data frame contains the following columns (species)

S1  Oystercatcher
S2  White-fronted Plover
S3  Kitt Lutz’s Plover
S4  Three-banded Plover
S5  Grey Plover
S6  Ringed Plover
S7  Bar-tailed Godwit
S8  Whimbrel
S9  Marsh Sandpiper
S10 Greenshank
S11 Common Sandpiper
S12 Turnstone
S13 Knot
S14 Sanderling
S15 Little Stint
S16 Curlew Sandpiper
S17 Ruff
S18 Avocet
S19 Black-winged Stilt

The rows are the sites: A = Namibia North coast B = Namibia North wetland C = Namibia South coast D = Namibia South wetland E = Cape North coast F = Cape North wetland G = Cape West coast H = Cape West wetland I = Cape South coast J= Cape South wetland K = Cape East coast L = Cape East wetland M = Transkei coast N = Natal coast O = Natal wetland

Source

J.C. Gower and D.J. Hand (1996) *Biplots* Chapman & Hall Table 9.1. Quoted as from:

Examples

```r
plot(corresp(waders, nf=2))
```
House Insulation: Whiteside’s Data

Description
Mr Derek Whiteside of the UK Building Research Station recorded the weekly gas consumption and average external temperature at his own house in south-east England for two heating seasons, one of 26 weeks before, and one of 30 weeks after cavity-wall insulation was installed. The object of the exercise was to assess the effect of the insulation on gas consumption.

Usage
whiteside

Format
The whiteside data frame has 56 rows and 3 columns:

- **Insul** A factor, before or after insulation.
- **Temp** Purportedly the average outside temperature in degrees Celsius. (These values is far too low for any 56-week period in the 1960s in South-East England. It might be the weekly average of daily minima.)
- **Gas** The weekly gas consumption in 1000s of cubic feet.

Source
A data set collected in the 1960s by Mr Derek Whiteside of the UK Building Research Station. Reported by


References

Examples
```
require(lattice)
xyplot(Gas ~ Temp | Insul, whiteside, panel =
  function(x, y, ...) {
    panel.xyplot(x, y, ...)
    panel.lmline(x, y, ...)
  }, xlab = "Average external temperature (deg. C)",
ylab = "Gas consumption (1000 cubic feet)", aspect = "xy",
strip = function(...) strip.default(..., style = 1))
gasB <- lm(Gas ~ Temp, whiteside, subset = Insul=="Before")
gasA <- update(gasB, subset = Insul=="After")
```
summary(gasB)
summary(gasA)
gasBA <- lm(Gas ~ Insul/Temp - 1, whiteside)
summary(gasBA)

gasQ <- lm(Gas ~ Insul/(Temp + I(Temp^2)) - 1, whiteside)
coef(summary(gasQ))

gasPR <- lm(Gas ~ Insul + Temp, whiteside)
anova(gasPR, gasBA)
options(contrasts = c("contr.treatment", "contr.poly"))
gasBA1 <- lm(Gas ~ Insul*Temp, whiteside)
coef(summary(gasBA1))

width.SJ Bandwidth Selection by Pilot Estimation of Derivatives

Description

Uses the method of Sheather & Jones (1991) to select the bandwidth of a Gaussian kernel density estimator.

Usage

width.SJ(x, nb = 1000, lower, upper, method = c("ste", "dpi"))

Arguments

x a numeric vector
nb number of bins to use.
upper, lower range over which to search for solution if method = "ste".
method Either "ste" ("solve-the-equation") or "dpi" ("direct plug-in").

Value

a bandwidth.

References


See Also

ucv, bcv, density
Examples

```r
attach(geyser)
width.SJ(duration, method = "dpi")
width.SJ(duration)
detach()

width.SJ(galaxies, method = "dpi")
width.SJ(galaxies)
```

---

**write.matrix**

*Write a Matrix or Data Frame*

**Description**

Writes a matrix or data frame to a file or the console, using column labels and a layout respecting columns.

**Usage**

```r
write.matrix(x, file = "", sep = " ", blocksize)
```

**Arguments**

- `x`: matrix or data frame.
- `file`: name of output file. The default ("") is the console.
- `sep`: The separator between columns.
- `blocksize`: If supplied and positive, the output is written in blocks of `blocksize` rows. Choose as large as possible consistent with the amount of memory available.

**Details**

If `x` is a matrix, supplying `blocksize` is more memory-efficient and enables larger matrices to be written, but each block of rows might be formatted slightly differently.

If `x` is a data frame, the conversion to a matrix may negate the memory saving.

**Side Effects**

A formatted file is produced, with column headings (if `x` has them) and columns of data.

**References**


**See Also**

`write.table`
Weight Loss Data from an Obese Patient

Description

The data frame gives the weight, in kilograms, of an obese patient at 52 time points over an 8 month period of a weight rehabilitation programme.

Usage

wtloss

Format

This data frame contains the following columns:

Days  Time in days since the start of the programme.
Weight  Weight in kilograms of the patient.

Source

Dr T. Davies, Adelaide.

References


Examples

wtloss.fm <- nls(Weight ~ b0 + b1*2^(-Days/th),
                  data = wtloss, start = list(b0=90, b1=95, th=120),
                  trace = TRUE)
Package ‘class’

Description Various functions for classification.

Title Functions for Classification

---

SOM Self-Organizing Maps: Online Algorithm

Description

Kohonen’s Self-Organizing Maps are a crude form of multidimensional scaling.

Usage

SOM(data, grid = somgrid(), rlen = 10000, alpha, radii, init)

Arguments

data a matrix or data frame of observations, scaled so that Euclidean distance is appropriate.

grid A grid for the representatives: see somgrid.

rlen the number of updates: used only in the defaults for alpha and radii.

alpha the amount of change: one update is done for each element of alpha. Default is to decline linearly from 0.05 to 0 over rlen updates.

radii the radii of the neighbourhood to be used for each update: must be the same length as alpha. Default is to decline linearly from 4 to 1 over rlen updates.

init the initial representatives. If missing, chosen (without replacement) randomly from data.

Details

alpha and radii can also be lists, in which case each component is used in turn, allowing two- or more phase training.
Value

an object of class "SOM" with components

grid the grid, an object of class "somgrid".
codes a matrix of representatives.

References


See Also

somgrid, batchSOM

Examples

data(crabs, package = "MASS")

lcrabs <- log(crabs[, 4:8])
crabs.grp <- factor(c("B", "b", "O", "o") [rep(1:4, rep(50,4))])
gr <- somgrid(topo = "hexagonal")
crabs.som <- SOM(lcrabs, gr)
plot(crabs.som)

## 2-phase training
crabs.som2 <- SOM(lcrabs, gr,
alpha = list(seq(0.05, 0, len = 1e4), seq(0.02, 0, len = 1e5)),
radii = list(seq(8, 1, len = 1e4), seq(4, 1, len = 1e5)))
plot(crabs.som2)

batchSOM

Self-Organizing Maps: Batch Algorithm

Description

Kohonen’s Self-Organizing Maps are a crude form of multidimensional scaling.

Usage

batchSOM(data, grid = somgrid(), radii, init)
**Arguments**

- **data**: a matrix or data frame of observations, scaled so that Euclidean distance is appropriate.
- **grid**: A grid for the representatives: see `somgrid`.
- **radii**: the radii of the neighbourhood to be used for each pass: one pass is run for each element of `radii`.
- **init**: the initial representatives. If missing, chosen (without replacement) randomly from `data`.

**Details**

The batch SOM algorithm of Kohonen(1995, section 3.14) is used.

**Value**

an object of class "SOM" with components

- **grid**: the grid, an object of class "somgrid".
- **codes**: a matrix of representatives.

**References**


**See Also**

`somgrid`, `SOM`

**Examples**

```r
data(crabs, package = "MASS")

lcrabs <- log(crabs[, 4:8])
crabs.grp <- factor(c("B", "b", "O", "o") [rep(1:4, rep(50,4))])
gr <- somgrid(topo = "hexagonal")
crabs.som <- batchSOM(lcrabs, gr, c(4, 4, 2, 2, 1, 1, 0, 0))
plot(crabs.som)

bins <- as.numeric(knn1(crabs.som$code, lcrabs, 0:47))
plot(crabs.som$grid, type = "n")
symbols(crabs.som$grid$pts[, 1], crabs.som$grid$pts[, 2],
circles = rep(0.4, 48), inches = FALSE, add = TRUE)
text(crabs.som$grid$pts[bins, ] + rnorm(400, 0, 0.1),
as.character(crabs.grp))
```
condense

Condense training set for k-NN classifier

Description

Condense training set for k-NN classifier

Usage

condense(train, class, store, trace = TRUE)

Arguments

- **train**: matrix for training set
- **class**: vector of classifications for test set
- **store**: initial store set. Default one randomly chosen element of the set.
- **trace**: logical. Trace iterations?

Details

The store set is used to 1-NN classify the rest, and misclassified patterns are added to the store set. The whole set is checked until no additions occur.

Value

index vector of cases to be retained (the final store set).

References


See Also

- `reduce.nn`, `multiedit`

Examples

data(iris3)
train <- rbind(iris3[1:25,,1], iris3[1:25,,2], iris3[1:25,,3])
test <- rbind(iris3[26:50,,1], iris3[26:50,,2], iris3[26:50,,3])
c1 <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
keep <- condense(train, c1)
knn(train[keep, , drop=FALSE], test, c1[keep])
keep2 <- reduce.nn(train, keep, c1)
knn(train[keep2, , drop=FALSE], test, c1[keep2])
knn

k-nearest neighbour classification for test set from training set. For each row of the test set, the k nearest (in Euclidean distance) training set vectors are found, and the classification is decided by majority vote, with ties broken at random. If there are ties for the kth nearest vector, all candidates are included in the vote.

Usage

knn(train, test, cl, k = 1, l = 0, prob = FALSE, use.all = TRUE)

Arguments

train matrix or data frame of training set cases.
test matrix or data frame of test set cases. A vector will be interpreted as a row vector for a single case.
cl factor of true classifications of training set
k number of neighbours considered.
l minimum vote for definite decision, otherwise doubt. (More precisely, less than k−l dissenting votes are allowed, even if k is increased by ties.)
prob If this is true, the proportion of the votes for the winning class are returned as attribute prob.
use.all controls handling of ties. If true, all distances equal to the kth largest are included. If false, a random selection of distances equal to the kth is chosen to use exactly k neighbours.

Value

factor of classifications of test set. doubt will be returned as NA.

References


See Also

knn1, knn.cv
Examples

```r
data(iris3)
train <- rbind(iris3[1:25,,1], iris3[1:25,,2], iris3[1:25,,3])
test <- rbind(iris3[26:50,,1], iris3[26:50,,2], iris3[26:50,,3])
cl <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
knn(train, test, cl, k = 3, prob=TRUE)
attributes(.Last.value)
```
**knn1**  

1-nearest neighbour classification

**Description**

Nearest neighbour classification for test set from training set. For each row of the test set, the nearest (by Euclidean distance) training set vector is found, and its classification used. If there is more than one nearest, a majority vote is used with ties broken at random.

**Usage**

```r
knn1(train, test, cl)
```

**Arguments**

- `train`: matrix or data frame of training set cases.
- `test`: matrix or data frame of test set cases. A vector will be interpreted as a row vector for a single case.
- `cl`: factor of true classification of training set.

**Value**

factor of classifications of test set.

**References**


**See Also**

`knn`
Examples

data(iris3)
train <- rbind(iris3[1:25,,1], iris3[1:25,,2], iris3[1:25,,3])
test <- rbind(iris3[26:50,,1], iris3[26:50,,2], iris3[26:50,,3])
c1 <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
knn1(train, test, cl)

---

**lvq1**  
Learning Vector Quantization 1

Description

Moves examples in a codebook to better represent the training set.

Usage

```r
lvq1(x, cl, codebk, niter = 100 * nrow(codebk$x), alpha = 0.03)
```

Arguments

- `x`  
a matrix or data frame of examples
- `cl`  
a vector or factor of classifications for the examples
- `codebk`  
a codebook
- `niter`  
number of iterations
- `alpha`  
constant for training

Details

Selects `niter` examples at random with replacement, and adjusts the nearest example in the codebook for each.

Value

A codebook, represented as a list with components `x` and `cl` giving the examples and classes.

References


See Also

`lvqinit`, `olqv1`, `lvq2`, `lvq3`, `lvqtest`
Examples

data(iris3)
train <- rbind(iris3[1:25,,1], iris3[1:25,,2], iris3[1:25,,3])
test <- rbind(iris3[26:50,,1], iris3[26:50,,2], iris3[26:50,,3])
cl <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
cd <- lvqinit(train, cl, 10)
lvqtest(cd, train)
cd0 <- olvq1(train, cl, cd)
lvqtest(cd0, train)
cd1 <- lvq1(train, cl, cd0)
lvqtest(cd1, train)

lvq2  Learning Vector Quantization 2.1

Description

Moves examples in a codebook to better represent the training set.

Usage

```r
lvq2(x, cl, codebk, niter = 100 * nrow(codebk$x), alpha = 0.03, win = 0.3)
```

Arguments

- `x`: a matrix or data frame of examples
- `cl`: a vector or factor of classifications for the examples
- `codebk`: a codebook
- `niter`: number of iterations
- `alpha`: constant for training
- `win`: a tolerance for the closeness of the two nearest vectors.

Details

Selects `niter` examples at random with replacement, and adjusts the nearest two examples in the codebook if one is correct and the other incorrect.

Value

A codebook, represented as a list with components `x` and `cl` giving the examples and classes.

References

See Also

lvqinit, lvq1, olvq1, lvq3, lvqtest

Examples

data(iris3)
train <- rbind(iris3[1:25,,1], iris3[1:25,,2], iris3[1:25,,3])
test <- rbind(iris3[26:50,,1], iris3[26:50,,2], iris3[26:50,,3])
c1 <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
cd <- lvqinit(train, c1, 10)
lvqtest(cd, train)


Reference

Learning Vector Quantization 3

Description

Moves examples in a codebook to better represent the training set.

Usage

lvq3(x, cl, codebk, niter = 100*nrow(codebk$x), alpha = 0.03,
     win = 0.3, epsilon = 0.1)

Arguments

x   a matrix or data frame of examples
cl  a vector or factor of classifications for the examples
   codebk a codebook
niter number of iterations
alpha constant for training
win   a tolerance for the closeness of the two nearest vectors.
epsilon proportion of move for correct vectors

Details

Selects niter examples at random with replacement, and adjusts the nearest two examples in the codebook for each.

Value

A codebook, represented as a list with components x and cl giving the examples and classes.
lvqinit  171

References


See Also

lvqinit, lvq1, olvq1, lvq2, lvqtest

Examples

data(iris3)
train <- rbind(iris3[1:25,,1], iris3[1:25,,2], iris3[1:25,,3])
test <- rbind(iris3[26:50,,1], iris3[26:50,,2], iris3[26:50,,3])
c1 <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
cd <- lvqinit(train, c1, 10)
lvqtest(cd, train)
cd0 <- olvq1(train, c1, cd)
lvqtest(cd0, train)
cd3 <- lvq3(train, c1, cd0)
lvqtest(cd3, train)

lvqinit

Initialize a LVQ Codebook

Description

Construct an initial codebook for LVQ methods.

Usage

lvqinit(x, cl, size, prior, k = 5)

Arguments

x a matrix or data frame of training examples, n by p.
cl the classifications for the training examples. A vector or factor of length n.
size the size of the codebook. Defaults to \( \min(\text{round}(0.4*\text{ng}*(\text{ng}-1 + p/2),0), n) \) where ng is the number of classes.
prior Probabilities to represent classes in the codebook. Default proportions in the training set.
k k used for k-NN test of correct classification. Default is 5.
Details

Selects size examples from the training set without replacement with proportions proportional to the prior or the original proportions.

Value

A codebook, represented as a list with components x and cl giving the examples and classes.

References


See Also

`lvq1`, `lvq2`, `lvq3`, `olvq1`, `lvqtest`

Examples

data(iris)
train <- rbind(iris[1:25,,1], iris[1:25,,2], iris[1:25,,3])
test <- rbind(iris[26:50,,1], iris[26:50,,2], iris[26:50,,3])
cl <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
cd <- lvqinit(train, cl, 10)
 lvqtest(cd, train)
cd1 <- olvq1(train, cl, cd)
 lvqtest(cd1, train)

`lvqtest`  
*Classify Test Set from LVQ Codebook*

Description

Classify a test set by 1-NN from a specified LVQ codebook.

Usage

`lvqtest(codebk, test)`

Arguments

codebk  codebook object returned by other LVQ software

test   matrix of test examples
Details

uses 1-NN to classify each test example against the codebook.

Value

factor of classification for each row of \( x \)

References


See Also

*lvqinit, olvq1*

Examples

```r
# The function is currently defined as
function(codebk, test) knn1(codebk$x, test, codebk$cl)
```

Description

Multiedit for k-NN classifier

Usage

```r
multiedit(x, class, k = 1, V = 3, I = 5, trace = TRUE)
```

Arguments

- **x**: matrix of training set.
- **class**: vector of classification of training set.
- **k**: number of neighbours used in k-NN.
- **V**: divide training set into \( V \) parts.
- **I**: number of null passes before quitting.
- **trace**: logical for statistics at each pass.

Value

index vector of cases to be retained.
References


See Also

`condense`, `reduce.nn`

Examples

```r
data(iris3)
tr <- sample(1:50, 25)
train <- rbind(iris3[tr,,1], iris3[tr,,2], iris3[tr,,3])
test <- rbind(iris3[-tr,,1], iris3[-tr,,2], iris3[-tr,,3])
cl <- factor(c(rep(1,25),rep(2,25), rep(3,25)), labels=c("s", "c", "v"))
table(cl, knn(train, test, cl, 3))
ind1 <- multiedit(train, cl, 3)
length(ind1)
table(cl, knn(train[ind1, , drop=FALSE], test, cl[ind1], l))
ntrain <- train[ind1,]; ncl <- cl[ind1]
ind2 <- condense(ntrain, ncl)
length(ind2)
table(cl, knn(ntrain[ind2, , drop=FALSE], test, ncl[ind2], l))
```

### olvq1

*Optimized Learning Vector Quantization 1*

#### Description

Moves examples in a codebook to better represent the training set.

#### Usage

```r
olvq1(x, cl, codebk, niter = 40 * nrow(codebk$x), alpha = 0.3)
```

#### Arguments

- `x`  
  a matrix or data frame of examples
- `cl`  
  a vector or factor of classifications for the examples
- `codebk`  
  a codebook
- `niter`  
  number of iterations
- `alpha`  
  constant for training
Details

Selects \texttt{niter} examples at random with replacement, and adjusts the nearest example in the codebook for each.

Value

A codebook, represented as a list with components \texttt{x} and \texttt{cl} giving the examples and classes.

References


See Also

\texttt{lvqinit}, \texttt{lvqtest}, \texttt{lvq1}, \texttt{lvq2}, \texttt{lvq3}

Examples

data(iris3)
train <- rbind(iris3[1:25,,1], iris3[1:25,,2], iris3[1:25,,3])
test <- rbind(iris3[26:50,,1], iris3[26:50,,2], iris3[26:50,,3])
c1 <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
cd <- lvqinit(train, c1, 10)
lvqtest(cd, train)
cd1 <- olvq1(train, c1, cd)
lvqtest(cd1, train)

---

\texttt{reduce.nn} \hspace{1cm} \textit{Reduce Training Set for a k-NN Classifier}

Description

Reduce training set for a k-NN classifier. Used after \texttt{condense}.

Usage

\texttt{reduce.nn(train, ind, class)}

Arguments

- \texttt{train}: matrix for training set
- \texttt{ind}: Initial list of members of the training set (from \texttt{condense}).
- \texttt{class}: vector of classifications for test set
All the members of the training set are tried in random order. Any which when dropped do not cause any members of the training set to be wrongly classified are dropped.

Value

index vector of cases to be retained.

References


See Also

`condense`, `multiedit`

Examples

data(iris3)
train <- rbind(iris3[1:25,,1], iris3[1:25,,2], iris3[1:25,,3])
test <- rbind(iris3[26:50,,1], iris3[26:50,,2], iris3[26:50,,3])
c1 <- factor(c(rep("s",25), rep("c",25), rep("v",25)))
keep <- condense(train, c1)
knn(train[keep,], test, c1[keep])
keep2 <- reduce.nn(train, keep, c1)
knn(train[keep2,], test, c1[keep2])

Description

Plotting functions for SOM results.

Usage

```r
somgrid(xdim = 8, ydim = 6, topo = c("rectangular", "hexagonal"))
```

```r
## S3 method for class 'somgrid':
plot(x, type = "p", ...)
```

```r
## S3 method for class 'SOM':
plot(x, ...)
```
Arguments

xdim, ydim  dimensions of the grid

topo      the topology of the grid.

x       an object inheriting from class "somgrid" or "SOM".

type, ...  graphical parameters.

Details

The class "somgrid" records the coordinates of the grid to be used for (batch or on-line) SOM: this has a plot method.

The plot method for class "SOM" plots a stars plot of the representative at each grid point.

Value

For somgrid, an object of class "somgrid", a list with components

pts  a two-column matrix giving locations for the grid points.

xdim, ydim, topo  as in the arguments to somgrid.

References


See Also

batchSOM, SOM
**Package ‘nnet’**

**Description**  Software for feed-forward neural networks with a single hidden layer, and for multinomial log-linear models.

**Title**  Feed-forward Neural Networks and Multinomial Log-Linear Models

---

### class.ind

**Generates Class Indicator Matrix from a Factor**

**Description**

Generates a class indicator function from a given factor.

**Usage**

```r
class.ind(cl)
```

**Arguments**

- `cl`  factor or vector of classes for cases.

**Value**

a matrix which is zero except for the column corresponding to the class.

**References**


**Examples**

```r
# The function is currently defined as
class.ind <- function(cl)
{
  n <- length(cl)
c1 <- as.factor(cl)
x <- matrix(0, n, length(levels(cl)))
x[(1:n) + n*(unclass(cl)-1)] <- 1
dimnames(x) <- list(names(cl), levels(cl))
}
```
multinom  

Fit Multinomial Log-linear Models

Description

Fits multinomial log-linear models via neural networks.

Usage

multinom(formula, data, weights, subset, na.action, 
          contrasts = NULL, Hess = FALSE, summ = 0, censored = FALSE, 
          model = FALSE, ...)

Arguments

- **formula**: a formula expression as for regression models, of the form response ~ predictors. The response should be a factor or a matrix with K columns, which will be interpreted as counts for each of K classes. A log-linear model is fitted, with coefficients zero for the first class. An offset can be included: it should be a numeric matrix with K columns if the response is either a matrix with K columns or a factor with K > 2 classes, or a numeric vector for a response factor with 2 levels. See the documentation of `formula()` for other details.
- **data**: an optional data frame in which to interpret the variables occurring in formula.
- **weights**: optional case weights in fitting.
- **subset**: expression saying which subset of the rows of the data should be used in the fit. All observations are included by default.
- **na.action**: a function to filter missing data.
- **contrasts**: a list of contrasts to be used for some or all of the factors appearing as variables in the model formula.
- **Hess**: logical for whether the Hessian (the observed/expected information matrix) should be returned.
- **summ**: integer; if non-zero summarize by deleting duplicate rows and adjust weights. Methods 1 and 2 differ in speed (2 uses C); method 3 also combines rows with the same X and different Y, which changes the baseline for the deviance.
- **censored**: If Y is a matrix with K > 2 columns, interpret the entries as one for possible classes, zero for impossible classes, rather than as counts.
- **model**: logical. If true, the model frame is saved as component model of the returned object.
- **...**: additional arguments for nnet
Details

`multinom` calls `nnet`. The variables on the rhs of the formula should be roughly scaled to [0,1] or the fit will be slow or may not converge at all.

Value

A `nnet` object with additional components:

- `deviance`: the residual deviance, compared to the full saturated model (that explains individual observations exactly). Also, minus twice log-likelihood.
- `edf`: the (effective) number of degrees of freedom used by the model
- `AIC`: the AIC for this fit.
- `Hessian`: (if `Hess` is true).
- `model`: (if `model` is true).

References


See Also

`nnet`

Examples

```r
options(contrasts = c("contr.treatment", "contr.poly"))
library(MASS)
example(birthwt)
(bwt.mu <- multinom(low ~ ., bwt))
## Not run: Call:
multinom(formula = low ~ ., data = bwt)

Coefficients:
  (Intercept)    age      lwt  raceblack raceother
     0.823477  -0.03724311 -0.01565475  1.192371  0.7406606
   smoke  ptd      ht      ui   ftv1   ftv2+
     0.7555234  1.343648  1.913213  0.6802007 -0.4363238  0.1789888

Residual Deviance: 195.4755
AIC: 217.4755
## End(Not run)
```
Evaluates Hessian for a Neural Network

Description

Evaluates the Hessian (matrix of second derivatives) of the specified neural network. Normally called via argument `Hess=TRUE` to `nnet` or via `vcov.multinom`.

Usage

```r
nnetHess(net, x, y, weights)
```

Arguments

- `net`: object of class `nnet` as returned by `nnet`.
- `x`: training data.
- `y`: classes for training data.
- `weights`: the (case) weights used in the `nnet` fit.

Value

square symmetric matrix of the Hessian evaluated at the weights stored in the net.

References


See Also

`nnet`, `predict.nnet`

Examples

```r
data(iris3)
# use half the iris data
ir <- rbind(iris3[,1], iris3[,2], iris3[,3])
targets <- matrix(c(rep(c(1,0,0),50), rep(c(0,1,0),50), rep(c(0,0,1),50)),
                  150, 3, byrow=TRUE)
samp <- c(sample(1:50,25), sample(51:100,25), sample(101:150,25))
ir1 <- nnet(ir[samp,], targets[samp,], size=2, rang=0.1, decay=5e-4, maxit=200)
eigen(nnetHess(ir1, ir[samp,], targets[samp,]), TRUE)$values
```
**nnet**

*Fit Neural Networks*

**Description**

Fit single-hidden-layer neural network, possibly with skip-layer connections.

**Usage**

```r
nnet(x, ...)  
## S3 method for class 'formula':
  nnet(formula, data, weights, ...,  
       subset, na.action, contrasts = NULL)
## Default S3 method:
  nnet(x, y, weights, size, Wts, mask,  
       linout = FALSE, entropy = FALSE, softmax = FALSE,  
       censored = FALSE, skip = FALSE, rang = 0.7, decay = 0,  
       maxit = 100, Hess = FALSE, trace = TRUE, MaxNWts = 1000,  
       abstol = 1.0e-4, reltol = 1.0e-8, ...)
```

**Arguments**

- `formula`: A formula of the form `class ~ x1 + x2 + ...`
- `x`: matrix or data frame of `x` values for examples.
- `y`: matrix or data frame of target values for examples.
- `weights`: (case) weights for each example – if missing defaults to 1.
- `size`: number of units in the hidden layer. Can be zero if there are skip-layer units.
- `data`: Data frame from which variables specified in `formula` are preferentially to be taken.
- `subset`: An index vector specifying the cases to be used in the training sample. (NOTE: If given, this argument must be named.)
- `na.action`: A function to specify the action to be taken if NAs are found. The default action is for the procedure to fail. An alternative is `na.omit`, which leads to rejection of cases with missing values on any required variable. (NOTE: If given, this argument must be named.)
- `contrasts`: a list of contrasts to be used for some or all of the factors appearing as variables in the model formula.
- `Wts`: initial parameter vector. If missing chosen at random.
- `mask`: logical parameter vector indicating which parameters should be optimized (default all).
- `linout`: switch for linear output units. Default logistic output units.
- `entropy`: switch for entropy (= maximum conditional likelihood) fitting. Default by least-squares.
softmax
switch for softmax (log-linear model) and maximum conditional likelihood fitting. linout, entropy, softmax and censored are mutually exclusive.
censored
A variant on softmax, in which non-zero targets mean possible classes. Thus for softmax a row of (0, 1, 1) means one example each of classes 2 and 3, but for censored it means one example whose class is only known to be 2 or 3.
skip
switch to add skip-layer connections from input to output.
rang
Initial random weights on [-rang, rang]. Value about 0.5 unless the inputs are large, in which case it should be chosen so that rang \times \max(|x|) is about 1.
decay
parameter for weight decay. Default 0.
maxit
maximum number of iterations. Default 100.
Hess
If true, the Hessian of the measure of fit at the best set of weights found is returned as component Hessian.
trace
switch for tracing optimization. Default TRUE.
MaxNWts
The maximum allowable number of weights. There is no intrinsic limit in the code, but increasing MaxNWts will probably allow fits that are very slow and time-consuming (and perhaps uninterruptable).
abstol
Stop if the fit criterion falls below abstol, indicating an essentially perfect fit.
reltol
Stop if the optimizer is unable to reduce the fit criterion by a factor of at least 1 - reltol.
...
arguments passed to or from other methods.

Details
If the response in formula is a factor, an appropriate classification network is constructed; this has one output and entropy fit if the number of levels is two, and a number of outputs equal to the number of classes and a softmax output stage for more levels. If the response is not a factor, it is passed on unchanged to nnet.default.
Optimization is done via the BFGS method of optim.

Value
object of class "nnet" or "nnet.formula". Mostly internal structure, but has components

- wts
  the best set of weights found
- value
  value of fitting criterion plus weight decay term.
- fitted.values
  the fitted values for the training data.
- residuals
  the residuals for the training data.

References
predict.nnet

Predict New Examples by a Trained Neural Net

Description

Predict new examples by a trained neural net.

Usage

## S3 method for class 'nnet':
predict(object, newdata, type = c("raw","class"), ...)  

Arguments

- **object**: an object of class `nnet` as returned by `nnet`.
- **newdata**: matrix or data frame of test examples. A vector is considered to be a row vector comprising a single case.
- **type**: Type of output
- **...**: arguments passed to or from other methods.

Examples

data(iris3)
# use half the iris data
ir <- rbind(iris3[,1], iris3[,2], iris3[,3])
targets <- class.ind( c(rep("s", 50), rep("c", 50), rep("v", 50)) )
samp <- c(sample(1:50,25), sample(51:100,25), sample(101:150,25))
ir1 <- nnet(ir[samp,], targets[samp,], size = 2, rang = 0.1,
          decay = 5e-4, maxit = 200)
test.cl <- function(true, pred) {
  true <- max.col(true)
  cres <- max.col(pred)
  table(true, cres)
}
test.cl(targets[-samp,], predict(ir1, ir[-samp,]))

# or
ird <- data.frame(rbind(iris3[,1], iris3[,2], iris3[,3]),
  species = factor(c(rep("s",50), rep("c", 50), rep("v", 50))))
ir.nn2 <- nnet(species ~ ., data = ird, subset = samp, size = 2, rang = 0.1,
             decay = 5e-4, maxit = 200)
table(ird$species[-samp], predict(ir.nn2, ird[-samp,], type = "class"))
Details

This function is a method for the generic function \texttt{predict}() for class "nnet". It can be invoked by calling \texttt{predict(x)} for an object \texttt{x} of the appropriate class, or directly by calling \texttt{predict.nnet(x)} regardless of the class of the object.

Value

If \texttt{type = "raw"}, the matrix of values returned by the trained network; if \texttt{type = "class"}, the corresponding class (which is probably only useful if the net was generated by \texttt{nnet.formula}).

References


See Also

\texttt{nnet}, \texttt{which.is.max}

Examples

data(iris3)
# use half the iris data
ir <- rbind(iris3[,1], iris3[,2], iris3[,3])
targets <- class.ind( c(rep("s", 50), rep("c", 50), rep("v", 50)) )
samp <- c(sample(1:50,25), sample(51:100,25), sample(101:150,25))
ir1 <- nnet(ir[samp,,], targets[samp,,],size = 2, rang = 0.1,
          decay = 5e-4, maxit = 200)
test.cl <- function(true, pred){
  true <- max.col(true)
  cres <- max.col(pred)
  table(true, cres)
}
test.cl(targets[-samp,,], predict(ir1, ir[-samp,,]))

# or
ird <- data.frame(rbind(iris3[,1], iris3[,2], iris3[,3]),
                  species=factor(c(rep("s",50), rep("c", 50), rep("v", 50))))
ir.nn2 <- nnet(species ~ ., data = ird, subset = samp, size = 2, rang = 0.1,
               decay = 5e-4, maxit = 200)
table(ird$species[-samp], predict(ir.nn2, ird[-samp,,], type = "class"))

\begin{verbatim}
which.is.max
\end{verbatim}

Find Maximum Position in Vector

Description

Find the maximum position in a vector, breaking ties at random.
which.is.max

Usage

which.is.max(x)

Arguments

x      a vector

Details

Ties are broken at random.

Value

index of a maximal value.

References


See Also

max.col, which.max which takes the first of ties.

Examples

```R
## Not run:
pred <- predict(nnet, test)
table(true, apply(pred,1,which.is.max))
## End(Not run)
```
which is max
Package ‘spatial’

Description  Functions for kriging and point pattern analysis.

Title  Functions for Kriging and Point Pattern Analysis

Kaver

Description

Forms the average of a series of (usually simulated) K-functions.

Usage

Kaver(fs, nsim, ...)

Arguments

fs full scale for K-fn
nsim number of simulations
... arguments to simulate one point process object

Value

list with components x and y of the average K-fn on L-scale.

References


See Also

Kfn, Kenvl
Examples

```r
towns <- ppinit("towns.dat")
par(pty="s")
plot(Kfn(towns, 40), type="b")
plot(Kfn(towns, 10), type="b", xlab="distance", ylab="L(t)")
for(i in 1:10) lines(Kfn(Psim(69), 10))
lims <- Kenvl(10,100,Psim(69))
lines(lims$x,lims$lower, lty=2, col="green")
lines(lims$x,lims$upper, lty=2, col="green")
lines(Kaver(10,25,Strauss(69,0.5,3.5)), col="red")
```

---

**Kenvl**

*Compute Envelope and Average of Simulations of K-fns*

Description

Computes envelope (upper and lower limits) and average of simulations of K-fns

Usage

```r
Kenvl(fs, nsim, ...)
```

Arguments

- `fs` full scale for K-fn
- `nsim` number of simulations
- `...` arguments to produce one simulation

Value

list with components

- `x` distances
- `lower` min of K-fns
- `upper` max of K-fns
- `aver` average of K-fns

References


See Also

Kfn, Kaver
Examples

towns <- ppinit("towns.dat")
par(pty="s")
plot(Kfn(towns, 40), type="b")
plot(Kfn(towns, 10), type="b", xlab="distance", ylab="L(t)")
for(i in 1:10) lines(Kfn(Psim(69), 10))
lims <- Kenvl(10,100,Psim(69))
lines(lims$x,lims$lower, lty=2, col="green")
lines(lims$x,lims$upper, lty=2, col="green")
lines(Kaver(10,25,Strauss(69,0.5,3.5)), col="red")

Kfn

Compute K-fn of a Point Pattern

Description
Actually computes L = sqrt(K/π).

Usage
Kfn(pp, fs, k=100)

Arguments
pp a list such as a pp object, including components x and y
fs full scale of the plot
k number of regularly spaced distances in (0, fs)

Details
relies on the domain D having been set by ppinit or ppregion.

Value
A list with components
x vector of distances
y vector of L-fn values
k number of distances returned – may be less than k if fs is too large
dmin minimum distance between pair of points
lm maximum deviation from L(t) = t

References
Psim

Simulate Binomial Spatial Point Process

Description
Simulate Binomial spatial point process.

Usage
Psim(n)

Arguments
n number of points

Details
relies on the region being set by ppinit or ppregion.

Value
list of vectors of x and y coordinates.

Side Effects
uses the random number generator.

References

See Also
SSI, Strauss

Examples
towns <- ppinit("towns.dat")
par(pty="s")
plot(Kfn(towns, 10), type="s", xlab="distance", ylab="L(t)")
Examples

```r
towns <- ppinit("towns.dat")
par(pty="s")
plot(Kfn(towns, 10), type="s", xlab="distance", ylab="L(t)")
for(i in 1:10) lines(Kfn(Psim(69), 10))
```

SSI

Simulates Sequential Spatial Inhibition Point Process

Description

Simulates SSI (sequential spatial inhibition) point process.

Usage

```r
SSI(n, r)
```

Arguments

- `n`: number of points
- `r`: inhibition distance

Details

uses the region set by `ppinit` or `ppregion`.

Value

list of vectors of x and y coordinates

Side Effects

uses the random number generator.

Warnings

will never return if r is too large and it cannot place n points.

References


See Also

`Psim`, `Strauss`
Examples

```r
towns <- ppinit("towns.dat")
par(pty = "s")
plot(Kfn(towns, 10), type = "b", xlab = "distance", ylab = "L(t)"
lines(Kaver(10, 25, SSI(69, 1.2)))
```

---

**Simulates Strauss Spatial Point Process**

**Description**

Simulates Strauss spatial point process.

**Usage**

```r
Strauss(n, c=0, r)
```

**Arguments**

- `n`: number of points
- `c`: parameter $c$ in $[0, 1]$. $c = 0$ corresponds to complete inhibition at distances up to $r$.
- `r`: inhibition distance

**Details**

Uses spatial birth-and-death process for $4n$ steps, or for $40n$ steps starting from a binomial pattern on the first call from an other function. Uses the region set by `ppinit` or `ppregion`.

**Value**

list of vectors of x and y coordinates

**Side Effects**

uses the random number generator

**References**


**See Also**

`Psim`, `SSI`
Examples

towns <- ppinit("towns.dat")
par(pty="s")
plot(Kfn(towns, 10), type="b", xlab="distance", ylab="L(t)")
lines(Kaver(10, 25, Strauss(69, 0.5, 3.5)))

Description

Compute analysis of variance tables for one or more fitted trend surface model objects; where anova.trls is called with multiple objects, it passes on the arguments to anovalist.trls.

Usage

## S3 method for class 'trls':
anova(object, ...)
anovalist.trls(object, ...)

Arguments

object A fitted trend surface model object from surf.ls
...
Further objects of the same kind

Value

anova.trls and anovalist.trls return objects corresponding to their printed tabular output.

References


See Also

surf.ls

Examples

library(stats)
data(topo, package="MASS")
topo0 <- surf.ls(0, topo)
topo1 <- surf.ls(1, topo)
topo2 <- surf.ls(2, topo)
topo3 <- surf.ls(3, topo)
topo4 <- surf.ls(4, topo)
anova(topo0, topo1, topo2, topo3, topo4)
summary(topo4)
correlogram  

Compute Spatial Correlograms

Description

Compute spatial correlograms of spatial data or residuals.

Usage

correlogram(krig, nint, plotit = TRUE, ...)

Arguments

  krig	trend-surface or kriging object with columns x, y, and z
  nint
    number of bins used
  plotit
    logical for plotting
  ...
    parameters for the plot

Details

Divides range of data into nint bins, and computes the covariance for pairs with separation in each bin, then divides by the variance. Returns results for bins with 6 or more pairs.

Value

  x and y coordinates of the correlogram, and cnt, the number of pairs averaged per bin.

Side Effects

Plots the correlogram if plotit = TRUE.

References


See Also

  variogram

Examples

data(topo, package="MASS")
topo.kr <- surf.ls(2, topo)
correlogram(topo.kr, 25)
d <- seq(0, 7, 0.1)
lines(d, expcov(d, 0.7))
### expcov

**Spatial Covariance Functions**

**Description**

Spatial covariance functions for use with `surf.gls`.

**Usage**

```r
expcov(r, d, alpha = 0, se = 1)
gaucov(r, d, alpha = 0, se = 1)
sphercov(r, d, alpha = 0, se = 1, D = 2)
```

**Arguments**

- `r`: vector of distances at which to evaluate the covariance
- `d`: range parameter
- `alpha`: proportion of nugget effect
- `se`: standard deviation at distance zero
- `D`: dimension of spheres.

**Value**

vector of covariance values.

**References**


**See Also**

`surf.gls`

**Examples**

```r
data(topo, package="MASS")
topo.kr <- surf.ls(2, topo)
correlogram(topo.kr, 25)
d <- seq(0, 7, 0.1)
lines(d, expcov(d, 0.7))
```
ppgetregion  Get Domain for Spatial Point Pattern Analyses

Description

Retrieves the rectangular domain \((x_l, x_u) \times (y_l, y_u)\) from the underlying C code.

Usage

ppgetregion()

Value

A vector of length four with names \(c("x_l", "x_u", "y_l", "y_u")\).

References


See Also

ppregion

ppinit  Read a Point Process Object from a File

Description

Read a file in standard format and create a point process object.

Usage

ppinit(file)

Arguments

file  string giving file name

Details

The file should contain
the number of points
a header (ignored)
x_l x_u y_l y_u scale
x y (repeated n times)
Value

class "pp" object with components x, y, xl, xu, yl, yu

Side Effects

Calls pppregion to set the domain.

References


See Also

ppregion

Examples

towns <- ppinit("towns.dat")
par(pty="s")
plot(Kfn(towns, 10), type="b", xlab="distance", ylab="L(t)")

---

**pplik**  
*Pseudo-likelihood Estimation of a Strauss Spatial Point Process*

Description

Pseudo-likelihood estimation of a Strauss spatial point process.

Usage

pplik(pp, R, ng=50, trace=FALSE)

Arguments

pp  
a pp object

R  
the fixed parameter R

ng  
use a ng x ng grid with border R in the domain for numerical integration.

trace  
logical? Should function evaluations be printed?

Value

estimate for c in the interval [0, 1].

References


ppregion

Set Domain for Spatial Point Pattern Analyses

Description

Sets the rectangular domain \((x_l, x_u) \times (y_l, y_u)\).

Usage

```
ppregion(xl = 0, xu = 1, yl = 0, yu = 1)
```

Arguments

- **xl**: Either \(xl\) or a list containing components \(xl, xu, yl, yu\) (such as a point-process object)
- **xu**
- **yl**
- **yu**

Value

none

Side Effects

initializes variables in the \(C\) subroutines.

References


See Also

```
ppinit, ppgetregion
```
predict.trls  
Predict method for trend surface fits

Description

Predicted values based on trend surface model object

Usage

```r
## S3 method for class 'trls':
predict(object, x, y, ...)
```

Arguments

- `object`: Fitted trend surface model object returned by `surf.ls`
- `x`: Vector of prediction location eastings (x coordinates)
- `y`: Vector of prediction location northings (y coordinates)
- `...`: Further arguments passed to or from other methods.

Value

`predict.trls` produces a vector of predictions corresponding to the prediction locations. To display the output with `image` or `contour`, use `trmat` or convert the returned vector to matrix form.

References


See Also

`surf.ls, trmat`

Examples

```r
data(topo, package="MASS")
topo2 <- surf.ls(2, topo)
topo4 <- surf.ls(4, topo)
x <- c(1.78, 2.21)
y <- c(6.15, 6.15)
z2 <- predict(topo2, x, y)
z4 <- predict(topo4, x, y)
cat("2nd order predictions: ", z2, "\n4th order predictions: ", z4, "\n")
```
Evaluate Kriging Surface over a Grid

Description

Evaluate Kriging surface over a grid.

Usage

```r
prmat(obj, xl, xu, yl, yu, n)
```

Arguments

- `obj` object returned by `surf.gls`
- `xl` limits of the rectangle for grid
- `xu`
- `yl`
- `yu`
- `n` use \( n \times n \) grid within the rectangle

Value

list with components \( x, y \) and \( z \) suitable for `contour` and `image`.

References


See Also

`surf.gls`, `trmat`, `semat`

Examples

```r
data(topo, package="MASS")
(topo.kr <- surf.gls(2, expcov, topo, d=0.7))
(prsurf <- prmat(topo.kr, 0, 6.5, 0, 6.5, 50))
contour(prsurf, levels=seq(700, 925, 25))
```
semat

Evaluate Kriging Standard Error of Prediction over a Grid

Description

Evaluate Kriging standard error of prediction over a grid.

Usage

semat(obj, xl, xu, yl, yu, n, se)

Arguments

obj
object returned by surf.gls

xl
limits of the rectangle for grid

xu

yl

yu

use n x n grid within the rectangle

se
standard error at distance zero as a multiple of the supplied covariance. Otherwise estimated, and it assumed that a correlation function was supplied.

Value

list with components x, y and z suitable for contour and image.

References


See Also

surf.gls, trmat, prmat

Examples

data(topo, package="MASS")
topo.kr <- surf.gls(2, expcov, topo, d=0.7)
prsurf <- prmat(topo.kr, 0, 6.5, 0, 6.5, 50)
contour(prsurf, levels=seq(700, 925, 25))

sesurf <- semat(topo.kr, 0, 6.5, 0, 6.5, 30)

contour(sesurf, levels=c(22,25))
surf.gls  Fits a Trend Surface by Generalized Least-squares

Description
Fits a trend surface by generalized least-squares.

Usage
surf.gls(np, covmod, x, y, z, nx = 1000, ...)

Arguments
- np: degree of polynomial surface
- covmod: function to evaluate covariance or correlation function
- x: x coordinates or a data frame with columns x, y, z
- y: y coordinates
- z: z coordinates. Will supersede x$z
- nx: Number of bins for table of the covariance. Increasing adds accuracy, and increases size of the object.
- ...: parameters for covmod

Value
list with components
- beta: the coefficients
- x
- y
- z: and others for internal use only.

References

See Also
trmat, surf.ls, prmat, semat, expcov, gaucov, sphericov
surf.ls

fits a trend surface by least-squares.

Usage

surf.ls(np, x, y, z)

Arguments

np degree of polynomial surface
x x coordinates or a data frame with columns x, y, z
y y coordinates
z z coordinates. Will supersede x$z

Value

list with components

beta the coefficients
x
y
z and others for internal use only.

References

trls.influence

Regression diagnostics for trend surfaces

Description

This function provides the basic quantities which are used in forming a variety of diagnostics for checking the quality of regression fits for trend surfaces calculated by surf.ls.

Usage

trls.influence(object)

## S3 method for class 'trls':
plot(x, border = "red", col = NA, pch = 4, cex = 0.6,
    add = FALSE, div = 8, ...)

Arguments

object, x
  Fitted trend surface model from surf.ls

div
  scaling factor for influence circle radii in plot.trls

add
  add influence plot to existing graphics if TRUE

border, col, pch, cex, ...
  additional graphical parameters

Value

trls.influence returns a list with components:

r
  raw residuals as given by residuals.trls

hii
  diagonal elements of the Hat matrix

stresid
  standardised residuals

D1
  Cook’s statistic

See Also

trmat.surf.gls

Examples

library(MASS)  # for eqscplot
data(topo, package="MASS")
topo.kr <- surf.ls(2, topo)
trsurf <- trmat(topo.kr, 0, 6.5, 0, 6.5, 50)
eqscplot(trsurf, type = "n")
contour(trsurf, add = TRUE)
points(topo)

eqscplot(trsurf, type = "n")
contour(trsurf, add = TRUE)
plot(topo.kr, add = TRUE)
title(xlab = "Circle radius proportional to Cook's influence statistic")
trmat

Evaluate Trend Surface over a Grid

Description

Evaluate trend surface over a grid.

Usage

trmat(obj, xl, xu, yl, yu, n)

Arguments

obj  
object returned by surf.ls or surf.gls

xl  
limits of the rectangle for grid

xu

yl

yu

n  
use n x n grid within the rectangle

Value

list with components x, y and z suitable for contour and image.
References


See Also

`surf.ls, surf.gls`

Examples

```r
data(topo, package="MASS")
topo.kr <- surf.ls(2, topo)
trsurf <- trmat(topo.kr, 0, 6.5, 0, 6.5, 50)
```

---

### variogram

**Compute Spatial Variogram**

**Description**

Compute spatial (semi-)variogram of spatial data or residuals.

**Usage**

```r
variogram(krig, nint, plotit = TRUE, ...)
```

**Arguments**

- **krig**: trend-surface or kriging object with columns `x`, `y`, and `z`
- **nint**: number of bins used
- **plotit**: logical for plotting
- **...**: parameters for the plot

**Details**

Divides range of data into `nint` bins, and computes the average squared difference for pairs with separation in each bin. Returns results for bins with 6 or more pairs.

**Value**

- `x` and `y` coordinates of the variogram and `cnt`, the number of pairs averaged per bin.

**Side Effects**

Plots the variogram if `plotit = TRUE`
References


See Also

correlogram

Examples

data(topo, package="MASS")
topo.kr <- surf.ls(2, topo)
variogram(topo.kr, 25)
variogram
Index

*Topic algebra
  ginv, 71
  Null, 9

*Topic category
  corresp, 44
  loglm, 91
  mca, 100
  predict.mca, 124

*Topic classif
  batchSOM, 162
  condense, 164
  knn, 165
  knn.cv, 166
  knn1, 167
  lvq1, 168
  lvq2, 169
  lvq3, 170
  lvqinit, 171
  lvqtest, 172
  multiedit, 173
  olvq1, 174
  reduce.nn, 175
  SOM, 161
  somgrid, 176

*Topic datasets
  abbey, 22
  accdeaths, 22
  Aids2, 1
  Animals, 2
  anorexia, 24
  bacteria, 27
  beav1, 30
  beav2, 31
  Belgian-phones, 116
  biopsy, 33
  birthwt, 34
  Boston, 3
  cabbages, 36
  caith, 37
  cars93, 4
  cement, 38
  chem, 39
  coop, 43
  cpus, 49
  crabs, 50
  Cushings, 5
  DDT, 6
  deaths, 51
  drivers, 53
  eagles, 55
  epil, 56
  farms, 59
  fgl, 60
  forbes, 62
  GAGurine, 6
  galaxies, 64
  gehan, 67
  genotype, 69
  geyser, 69
  gilgais, 70
  hills, 75
  housing, 76
  immer, 80
  Insurance, 7
  leuk, 87
  mammals, 99
  mcycle, 101
  Melanoma, 8
  menarche, 101
  michelson, 102
  minn38, 103
  motors, 103
  muscle, 104
  newcomb, 108
  nlschools, 108
  npk, 109
  nppl, 110
oats, 111
OME, 10
painters, 112
petrol, 115
Pima.tr, 13
quine, 128
Rabbit, 14
road, 135
rotifer, 136
Rubber, 14
ships, 138
shoes, 138
shrimp, 139
shuttle, 139
Sitka, 16
Sitka89, 16
Skye, 17
snails, 140
SP500, 15
steam, 142
stormer, 145
survey, 150
synth.tr, 151
topo, 153
Traffic, 18
UScereal, 19
UScrime, 20
VA, 21
waders, 155
whiteside, 157
wtloss, 160
*Topic distribution
  fitdistr, 61
  mvrnorm, 106
  rnegbin, 134
*Topic dplot
  bandwidth.nrd, 28
  bcv, 29
  hist.scott, 76
  kde2d, 82
  ldahist, 86
  truehist, 153
  ucv, 155
  width.SJ, 158
*Topic file
  write.matrix, 159
*Topic hplot
  boxcox, 35
eqscplot, 58
  hist.scott, 76
  ldahist, 86
  logtrans, 95
  pairs.lda, 113
  parcoord, 114
  plot.lda, 116
  plot.mca, 117
  truehist, 153
*Topic htest
  fitdistr, 61
*Topic internal
  loglm, 94
  MASS-internal, 8
*Topic math
  fractions, 63
  rational, 129
*Topic misc
  con2tr, 40
*Topic models
  addterm, 23
  boxcox, 35
  confint-MASS, 41
  contr.sdif, 42
  denumerate, 51
  dose.p, 52
  dropterm, 54
  gamma.dispersion, 65
  gamma.shape, 66
  glm.convert, 72
  glm.nb, 73
  glmMPL, 74
  lm.gls, 88
  lm.ridge, 89
  loglm, 91
  logtrans, 95
  lqs, 96
  multinom, 180
  negative.binomial, 107
  polr, 118
  predict.glmlmPQL, 120
  predict.lqs, 123
  renumerate, 130
  rlm, 131
  stdres, 141
  stepAIC, 142
  studres, 146
  summary.loglm, 146
summary.negbin, 147
theta.md, 151
*Topic multivariate
corresp, 44
cov.rob, 45
cov.trob, 47
isoMDS, 81
lda, 84
mca, 100
mvnrnorm, 106
pairs.lda, 113
plot.lda, 116
plot.mca, 117
predict.lda, 121
predict.mca, 124
predict.qda, 125
qda, 126
sammon, 136
*Topic neural
class.ind, 179
multinom, 180
nnet, 183
nnetHess, 182
predict.nnet, 185
*Topic nonlinear
area, 26
rms.curv, 133
*Topic print
write.matrix, 159
*Topic regression
anova.negbin, 25
boxcox, 35
dose.p, 52
glm.convert, 72
glm.nb, 73
logtrans, 95
negative.binomial, 107
*Topic robust
cov.rob, 45
huber, 78
hubers, 79
lqs, 96
rlm, 131
summary.rlm, 148
*Topic spatial
anova.trls, 195
correlogram, 196
expccov, 197
Kaver, 189
Kenvl, 190
Kfn, 191
ppgetregion, 198
ppinit, 198
pplik, 199
ppregion, 200
predict.trls, 201
prmat, 202
Psim, 192
semat, 203
SSI, 193
Strauss, 194
surf.gls, 204
surf.ls, 205
trls.influence, 206
trmat, 207
variogram, 208
*Topic utilities
class.ind, 179
which.is.max, 186
.rat(rational), 129
[.fractions(fractions), 63
[<-.fractions(fractions), 63
abbey, 22, 44
accdeaths, 22
add.net(nnet), 183
add1.multinom(multinom), 180
addterm, 23, 55, 144
Aids2, 1
Animals, 2
anorexia, 24
anova, 119
anova.glm, 25
anova.multinom(multinom), 180
anova.negbin, 25, 74, 107, 148
anova.trls, 195
anovalist.trls(anova.trls), 195
aov, 143
area, 26
as.character.fractions
(fractions), 63
as.fractions(fractions), 63
bacteria, 27
bandwidth.nrd, 28, 83
batchSOM, 162, 162, 177
bcv, 29, 155, 158
<p>| beav1, 30, 32 | DDT, 6 |
| beav2, 31, 31 | deaths, 51 |
| Belgian-phones, 116 | density, 30, 155, 158 |
| biopsy, 33 | denumerate, 51, 130 |
| birthwt, 34 | deriv3, 134 |
| Boston, 3 | dose.p, 52 |
| boxcox, 35, 96 | drivers, 53 |
| cabbages, 36 | dropl.multinom(multinom), 180 |
| caith, 37 | dropterm, 24, 54, 144 |
| Cars93, 4 | eagles, 55 |
| cats, 38 | eigen, 71 |
| cement, 39 | enlist(MASS-internal), 8 |
| chem, 39, 44 | epil, 56 |
| class.ind, 179 | eqscplot, 58 |
| cmdscale, 82, 137 | eval.nn(nnet), 183 |
| coef, 61, 90 | expcov, 197, 204 |
| coef.fitdistr(fitdistr), 61 | extractAIC, 143 |
| coef.lda(lda), 84 | extractAIC.gls(stepAIC), 142 |
| coef.multinom(multinom), 180 | extractAIC.lme(stepAIC), 142 |
| coef.nnet(nnet), 183 | extractAIC.multinom(multinom), 180 |
| con2tr, 40 | faithful, 70 |
| condense, 164, 174, 176 | family.negbin(glm.nb), 73 |
| confint, 41, 119 | farms, 59 |
| confint–MASS, 41 | fbeta(MASS-internal), 8 |
| confint.glm(confint–MASS), 41 | fgl, 60 |
| confint.nls(confint–MASS), 41 | fitdistr, 61 |
| confint.profile.glm | forbes, 62 |
| (confint–MASS), 41 | formula, 180 |
| confint.profile.nls | fractions, 63, 129 |
| (confint–MASS), 41 | frequency.polygon |
| contr.helmert, 43 | (MASS-internal), 8 |
| contr.sdif, 42 | GAGurine, 6 |
| contr.sum, 43 | galaxies, 64 |
| contr.treatment, 43 | gamma.dispersion, 65, 67 |
| coop, 43 | gamma.shape, 66 |
| correlogram, 196, 209 | gamma.shape.glm, 66 |
| corresp, 44, 100 | gaukov, 204 |
| cov, 48 | gaukov(expcov), 197 |
| cov.mcd(cov.rob), 45 | gehan, 67 |
| cov.mve, 48, 85 | genotype, 69 |
| cov.mve(cov.rob), 45 | geyser, 69 |
| cov.rob, 45 | gilgais, 70 |
| cov.trob, 47 | ginv, 71 |
| cov.wt, 48 | glm, 72–74, 120, 143 |
| cpua, 49 | glm.convert, 72 |
| crabs, 50 | Cushings, 5 |</p>
<table>
<thead>
<tr>
<th>Function</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>glm.nb</td>
<td>25, 26, 72, 73, 107, 148, 152</td>
</tr>
<tr>
<td>glmmPQL</td>
<td>74, 121</td>
</tr>
<tr>
<td>gis</td>
<td>89</td>
</tr>
<tr>
<td>hills</td>
<td>75</td>
</tr>
<tr>
<td>hist</td>
<td>76, 154</td>
</tr>
<tr>
<td>hist.FD(hist.scott)</td>
<td>76</td>
</tr>
<tr>
<td>hist.scott</td>
<td>76</td>
</tr>
<tr>
<td>housing</td>
<td>76</td>
</tr>
<tr>
<td>huber</td>
<td>78, 80</td>
</tr>
<tr>
<td>hubers</td>
<td>79, 79</td>
</tr>
<tr>
<td>immer</td>
<td>80</td>
</tr>
<tr>
<td>influence.measures</td>
<td>207</td>
</tr>
<tr>
<td>Insurance</td>
<td>7</td>
</tr>
<tr>
<td>is.fractions(fractions)</td>
<td>63</td>
</tr>
<tr>
<td>isoMDS</td>
<td>81, 137</td>
</tr>
<tr>
<td>Kaver</td>
<td>189, 190, 192</td>
</tr>
<tr>
<td>kde2d</td>
<td>82</td>
</tr>
<tr>
<td>Kenvl</td>
<td>189, 190, 192</td>
</tr>
<tr>
<td>Kfn</td>
<td>189, 190, 191</td>
</tr>
<tr>
<td>knn</td>
<td>165, 167</td>
</tr>
<tr>
<td>knn.cv</td>
<td>165, 166</td>
</tr>
<tr>
<td>knnl</td>
<td>165, 167</td>
</tr>
<tr>
<td>lda</td>
<td>84, 117, 122, 126, 128</td>
</tr>
<tr>
<td>lda.hist</td>
<td>86, 117</td>
</tr>
<tr>
<td>lddeaths</td>
<td>51</td>
</tr>
<tr>
<td>leuk</td>
<td>87</td>
</tr>
<tr>
<td>link~glm</td>
<td>107</td>
</tr>
<tr>
<td>lm</td>
<td>89, 90, 133, 143</td>
</tr>
<tr>
<td>lm.fit</td>
<td>89, 90</td>
</tr>
<tr>
<td>lm.gls</td>
<td>88</td>
</tr>
<tr>
<td>lm.ridge</td>
<td>89, 89</td>
</tr>
<tr>
<td>lme</td>
<td>74, 75</td>
</tr>
<tr>
<td>lmeObject</td>
<td>74</td>
</tr>
<tr>
<td>lmsreg(lqs)</td>
<td>96</td>
</tr>
<tr>
<td>lmwork(stdres)</td>
<td>141</td>
</tr>
<tr>
<td>logLik</td>
<td>61</td>
</tr>
<tr>
<td>logLik.fitdistr(fitdistr)</td>
<td>61</td>
</tr>
<tr>
<td>logLik.multinom(multinom)</td>
<td>180</td>
</tr>
<tr>
<td>logLik.negbin(glm.nb)</td>
<td>73</td>
</tr>
<tr>
<td>loglin</td>
<td>92, 94, 95</td>
</tr>
<tr>
<td>loglm</td>
<td>51, 52, 91, 94, 95, 130, 147</td>
</tr>
<tr>
<td>loglml</td>
<td>91, 92, 94</td>
</tr>
<tr>
<td>logtrans</td>
<td>95</td>
</tr>
<tr>
<td>lqs</td>
<td>47, 96, 123, 132, 133</td>
</tr>
<tr>
<td>ltsreg(lqs)</td>
<td>96</td>
</tr>
<tr>
<td>lvq1</td>
<td>168, 170–172, 175</td>
</tr>
<tr>
<td>lvq2</td>
<td>168, 169, 171, 172, 175</td>
</tr>
<tr>
<td>lvq3</td>
<td>168, 170, 170, 172, 175</td>
</tr>
<tr>
<td>lvqinit</td>
<td>168, 170, 171, 173, 175</td>
</tr>
<tr>
<td>lvqtest</td>
<td>168, 170, 171, 172, 175</td>
</tr>
<tr>
<td>mad</td>
<td>79</td>
</tr>
<tr>
<td>mammals</td>
<td>99</td>
</tr>
<tr>
<td>MASS-internal</td>
<td>8</td>
</tr>
<tr>
<td>Math.fractions(fractions)</td>
<td>63</td>
</tr>
<tr>
<td>max.col</td>
<td>187</td>
</tr>
<tr>
<td>mca</td>
<td>100, 117, 124</td>
</tr>
<tr>
<td>mcycle</td>
<td>101</td>
</tr>
<tr>
<td>Melanoma</td>
<td>8</td>
</tr>
<tr>
<td>menarche</td>
<td>101</td>
</tr>
<tr>
<td>michelson</td>
<td>102</td>
</tr>
<tr>
<td>minn38</td>
<td>103</td>
</tr>
<tr>
<td>model.frame</td>
<td>119</td>
</tr>
<tr>
<td>model.frame.lda(lda)</td>
<td>84</td>
</tr>
<tr>
<td>model.frame.multinom(multinom)</td>
<td>180</td>
</tr>
<tr>
<td>model.frame.qda(qda)</td>
<td>126</td>
</tr>
<tr>
<td>model.matrix.default</td>
<td>90, 97</td>
</tr>
<tr>
<td>motors</td>
<td>103</td>
</tr>
<tr>
<td>multiedit</td>
<td>164, 173, 176</td>
</tr>
<tr>
<td>multinom</td>
<td>120, 180</td>
</tr>
<tr>
<td>muscle</td>
<td>104</td>
</tr>
<tr>
<td>mvrnorm</td>
<td>106</td>
</tr>
<tr>
<td>na.exclude</td>
<td>97</td>
</tr>
<tr>
<td>na.omit</td>
<td>97</td>
</tr>
<tr>
<td>nclass.freq(MASS-internal)</td>
<td>8</td>
</tr>
<tr>
<td>neg.bin(MASS-internal)</td>
<td>8</td>
</tr>
<tr>
<td>negative.binomial, 26, 72, 74, 107, 148</td>
<td></td>
</tr>
<tr>
<td>negexp.SSival(MASS-internal)</td>
<td>8</td>
</tr>
<tr>
<td>newcomb</td>
<td>108</td>
</tr>
<tr>
<td>nlschools</td>
<td>108</td>
</tr>
<tr>
<td>nnet</td>
<td>181, 182, 183, 186</td>
</tr>
<tr>
<td>nnetHess</td>
<td>182, 185</td>
</tr>
<tr>
<td>norm.net(nnet)</td>
<td>183</td>
</tr>
<tr>
<td>npk</td>
<td>109</td>
</tr>
<tr>
<td>npr1</td>
<td>110</td>
</tr>
<tr>
<td>Null</td>
<td>9</td>
</tr>
<tr>
<td>oats</td>
<td>111</td>
</tr>
<tr>
<td>offset</td>
<td>73, 90</td>
</tr>
<tr>
<td>olvq1</td>
<td>168, 170–173, 174</td>
</tr>
<tr>
<td>OME</td>
<td>10</td>
</tr>
<tr>
<td>Ops.fractions(fractions)</td>
<td>63</td>
</tr>
</tbody>
</table>
optim, 61, 118, 120, 184

painters, 112
pairs, 113
pairs.default, 113
pairs.lda, 113, 117
par, 59
parcoord, 114
petrol, 115
phones (Belgian-phones), 116
Pima.te (Pima.tr), 13
Pima.tr, 13
Pima.tr2 (Pima.tr), 13
plot, 59, 154
plot.lda, 87, 116
plot.lm, 207
plot.mca, 100, 117, 124
plot.ridgelm (lm.ridge), 89
plot.SOM (somgrid), 176
plot.somgrid (somgrid), 176
plot.trls (trls.influence), 206
polr, 118
ppgetregion, 198, 200
ppinit, 192, 198, 200
pplik, 199
ppregion, 198, 199, 200
predict, 119
predict.glmmPQL, 120
predict.lda, 86, 117, 121, 126
predict.lme, 121
predict.lda, 86, 99, 123
predict.mca, 100, 117, 124
predict.multinom (multinom), 180
predict.nnet, 182, 185, 185
predict.qda, 86, 122, 125, 128
predict.rlm (rlm), 131
predict.trls, 201
princomp, 45
print, 61
print.fdist (fdist), 61
print.fractions (fractions), 63
print.gamma.shape (gamma.shape), 66
print.glm.dose (dose.p), 52
print.lda (lda), 84
print.mca (mca), 100
print.multinom (multinom), 180
print.nnet (nnet), 183
print.qda (qda), 126
print.ridgelm (lm.ridge), 89
print.rlm (rlm), 131
print.rms.curv (rms.curv), 133
print.summary.loglm (summary.loglm), 146
print.summary.multinom (multinom), 180
print.summary.negbin (summary.negbin), 147
print.summary.nnet (nnet), 183
print.summary.rlm (summary.rlm), 148
prmat, 202, 203, 204
profile, 41, 119
profile.glm (confint MASS), 41
psi.bisquare (rlm), 131
psi.hampel (rlm), 131
psi.huber (rlm), 131
Psim, 192, 193, 194
qda, 86, 122, 126, 126
qr, 9
qr.Q, 9
quine, 128
Rabbit, 14
rational, 64, 129
rect, 154
reduce.nn, 164, 174, 175
renumerate, 52, 130
residuals, 141, 146
rlm, 131
rms.curv, 133
rnegbin, 134
RNGkind, 46
rnorm, 106
road, 135
rotifer, 136
Rubber, 14
sammon, 82, 136
select (lm.ridge), 89
semat, 202, 203, 204
Shepard (isoMDS), 81
ships, 138
shoes, 138
shrimp, 139
shuttle, 139
Sitka, 16, 17
INDEX

Sitka89, 16, 16
Skye, 17
snails, 140
solve, 71
SOM, 161, 163, 177
somgrid, 161–163, 176
SP500, 15
sphercov, 204
sphercov (expcov), 197
splom, 113
SSI, 192, 193, 194
stars, 177
stdres, 141, 146
steam, 142
step, 144
stepAIC, 24, 55, 119, 142
stormer, 145
Strauss, 192, 193, 194, 200
studres, 141, 146
summary, 119, 147–149
Summary.fractions (fractions), 63
summary.loglm, 146
summary.multinom (multinom), 180
summary.negbin, 26, 74, 107, 147
summary.nnet (nnet), 183
summary.rlm, 148
surf.gls, 197, 202, 203, 204, 206, 208
surf.ls, 195, 201, 204, 205, 207, 208
survey, 150
svd, 45, 71
synth.te (synth.tr), 151
synth.tr, 151
t.fractions (fractions), 63
terms, 51, 52, 130
terms.gls (stepAIC), 142
terms.ime (stepAIC), 142
theta.md, 74, 151
theta.ml (theta.md), 151
theta.mm (theta.md), 151
topo, 153
Traffic, 18
trls.influence, 206
trmat, 201–204, 206, 207
truhist, 153
ucv, 30, 155, 158
update.formula, 143
UScereal, 19
UScrime, 20
VA, 21
vargr, 196, 208
vcov, 119
vcov.multinom (multinom), 180
waders, 155
which.is.max, 186, 186
which.max, 187
whiteside, 157
width.SJ, 30, 155, 158
write.matrix, 159
write.table, 159
wtloss, 160
xtabs, 44, 91, 94