Package ‘SASmixed’

October 12, 2022

Title Data sets from "SAS System for Mixed Models"
Version 1.0-4
Date 2014-03-11
Maintainer Steven Walker <steve.walker@utoronto.ca>
Contact LME4 Authors <lme4-authors@lists.r-forge.r-project.org>
Author Original by Littell, Milliken, Stroup, and Wolfinger,
modifications by Douglas Bates <bates@stat.wisc.edu>, Martin Maechler,
Ben Bolker and Steven Walker
Description Data sets and sample lmer analyses corresponding
to the examples in Littell, Milliken, Stroup and Wolfinger
Depends R (>= 2.14.0),
Suggests lme4, lattice
LazyData yes
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2014-03-11 16:41:14

R topics documented:

Animal ................................................................. 2
AvgDailyGain ......................................................... 3
BIB ................................................................. 4
Bond ................................................................. 5
Cultivation ........................................................... 5
Demand ............................................................. 6
Genetics ............................................................. 7
HR ................................................................. 8
IncBlk ............................................................. 9
Mississippi ......................................................... 10
Animal breeding experiment

Description

The Animal data frame has 20 rows and 3 columns giving the average daily weight gains for animals with different genetic backgrounds.

Format

This data frame contains the following columns:

- **Sire** a factor denoting the sire. (5 levels)
- **Dam** a factor denoting the dam. (2 levels)
- **AvgDailyGain** a numeric vector of average daily weight gains

Details

This appears to be a constructed data set.

Source


Examples

str(Animal)
Description

The `AvgDailyGain` data frame has 32 rows and 6 columns.

Format

This data frame contains the following columns:

- **Id**: the animal number
- **Block**: an ordered factor indicating the barn in which the steer was housed.
- **Treatment**: an ordered factor with levels $0 < 10 < 20 < 30$ indicating the amount of medicated feed additive added to the base ration.
- **adg**: a numeric vector of average daily weight gains over a period of 160 days.
- **InitWt**: a numeric vector giving the initial weight of the animal
- **Trt**: the `Treatment` as a numeric variable

Source


Examples

```r
str(AvgDailyGain)
if (require("lattice", quietly = TRUE, character = TRUE)) {
  ## plot of adg versus Treatment by Block
  xyplot(adg ~ Treatment | Block, AvgDailyGain, type = c("g", "p", "r"),
         xlab = "Treatment (amount of feed additive)",
         ylab = "Average daily weight gain (lb.)", aspect = "xy",
         index.cond = function(x, y) coef(lm(y ~ x))[1])
}
if (require("lme4", quietly = TRUE, character = TRUE)) {
  options(contrasts = c(unordered = "contr.SAS", ordered = "contr.poly"))
  ## compare with output 5.1, p. 178
  print(fm1Adg <- lmer(adg ~ InitWt * Treatment - 1 + (1 | Block),
                       AvgDailyGain))
  print(anova(fm1Adg))  # checking significance of terms
  print(fm2Adg <- lmer(adg ~ InitWt + Treatment + (1 | Block),
                        AvgDailyGain))
  print(anova(fm2Adg))
  print(lmer(adg ~ InitWt + Treatment - 1 + (1 | Block), AvgDailyGain))
}
```
Data from a balanced incomplete block design

Description

The BIB data frame has 24 rows and 5 columns.

Format

This data frame contains the following columns:

- **Block**: an ordered factor with levels 1 < 2 < 3 < 8 < 5 < 6 < 7
- **Treatment**: a treatment factor with levels 1 to 4.
- **y**: a numeric vector representing the response
- **x**: a numeric vector representing the covariate
- **Grp**: a factor with levels 13 and 24

Details

These appear to be constructed data.

Source


Examples

```r
str(BIB)
if (require("lattice", quietly = TRUE, character = TRUE)) {
  xyplot(y ~ x | Block, BIB, groups = Treatment, type = c("g", "p"),
         aspect = "xy", auto.key = list(points = TRUE, space = "right",
         lines = FALSE))
}
if (require("lme4", quietly = TRUE, character = TRUE)) {
  options(contrasts = c(unordered = "contr.SAS", ordered = "contr.poly"))
  ## compare with Output 5.7, p. 188
  print(fm1BIB <- lmer(y ~ Treatment * x + (1 | Block), BIB))
  print(anova(fm1BIB))  # strong evidence of different slopes
  ## compare with Output 5.9, p. 193
  print(fm2BIB <- lmer(y ~ Treatment + x : Grp + (1 | Block), BIB))
  print(anova(fm2BIB))
}
```
**Bond**

### Strengths of metal bonds

#### Description

The Bond data frame has 21 rows and 3 columns of data on the strength required to break metal bonds according to the metal and the ingot.

#### Format

This data frame contains the following columns:

- **pressure** a numeric vector of pressures required to break the bond
- **Metal** a factor with levels c, i and n indicating the metal involved (copper, iron or nickel).
- **Ingot** an ordered factor indicating the ingot of the composition material.

#### Source


#### Examples

```r
str(Bond)
options(contrasts = c(unordered = "contr.SAS", ordered = "contr.poly"))
if (require("lme4", quietly = TRUE, character = TRUE)) {
    # compare with output 1.1 on p. 6
    print(fm1Bond <- lmer(pressure ~ Metal + (1|Ingot), Bond))
    print(anova(fm1Bond))
}
```

---

**Cultivation**

### Bacterial inoculation applied to grass cultivars

#### Description

The Cultivation data frame has 24 rows and 4 columns of data from an experiment on the effect on dry weight yield of three bacterial inoculation treatments applied to two grass cultivars.
Demand

Per-capita demand deposits by state and year

Description

The Demand data frame has 77 rows and 8 columns of data on per-capita demand deposits by state and year.

Format

This data frame contains the following columns:

**State**  an ordered factor with levels WA < FL < CA < TX < IL < DC < NY
**Year**  an ordered factor with levels 1949 < ... < 1959
**d**  a numeric vector of per-capita demand deposits
y a numeric vector of permanent per-capita personal income
rd a numeric vector of service charges on demand deposits
rt a numeric vector of interest rates on time deposits
rs a numeric vector of interest rates on savings and loan association shares.

Source


Examples

```r
str(Demand)
if (require("lme4", quietly = TRUE, character = TRUE)) {
  ## compare to output 3.13, p. 132
  summary(fm1Demand <- lmer(log(d) ~ log(y) + log(rd) + log(rt) + log(rs) + (1|State) + (1|Year), Demand))
}
```

---

### Genetics

Heritability data

---

Description

The Genetics data frame has 60 rows and 4 columns.

Format

This data frame contains the following columns:

- **Location** a factor with levels 1 to 4
- **Block** a factor with levels 1 to 3
- **Family** a factor with levels 1 to 5
- **Yield** a numeric vector of crop yields

Source

Examples

```r
str(Genetics)
if (require("lme4", quietly = TRUE, character = TRUE)) {
  options(contrasts = c(unordered = "contr.SAS", ordered = "contr.poly"))
  summary(fm1Gen <- lmer(Yield ~ Family + (1|Location/Block), Genetics))
}
```

---

Heart rates of patients on different drug treatments

Description

The HR data frame has 120 rows and 5 columns of the heart rates of patients under one of three possible drug treatments.

Format

This data frame contains the following columns:

- **Patient**: an ordered factor indicating the patient.
- **Drug**: the drug treatment - a factor with levels a, b and p where p represents the placebo.
- **baseHR**: the patient's base heart rate.
- **HR**: the observed heart rate at different times in the experiment.
- **Time**: the time of the observation.

Source


Examples

```r
str(HR)
if (require("lattice", quietly = TRUE, character = TRUE)) {
  xyplot(HR ~ Time | Patient, HR, type = c("g", "p", "r"), aspect = "xy",
         index.cond = function(x, y) coef(lm(y ~ x))[1],
         ylab = "Heart rate (beats/min)"
  }
  if (require("lme4", quietly = TRUE, character = TRUE)) {
    options(contrasts = c(unordered = "contr.SAS", ordered = "contr.poly"))
    ## linear trend in time
    print(fm1HR <- lmer(HR ~ Time * Drug + baseHR + (Time|Patient), HR))
    print(anova(fm1HR))
    ## Not run:
    fm2HR <- update(fm1HR, weights = varPower(0.5)) # use power-of-mean variance
    summary(fm2HR)
    intervals(fm2HR) # variance function does not seem significant
    anova(fm1HR, fm2HR) # confirm with likelihood ratio
```
IncBlk

An unbalanced incomplete block experiment

Description

The IncBlk data frame has 24 rows and 4 columns.

Format

This data frame contains the following columns:

- **Block**  an ordered factor giving the block
- **Treatment**  a factor with levels 1 to 4
- **y**  a numeric vector
- **x**  a numeric vector

Details

These data are probably constructed data.

Source


Examples

str(IncBlk)
Mississippi

Nitrogen concentrations in the Mississippi River

Description

The Mississippi data frame has 37 rows and 3 columns.

Format

This data frame contains the following columns:

- **influent**: an ordered factor with levels 3 < 5 < 2 < 1 < 4 < 6
- **y**: a numeric vector
- **Type**: a factor with levels 1 2 3

Source


Examples

```r
str(Mississippi)
if (require("lattice", quietly = TRUE, character = TRUE)) {
  dotplot(drop(influent:Type) ~ y, groups = Type, Mississippi)
}
if (require("lme4", quietly = TRUE, character = TRUE)) {
  options(contrasts = c(unordered = "contr.SAS", ordered = "contr.poly"))
  ## compare with output 4.1, p. 142
  print(fm1Miss <- lmer(y ~ 1 + (1|influent), Mississippi))
  ## compare with output 4.2, p. 143
  print(fm1MLMiss <- update(fm1Miss, REML=FALSE))
  ## BLUP's of random effects on p. 142
  ranef(fm1Miss)
  ## BLUP's of random effects on p. 144
  print(ranef(fm1MLMiss))
  #intervals(fm1Miss)  # interval estimates of variance components
  ## compare to output 4.8 and 4.9, pp. 150-152
  print(fm2Miss <- lmer(y ~ Type+(1|influent), Mississippi, REML=TRUE))
  print(anova(fm2Miss))
}
```
Multilocation

A multilocation trial

Description

The Multilocation data frame has 108 rows and 7 columns.

Format

This data frame contains the following columns:

- **obs** a numeric vector
- **Location** an ordered factor with levels B < D < E < I < G < A < C < F < H
- **Block** a factor with levels 1 to 3
- **Trt** a factor with levels 1 to 4
- **Adj** a numeric vector
- **Fe** a numeric vector
- **Grp** an ordered factor with levels B/1 < B/2 < B/3 < D/1 < D/2 < D/3 < E/1 < E/2 < E/3 < I/1 < I/2 < I/3 < G/1 < G/2 < G/3 < A/1 < A/2 < A/3 < C/1 < C/2 < C/3 < F/1 < F/2 < F/3 < H/1 < H/2 < H/3

Source


Examples

```r
str(Multilocation)
if (require("lme4", quietly = TRUE, character = TRUE)) {
  options(contrasts = c(unordered = "contr.SAS", ordered = "contr.poly"))
  ### Create a Block %in% Location factor
  Multilocation$Grp <- with(Multilocation, Block:Location)
  print(fm1Mult <- lmer(Adj ~ Location * Trt + (1|Grp), Multilocation))
  print(anova(fm1Mult))
  print(fm2Mult <- lmer(Adj ~ Location + Trt + (1|Grp), Multilocation), corr=FALSE)
  print(fm3Mult <- lmer(Adj ~ Location + (1|Grp), Multilocation), corr=FALSE)
  print(fm4Mult <- lmer(Adj ~ Trt + (1|Grp), Multilocation))
  print(anova(fm1Mult))
  print(anova(fm1Mult, fm2Mult, fm3Mult, fm4Mult, fm5Mult))
  ### Treating the location as a random effect
  print(fm1MultR <- lmer(Adj ~ Trt + (1|Location/Trt) + (1|Grp), Multilocation))
  print(anova(fm1MultR))
  fm2MultR <- lmer(Adj ~ Trt + (Trt - 1|Location) + (1|Block), Multilocation)
  ## Warning (not error ?!): Convergence failure in 10000 iter %__FIXME__
```
PBIB

A partially balanced incomplete block experiment

Description

The PBIB data frame has 60 rows and 3 columns.

Format

This data frame contains the following columns:

- **response**: a numeric vector
- **Treatment**: a factor with levels 1 to 15
- **Block**: an ordered factor with levels 1 to 15

Source


Examples

```r
str(PBIB)
if (require("lme4", quietly = TRUE, character = TRUE)) {
  options(contrasts = c(unordered = "contr.SAS", ordered = "contr.poly"))
  ## compare with output 1.7 pp. 24-25
  print(fm1PBIB <- lmer(response ~ Treatment + (1|Block), PBIB))
  print(anova(fm1PBIB))
}
```
Semi2

Oxide layer thicknesses on semiconductors

Description

The Semi2 data frame has 72 rows and 5 columns.

Format

This data frame contains the following columns:

- **Source**: a factor with levels 1 and 2
- **Lot**: a factor with levels 1 to 8
- **Wafer**: a factor with levels 1 to 3
- **Site**: a factor with levels 1 to 3
- **Thickness**: a numeric vector

Source


Examples

```r
str(Semi2)
xtabs(~Lot + Wafer, Semi2)
if (require("lme4", quietly = TRUE, character = TRUE)) {
  options(contrasts = c(unordered = "contr.SAS", ordered = "contr.poly"))
  ## compare with output 4.13, p. 156
  print(fm1Semi2 <- lmer(Thickness ~ 1 + (1|Lot/Wafer), Semi2))
  ## compare with output 4.15, p. 159
  print(fm2Semi2 <- lmer(Thickness ~ Source + (1|Lot/Wafer), Semi2))
  print(anova(fm2Semi2))
  ## compare with output 4.17, p. 163
  print(fm3Semi2 <- lmer(Thickness ~ Source + (1|Lot/Wafer) + (1|Lot:Source),
                           Semi2))
  ## This is not the same as the SAS model.
}
```
Description

The Semiconductor data frame has 48 rows and 5 columns.

Format

This data frame contains the following columns:

- `resistance` a numeric vector
- `ET` a factor with levels 1 to 4 representing etch time.
- `Wafer` a factor with levels 1 to 3
- `position` a factor with levels 1 to 4
- `Grp` an ordered factor with levels 1/1 < 1/2 < 1/3 < 2/1 < 2/2 < 2/3 < 3/1 < 3/2 < 3/3 < 4/1 < 4/2 < 4/3

Source


Examples

```r
str(Semiconductor)
if (require("lme4", quietly = TRUE, character = TRUE)) {
  options(contrasts = c(unordered = "contr.SAS", ordered = "contr.poly"))
  print(fm1Semi <- lmer(resistance ~ ET * position + (1|Grp), Semiconductor))
  print(anova(fm1Semi))
  print((fm2Semi <- lmer(resistance ~ ET + position + (1|Grp), Semiconductor)))
  print(anova(fm2Semi))
}
```

Description

The SIMS data frame has 3691 rows and 3 columns.

Format

This data frame contains the following columns:

- `Pretot` a numeric vector giving the student’s pre-test total score
- `Gain` a numeric vector giving gains from pre-test to the final test
- `Class` an ordered factor giving the student’s class
TeachingI

Source


Examples

```r
str(SIMS)
if (require("lme4", quietly = TRUE, character = TRUE)) {
  options(contrasts = c(unordered = "contr.SAS", ordered = "contr.poly"))
  ## compare to output 7.4, p. 262
  print(fm1SIMS <- lmer(Gain ~ Pretot + (Pretot | Class), data = SIMS))
  print(anova(fm1SIMS))
}
```

---

**TeachingI**

*Teaching Methods I*

Description

The TeachingI data frame has 96 rows and 7 columns.

Format

This data frame contains the following columns:

- **Method** a factor with levels 1 to 3
- **Teacher** a factor with levels 1 to 4
- **Gender** a factor with levels f and m
- **Student** a factor with levels 1 to 4
- **score** a numeric vector
- **Experience** a numeric vector
- **uTeacher** an ordered factor with levels

Source


Examples

```
str(TeachingI)
```
TeachingII  

*Teaching Methods II*

**Description**

The TeachingII data frame has 96 rows and 6 columns.

**Format**

This data frame contains the following columns:

- **Method** a factor with levels 1 to 3
- **Teacher** a factor with levels 1 to 4
- **Gender** a factor with levels f and m
- **IQ** a numeric vector
- **score** a numeric vector
- **uTeacher** an ordered factor with levels

**Source**


**Examples**

```r
str(TeachingII)
```

---

WaferTypes  

*Data on different types of silicon wafers*

**Description**

The WaferTypes data frame has 144 rows and 8 columns.

**Format**

This data frame contains the following columns:

- **Group** a factor with levels 1 to 4
- **Temperature** an ordered factor with levels 900 < 1000 < 1100
- **Type** a factor with levels A and B
- **Wafer** a numeric vector
- **Site** a numeric vector
- **delta** a numeric vector
- **Thick** a numeric vector
- **uWafer** an ordered factor giving a unique code to each group, temperature, type and wafer combination.
Weights

Source


Examples

```r
str(WaferTypes)
```

Description

The `Weights` data frame has 399 rows and 5 columns.

Format

This data frame contains the following columns:

- **strength**: a numeric vector
- **Subject**: a factor with levels 1 to 21
- **Program**: a factor with levels CONT (continuous repetitions and weights), RI (repetitions increasing) and WI (weights increasing)
- **Subj**: an ordered factor indicating the subject on which the measurement is made
- **Time**: a numeric vector indicating the time of the measurement

Source


Examples

```r
str(Weights)
if (require("lme4", quietly = TRUE, character = TRUE)) {
  options(contrasts = c(unordered = "contr.SAS", ordered = "contr.poly"))
  ## compare with output 3.1, p. 91
  print(fm1Weight <- lmer(strength ~ Program * Time + (1|Subj), Weights))
  print(anova(fm1Weight))
  print(fm2Weight <- lmer(strength ~ Program * Time + (Time|Subj), Weights))
  print(anova(fm1Weight, fm2Weight))
  ## Not run:
  intervals(fm2Weight)
  fm3Weight <- update(fm2Weight, correlation = corAR1())
  anova(fm2Weight, fm3Weight)
  fm4Weight <- update(fm3Weight, strength ~ Program * (Time + I(Time^2)),
                      random = ~Time|Subj)
}
```
The `WWheat` data frame has 60 rows and 3 columns.

This data frame contains the following columns:

- **Variety** an ordered factor with 10 levels
- **Yield** a numeric vector of yields
- **Moisture** a numeric vector of soil moisture contents


Examples

```r
str(WWheat)
```
Index

* datasets
  Animal, 2
  AvgDailyGain, 3
  BIB, 4
  Bond, 5
  Cultivation, 5
  Demand, 6
  Genetics, 7
  HR, 8
  IncBlk, 9
  Mississippi, 10
  Multilocation, 11
  PBIB, 12
  Semi2, 13
  Semiconductor, 14
  SIMS, 14
  TeachingI, 15
  TeachingII, 16
  WaferTypes, 16
  Weights, 17
  WWheat, 18

Animal, 2
AvgDailyGain, 3
BIB, 4
Bond, 5
Cultivation, 5
Demand, 6
factor, 11
Genetics, 7
HR, 8
IncBlk, 9
Mississippi, 10
Multilocation, 11
ordered, 11
PBIB, 12
Semi2, 13
Semiconductor, 14
SIMS, 14
TeachingI, 15
TeachingII, 16
WaferTypes, 16
Weights, 17
WWheat, 18