Package ‘ER’

March 12, 2021

Encoding UTF-8
Type Package
Title Effect + Residual Modelling
Version 1.1.0
Date 2021-03-11
Depends R (>= 3.5.0)
Imports ggplot2, scales, gridExtra, glmnet, pls, plsVarSel
License GPL
LazyData TRUE
RoxygenNote 7.1.1
NeedsCompilation no
Author Kristian Hovde Liland [aut, cre],
    Ellen Færgestad Mosleth [ctb]
Maintainer Kristian Hovde Liland <kristian.liland@nmbu.no>
Repository CRAN
Date/Publication 2021-03-12 05:40:03 UTC

R topics documented:

  confints ................................................................. 2
  Diabetes ................................................................. 3
  elastic ................................................................. 4
  ER ................................................................. 5
  Lactobacillus .......................................................... 8
  MS ................................................................. 9
  pls ................................................................. 10

Index 12
Confidence Intervals of Effect Differences

**Description**

Confidence Intervals of Effect Differences

**Usage**

confints(X1, X2, confidence = 0.95, df.used = 0)

```r
## S3 method for class 'confints'
plot(
  x,
  y,
  xlab = "", 
  ylab = "normalised log2",
  sorted = TRUE,
  labels = FALSE,
  nonZero = FALSE,
  xlim = NULL,
  ylim = NULL,
  text.pt = 12,
  ...
)
```

**Arguments**

- **X1**  
  data.frame containing first effect.
- **X2**  
  data.frame containing second effect.
- **confidence**  
  Level of confidence, default = 0.95.
- **df.used**  
  Optional argument indicating how many degrees of freedom have been consumed during deflation. Default = 0.
- **x**  
  Object of class confint.
- **y**  
  Not used.
- **xlab**  
  X label (character)
- **ylab**  
  Y label (character)
- **sorted**  
  Logical indicating if intervals should be sorted according to their mean values, or a vector of indices/labels to sort by.
- **labels**  
  Logical indicating if sample labels should be used on x axis.
- **nonZero**  
  Logical indicating if intervals are required not to include zero.
- **xlim**  
  Limits of the horizontal scale.
- **ylim**  
  Limits of the vertical scale.
- **text.pt**  
  Size scaling of text in the plot (default = 16).
- **...**  
  Further arguments to qplot.
Value

An object of class confints, which holds the information needed to perform statistics or plot the confidence intervals is returned from confints. The plotting routine returns a ggplot structure for plotting.

See Also

ER, elastic and pls.

Examples

data(MS)
# Compare MS and non-MS patients within cluster 1
conf <- with(MS, confints(proteins[MS == "yes" & cluster == 1,],
                       proteins[MS == "no" & cluster == 1,]))

p1 <- plot(conf)
p2 <- plot(conf, nonZero = TRUE)  # Only intervals without 0.
grid.arrange(p1,p2)

# Shorter plot with labels
confShort <- conf[1:10,]
p1 <- plot(confShort, labels = TRUE)
p2 <- plot(confShort, labels = TRUE, nonZero = TRUE)
grid.arrange(p1,p2)

Diabetes

Diabetes data

Description

A data.frame with a design and transcriptomic data.

Usage

data(Diabetes)

Details

Clinical study on humans was performed as a 2-way factorial design with two factors both on two levels: bariatric surgery on two levels (before and after the bariatric surgery) and type 2 diabetes (T2D) on two levels (with and without T2D). There were 8 patients without T2D and 7 with T2D. It was discovered that the patients with T2D would be separated in two groups: 3 patients in the group called T2D1 and 4 patients in the group called T2D2. The experiment can therefore also analysed as 2 way factorial design where the disease factor is on three levels. All patients were obese before bariatric surgery (BMI >45). Transcriptome in the subcutaneous adipose tissue were obtained before and one year after bariatric surgery.
Author(s)
Ellen Færgestad Mosleth

References

Examples
data(Diabetes)
str(Diabetes)

---
elastic  
Elastic-net modeling of ER objects.

Description
Elastic-net modeling of ER objects.

Usage
elastic(er, ...)

## S3 method for class 'ER'
elastic(
er,
effect,
alpha = 0.5,
newdata = NULL,
validation,
segments = NULL,
measure = measure,
family = family,
...
)

Arguments
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>er</td>
<td>Object of class ER.</td>
</tr>
<tr>
<td>...</td>
<td>Additional arguments for pls::cvsegments.</td>
</tr>
<tr>
<td>effect</td>
<td>The effect to be used as response.</td>
</tr>
<tr>
<td>alpha</td>
<td>The elasticnet mixing parameter.</td>
</tr>
<tr>
<td>newdata</td>
<td>Optional new data matrix for prediction.</td>
</tr>
<tr>
<td>validation</td>
<td>Optional validation parameters.</td>
</tr>
</tbody>
</table>
segments | number of segments or list of segments (optional)
measure | Type of performance summary, default = 'class' (see glmnet)
family | Type of model response, default = 'multinomial'.

See Also

ER, pls and confints.

Examples

```r
## Multiple Sclerosis data
data(MS, package = "ER")
er <- ER(proteins ~ MS * cluster, data = MS)
elasticMod <- elastic(er, 'MS', validation = "CV")
sum(elasticMod$classes == MS$MS)
plot(elasticMod)  # Model fit
plot(elasticMod$glmnet.fit)  # Coefficient trajectories

# Select all proteins with non-zeros coefficients
doefs <- coef(elasticMod, s = "lambda.min", exact = TRUE)
(selected <- rownames(doefs)[[1]]
unique(unlist(lapply(coefs,
function(x) which(as.vector(x) != 0)))))[-1])

## Diabetes data
data(Diabetes, package = "ER")
er Dia <- ER(transcriptome ~ surgery * T2D, data = Diabetes)
elasticMod <- elastic(er Dia, 'T2D', validation = "LOO")
```

ER | **Effect + Residual Modelling**

Description

Effect + Residual Modelling

Usage

ER(formula, data)

## S3 method for class 'ER'
print(x, ...)

## S3 method for class 'ER'
plot(
x, 
y = 1,
what = "raw",
col = NULL,
pch = NULL,
model.line = (what %in% c("raw")),
ylim = NULL,
ylab = "",
xlab = "",
main = NULL,
...)

tableER(object, variable)

Arguments

formula a model formula specifying features and effects.
data a data.frame containing response variables (features) and design factors or other groupings/continuous variables.
x Object of class ER.
... Additional arguments to plot
y Response name or number.
what What part of ER to plot; raw data (default), fits, residuals or a named model effect (can be combined with 'effect', see Examples).
col Color of points, defaults to grouping. Usually set to a factor name.
pch Plot character of points, defaults to 1. Usually set to a factor name.
model.line Include line indicating estimates, default = TRUE. Can be an effect name.
ylim Y axis limits (numeric, but defaults to NULL)
ylab Y label (character)
xlab X label (character)
main Main title, defaults to y with description from what.
object ER object.
variable Numeric for selecting a variable for extraction.

Value

ER returns an object of class ER containing effects, ER values, fitted values, residuals, features, coefficients, dummy design, symbolic design, dimensions, highest level interaction and feature names.

References


See Also

Analyses using ER: elastic and pls. Confidence interval plots confints.

Examples

```r
## Multiple Sclerosis
data(MS, package = "ER")
er <- ER(proteins ~ MS * cluster, data = MS)
print(er)
plot(er) # Raw data, first feature
plot(er,2) # Raw data, numbered feature
plot(er,'Q76L83', col='MS', pch='cluster') # Selected colour and plot character
plot(er,'Q76L83', what='effect MS',
     model.line='effect cluster') # Comparison of factors (points and lines)

# Example compound plot
old.par <- par(c("mfrow", "mar"))
# on.exit(par(old.par))
par(mfrow = c(3,3), mar = c(2,4,4,1))
plot(er,'Q76L83') # Raw data, named feature
plot(er,'Q76L83', what='fits') # Fitted values
plot(er,'Q76L83', what='residuals') # Residuals
plot(er,'Q76L83', what='effect MS') # Effect levels
plot(er,'Q76L83', what='effect cluster') # ----||----
plot(er,'Q76L83', what='effect MS:cluster') # ----||----
plot(er,'Q76L83', what='MS') # ER values
plot(er,'Q76L83', what='cluster') # --------||--------
plot(er,'Q76L83', what='MS:cluster') # --------||--------
par(old.par)

# Complete overview of ER
tab <- tableER(er, 1)

# In general there can be more than two, effects, more than two levels, and continuous effects:
# MS$three <- factor(c(rep(1:3,33),1:2))
# er3 <- ER(proteins ~ MS * cluster + three, data = MS)

## Lactobacillus
data(Lactobacillus, package = "ER")
erLac <- ER(proteome ~ strain * growthrate, data = Lactobacillus)
print(erLac)
plot(erLac) # Raw data, first feature
plot(erLac,2) # Raw data, numbered feature
plot(erLac,'P.LSA0316', col='strain',
     pch='growthrate') # Selected colour and plot character
plot(erLac,'P.LSA0316', what='strain',
     model.line='growthrate') # Selected model.line

## Diabetes
```

Lactobacillus

Lactobacillus data

Description
A data frame with a design and proteomic data, transcriptomic data and phenotypic data.

Usage

data(Lactobacillus)

Details
Experiment on Lactobacillus sakei was performed as a 2-way factorial design with two factors both on two levels: strain (L. sakei strains LS25 and 23K) (factor A) and growth condition (high and low glucose availability) (factor B) both on two levels, and their interaction term (factor AB). There were three biological replicates within each group. Transcriptome, proteome and end product profile (lactate, formate, acetate and ethanol) were observed.

Author(s)
Ellen Færgestad Mosleth

References

Examples

data(Lactobacillus)
str(Lactobacillus)
**Multiple Sclerosis data**

**Description**

A data.frame with a design and proteomic data.

**Usage**

```r
data(MS)
```

**Details**

Data from biobank are analysed a study population of 101 patients, 37 were diagnosed with multiple sclerosis, and 64 without multiple sclerosis. Of the patients without multiple sclerosis, 50 were diagnosed with other neurological disorders and 14 were neurologically healthy patients who had undergone spinal anaesthesia for orthopaedic surgery on the knee or ankle, i.e. neurologically healthy controls. Unless otherwise stated, all the patients without multiple sclerosis were considered as controls for this study. All patients with multiple sclerosis had relapsing remitting multiple sclerosis. The proteome were obtained on cerebrospinal fluid samples from all patients prior medical treatment for multiple sclerosis. It was discovered the patients separated into two clusters, called cluster 1 and cluster 2. This is utilised in the data analysis by considering the data as 2-way factorial design with the two factors: MS and clusters both on two levels.

**Author(s)**

Ellen Færgestad Mosleth

**References**


**Examples**

```r
data(MS)
str(MS)
```
**pls**  
*Partial Least Squares modelling of ER objects.*

**Description**  
Partial Least Squares modelling of ER objects.

**Usage**  
`pls(er, ...)`

## S3 method for class 'ER'

`pls(
er,
effect,
ncomp,
newdata = NULL,
er2,
validation,
jackknife = NULL,
shave = NULL,
df.used = NULL,
...)
```

**Arguments**

- **er**: Object of class ER.
- **...**: Additional arguments for `plsr`.
- **effect**: The effect to be used as response.
- **ncomp**: Number of PLS components.
- **newdata**: Optional new data matrix for prediction.
- **er2**: Second object of class ER for comparison.
- **validation**: Optional validation parameters for `plsr`.
- **jackknife**: Optional argument specifying if jackknifing should be applied.
- **shave**: Optional argument indicating if variable shaving should be used. shave should be a list with two elements: the PLS filter method and the proportion to remove. shave = TRUE uses defaults: list("sMC",0.2).
- **df.used**: Optional argument indicating how many degrees of freedom have been consumed during deflation. Default value from input object.

**See Also**

`ER`, `elastic` and `confints`.


Examples

```r
data(MS, package = "ER")
er <- ER(proteins ~ MS * cluster, data = MS[-1,])

plsMod <- pls(er, 'MS', 6, validation = "CV",
              type = "interleaved", length.seg=25, shave = TRUE)
# Error as a function of remaining variables
plot(plsMod$shave)
# Selected variables for minimum error
with(plsMod$shave, colnames(X)[variables[[min.red+1]]])

plsMod <- pls(er, 'MS', 5, validation = "LOO",
              type = "interleaved", length.seg=25, jackknife = TRUE)
colSums(plsMod$classes == as.numeric(MS$MS[-1]))
# Jackknifed coefficient P-values (sorted)
plot(sort(plsMod$jack[,1,1]), pch = '.', ylab = 'P-value')
abline(h=c(0.01, 0.05), col=2:3)

scoreplot(plsMod)
scoreplot(plsMod, comps=c(1,3))  # Selected components
# Use MS categories for colouring and clusters for plot characters.
scoreplot(plsMod, col = er$symbolicDesign[['MS']],
          pch = 20+as.numeric(er$symbolicDesign[['cluster']]))
loadingplot(plsMod, scatter=TRUE)  # scatter=TRUE for scatter plot
```
Index

cconfints, 2, 5, 7, 10
Diabetes, 3
elastic, 3, 4, 7, 10
ER, 3, 5, 5, 10
glmnet, 5
Lactobacillus, 8
MS, 9
plot.confints(confints), 2
plot.ER(ER), 5
pls, 3, 5, 7, 10
print.ER(ER), 5

tableER(ER), 5