Package `pcsstools`

March 23, 2021

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Tools for Regression Using Pre-Computed Summary Statistics</td>
</tr>
<tr>
<td>Version</td>
<td>0.1.1</td>
</tr>
</tbody>
</table>

**Description**

Defines functions to describe regression models using only pre-computed summary statistics (i.e. means, variances, and covariances) in place of individual participant data. Possible models include linear models for linear combinations, products, and logical combinations of phenotypes. Implements methods presented in Wolf et al. (2021) <doi:10.1101/2021.03.08.433979> Wolf et al. (2020) <doi:10.1142/9789811215636_0063> and Gasdaska et al. (2019) <doi:10.1142/9789813279827_0036>.

**License**

GPL (>= 3)

**Encoding**

UTF-8

**LazyData**

true

**Depends**

R (>= 3.5.0)

**Imports**

gtools, Rdpack, stats

**RdMacros**

Rdpack

**RoxygenNote**

7.1.1

**Suggests**

testthat, knitr, rmarkdown, spelling

**URL**

https://github.com/jackmwolf/pcsstools/

**BugReports**

https://github.com/jackmwolf/pcsstools/issues

**Language**

en-US

**NeedsCompilation**

no

**Author**

Jack Wolf [aut, cre, cph] (<https://orcid.org/0000-0002-8919-8740>), R Core Team and contributors worldwide [cph, aut] (Author and copyright holder of modified 'stats' fragments)

**Maintainer**

Jack Wolf <jackwolf910@gmail.com>

**Repository**

CRAN

**Date/Publication**

2021-03-23 19:30:02 UTC
\section*{R topics documented:}

- `anova.pcsllm` .................................................. 2
- `approx_and` .................................................. 3
- `approx_conditional` ......................................... 4
- `approx_mult_prod` ........................................... 5
- `approx_or` .................................................. 6
- `approx_prod_stats` .......................................... 7
- `approx_response_cov_recursive` ......................... 8
- `calculate_lm` .................................................. 9
- `calculate_lm_combo` ........................................ 10
- `check_terms` .................................................. 12
- `extract_predictors` ......................................... 12
- `extract_response` ........................................... 13
- `get_pcor` .................................................. 13
- `guess_response` ............................................. 14
- `make_permutations` .......................................... 14
- `model_and` .................................................. 15
- `model_combo` ................................................ 16
- `model_or` .................................................. 18
- `model_prcomp` ............................................... 19
- `model_product` ............................................... 21
- `model_singular` ............................................ 23
- `new_predictor` ............................................... 24
- `new_predictor_binary` ..................................... 24
- `new_predictor_normal` ..................................... 25
- `new_predictor_snp` ......................................... 26
- `pcsllm` ..................................................... 27
- `pcsstools` .................................................. 30
- `pcsstools_example` ......................................... 30
- `print.pcsllm` ................................................ 31

\section*{Index} ......................................................... 33

\begin{itemize}
\item \texttt{anova.pcsllm} \hspace{1cm} ANOVA for linear models fit using PCSS
\end{itemize}

\section*{Description}

Compute an analysis of variance table for one or more linear model fitted using PCSS.

\section*{Usage}

\begin{verbatim}
## S3 method for class 'pcsllm'
anova(object, ...)

## S3 method for class 'pcsllmlist'
anova(object, ..., scale = 0, test = "F")
\end{verbatim}
approx_and

Arguments

object, ... objects of class pcsslm.
scale numeric. An estimate of the noise variance $\sigma^2$. If zero this will be estimated from the largest model considered.
test a character string specifying the test statistic to be used. Can be one of "F", "Chisq" or "Cp", with partial matching allowed, or NULL for no test.

Value

An object of class "anova" inheriting from class "data.frame".

Author(s)

R Core Team and contributors worldwide. Modified by Jack Wolf

approx_and

Approximate a linear model for a series of logical AND statements

Description

approx_and approximates the linear model for a conjunction of m phenotypes as a function of a set of predictors.

Usage

approx_and(
  means,
  covs,
  n,
  predictors,
  add_intercept = TRUE,
  verbose = FALSE,
  response_assumption = "binary",
  ...
)

Arguments

means vector of predictor and response means with the last m means being the means of m binary responses to combine in a logical and statement.
covs a matrix of the covariance of all model predictors and the responses with the order of rows/columns corresponding to the order of means.
n sample size.
predictors list of objects of class predictor corresponding to the order of the predictors in means.
approx_conditional

- **add_intercept**: logical. Should the linear model add an intercept term?
- **verbose**: should output be printed to console?
- **response_assumption**: character. Either "binary" or "continuous". If "binary", specific calculations will be done to estimate product means and variances.
- ... additional arguments

**Value**

An object of class "pcsslm".

An object of class "pcsslm" is a list containing at least the following components:

- **call**: the matched call
- **terms**: the terms object used
- **coefficients**: a px4 matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value.
- **sigma**: the square root of the estimated variance of the random error.
- **df**: degrees of freedom, a 3-vector \( p, n - p, p \star \), the first being the number of non-aliased coefficients, the last being the total number of coefficients.
- **fstatistic**: a 3-vector with the value of the F-statistic with its numerator and denominator degrees of freedom.
- **r.squared**: \( R^2 \), the 'fraction of variance explained by the model'.
- **adj.r.squared**: the above \( R^2 \) statistic 'adjusted', penalizing for higher \( p \).
- **cov.unscaled**: a pxp matrix of (unscaled) covariances of the \( \text{coef}[j], j = 1, \ldots, p \).
- **Sum Sq**: a 3-vector with the model's Sum of Squares Regression (SSR), Sum of Squares Error (SSE), and Sum of Squares Total (SST).

**References**


approx_conditional

**Approach the mean of Y conditional on X**

**Description**

Approximate the mean of Y conditional on X

**Usage**

```
approx_conditional(means, covs, response, n)
```
approx_mult_prod

Arguments

- **means**: Vector of the mean of X and the mean of Y
- **covs**: Matrix of covariances for X and Y
- **response**: Character. If "binary" truncates means to interval [0, 1]. If "continuous" does not restrict.
- **n**: Sample size

Value

A list of length 2 consisting of 2 functions that give the estimated conditional mean and conditional variance of Y as a function of X

Usage

```r
approx_mult_prod(
  means,
  covs,
  n,
  response,
  predictors,
  responses,
  verbose = FALSE
)
```

Arguments

- **means**: a vector of predictor and response means with all response means at the end of the vector.
- **covs**: covariance matrix of all predictors and responses with column and row order corresponding to the order of means.
- **n**: sample size (an integer).
- **response**: a string. Currently supports "binary" or "continuous".
- **predictors, responses**: lists of objects of class predictor where each entry corresponds to one predictor/response variable.
- **verbose**: logical.

Description

approx_mult_prod recursively estimates the covariances and means of a set of responses. Estimates are approximated using all unique response orderings and aggregated.
approx_or

Value

A list containing the following elements:

means a vector of the (approximated) means of all predictors and the product of responses

covs a matrix of (approximated) covariances between all predictors and the product of responses

References


approx_or Approximate a linear model for a series of logical OR statements

Description

approx_or approximates the linear model for a disjunction of m phenotypes as a function of a set of predictors.

Usage

approx_or(means, covs, n, predictors, add_intercept = TRUE, verbose = FALSE, response_assumption = "binary", ...)

Arguments

means vector of predictor and response means with the last m means being the means of m binary responses to combine in a logical OR statement.

covs a matrix of the covariance of all model predictors and the responses with the order of rows/columns corresponding to the order of means.

n sample size.

predictors list of objects of class predictor corresponding to the order of the predictors in means.

add_intercept logical. Should the linear model add an intercept term?
approx_prod_stats

verbose should output be printed to console?
response_assumption character. Either "binary" or "continuous". If "binary", specific calculations will be done to estimate product means and variances.

Value

an object of class "pcsslm".
An object of class "pcsslm" is a list containing at least the following components:
call the matched call
terms the terms object used
coefficients a px4 matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value.
sigma the square root of the estimated variance of the random error.
df degrees of freedom, a 3-vector p, n − p, p∗, the first being the number of non-aliased coefficients, the last being the total number of coefficients.
fstatistic a 3-vector with the value of the F-statistic with its numerator and denominator degrees of freedom.
r.squared R², the 'fraction of variance explained by the model'.
adj.r.squared the above R² statistic 'adjusted', penalizing for higher p.
cov.unscaled a pxp matrix of (unscaled) covariances of the coef[j], j = 1,...p.
Sum Sq a 3-vector with the model’s Sum of Squares Regression (SSR), Sum of Squares Error (SSE), and Sum of Squares Total (SST).

References


approx_prod_stats

Approximate summary statistics for a product of phenotypes and a set of predictors

Description

Approximate summary statistics for a product of phenotypes and a set of predictors

Usage

approx_prod_stats(means, covs, n, response, predictors)
approx_response_cov_recursive

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>means</td>
<td>Vector of means of predictors and the two phenotypes to be multiplied</td>
</tr>
<tr>
<td>covs</td>
<td>Covariance matrix of all predictors and the two phenotypes</td>
</tr>
<tr>
<td>n</td>
<td>Sample size</td>
</tr>
<tr>
<td>response</td>
<td>character. Either &quot;binary&quot; or &quot;continuous&quot;.</td>
</tr>
<tr>
<td>predictors</td>
<td>a list of elements of class predictor</td>
</tr>
</tbody>
</table>

**Value**

A list with the predicted covariance matrix of all predictors and the product and the means of all predictors and the product.

---

**approx_response_cov_recursive**

*Approximate the covariance of one response with an arbitrary product of responses.*

---

**Description**

Approximate the covariance of one response with an arbitrary product of responses.

**Usage**

```r
approx_response_cov_recursive(
  ids,
  r_covs,
  r_means,
  n,
  responses,
  response,
  verbose = FALSE
)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ids</td>
<td>Column ids of responses to use. First is taken alone while 2nd to last are to be multiplied</td>
</tr>
<tr>
<td>r_covs</td>
<td>Response covariance matrix</td>
</tr>
<tr>
<td>r_means</td>
<td>Response means (vector)</td>
</tr>
<tr>
<td>n</td>
<td>Sample size</td>
</tr>
<tr>
<td>responses</td>
<td>List of lists with elements of class predictor</td>
</tr>
<tr>
<td>response</td>
<td>Character, Either &quot;binary&quot; or &quot;continuous&quot;</td>
</tr>
<tr>
<td>verbose</td>
<td>logical</td>
</tr>
</tbody>
</table>
**calculate_lm**

*Calculate a linear model using PCSS*

**Value**

A vector with the approximated covariance, and approximated mean and variance of the product

**Description**

calculate_lm describes the linear model of the last listed variable in means and covs as a function of all other variables in means and covs.

**Usage**

```r
calculate_lm(
  means,
  covs,
  n,
  add_intercept = FALSE,
  keep_pcss = FALSE,
  terms = NULL
)
```

**Arguments**

- **means**: a vector of means of all model predictors and the response with the last element the response mean.
- **covs**: a matrix of the covariance of all model predictors and the response with the order of rows/columns corresponding to the order of means.
- **n**: sample size
- **add_intercept**: logical. If TRUE adds an intercept to the model.
- **keep_pcss**: logical. If TRUE, returns means and covs.
- **terms**: terms

**Value**

an object of class "pcsslm".

An object of class "pcsslm" is a list containing at least the following components:

- **call**: the matched call
- **terms**: the terms object used
- **coefficients**: a px4 matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value.
- **sigma**: the square root of the estimated variance of the random error.
calculate_lm_combo

**Calculate a linear model for a linear combination of responses**

**Description**

calculate_lm_combo describes the linear model for a linear combination of responses as a function of a set of predictors.

**Usage**

calculate_lm_combo(means, covs, n, phi, m = length(phi), add_intercept, ...)

**Arguments**

- **means**: a vector of means of all model predictors and the response with the last m elements the response means (with order corresponding to the order of weights in phi).
- **covs**: a matrix of the covariance of all model predictors and the responses with the order of rows/columns corresponding to the order of means.
- **n**: sample size.

**df**

degrees of freedom, a 3-vector \( p, n - p, p \), the first being the number of non-aliased coefficients, the last being the total number of coefficients.

**fstatistic**

a 3-vector with the value of the F-statistic with its numerator and denominator degrees of freedom.

**r.squared**

\( R^2 \), the 'fraction of variance explained by the model'.

**adj.r.squared**

the above \( R^2 \) statistic 'adjusted', penalizing for higher \( p \).

**cov.unscaled**

a \( pxp \) matrix of (unscaled) covariances of the \( coef[j], j = 1, ...p \).

**Sum Sq**

a 3-vector with the model’s Sum of Squares Regression (SSR), Sum of Squares Error (SSE), and Sum of Squares Total (SST).

**References**


**calculate_lm_combo**

phi  vector of linear combination weights with one entry per response variable.

m  number of responses to combine. Defaults to length(weighs).

add_intercept  logical. If TRUE adds an intercept to the model.

...  additional arguments

**Value**

an object of class "pcsslm".

An object of class "pcsslm" is a list containing at least the following components:

call  the matched call

terms  the terms object used

coefficients  a p x 4 matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value.

sigma  the square root of the estimated variance of the random error.

df  degrees of freedom, a 3-vector , the first being the number of non-aliased coefficients, the last being the total number of coefficients.

fstatistic  a 3-vector with the value of the F-statistic with its numerator and denominator degrees of freedom.

r.squared  \( R^2 \), the ‘fraction of variance explained by the model’.

adj.r.squared  the above \( R^2 \) statistic ‘adjusted’, penalizing for higher \( p \).

cov.unscaled  a p x p matrix of (unscaled) covariances of the \( \text{coef}[j], j = 1, \ldots, p \).

Sum Sq  a 3-vector with the model’s Sum of Squares Regression (SSR), Sum of Squares Error (SSE), and Sum of Squares Total (SST).

**References**


check_terms  
Check that independent and dependent variables are accounted for through PCSS

Description
Check that independent and dependent variables are accounted for through PCSS

Usage
check_terms(xterms, yterms, pcssterms, pcsstype)

Arguments
xterms, yterms  character vector of model’s independent variables or variables combined to the dependent variable
pcssterms  character vector of variables with provided PCSS
pcsstype  character describing the PCSS being checked. Either "means", "covs", "predictors", or "responses".

Value
No return value, called for side effects

extract_predictors  Extract independent variables from a formula

Description
Extract independent variables from a formula

Usage
extract_predictors(formula = formula())

Arguments
formula  an object of class formula.

Value
A list with a character vector of all predictors and a logical value indicating whether the model includes an intercept term.
extract_response

**Description**

Extract dependent variables from a formula as a string

**Usage**

`extract_response(formula = formula())`

**Arguments**

- `formula` an object of class `formula`.

**Value**

a character vector of all responses

---

get_pcor

**Description**

Approximate the partial correlation of Y and Z given X

**Usage**

`get_pcor(covs, cors = cov2cor(covs))`

**Arguments**

- `covs` Covariance matrix of X, Y, and Z.
- `cors` Correlation matrix of X, Y, and Z.

**Value**

Approximated partial correlation of the later two terms given the first
**guess_response**  
*Guess the function that is applied to a set of responses*

**Description**  
guess_response takes a character vector of the dependent variable from a formula object and identifies which function separates the individual variables that make up the response. It then returns the model_* function to model the appropriate response using PCSS.

**Usage**  
guess_response(response = character())

**Arguments**  
response character. Output of extract_response.

**Value**  
A character. Either "model_combo", "model_product", "model_or", "model_and", or "model_singular".

---

**make_permutations**  
*List all permutations of a sequence of integers*

**Description**  
Lists all permutations of 1,2,...,m unique up to the first two elements

**Usage**  
make_permutations(m)

**Arguments**  
m number of elements to permute

**Value**  
A list of vectors of permutations of 1,2,...,m.
model_and

Approximate a linear model for a series of logical AND statements using PCSS

Description

model_and approximates the linear model for the conjunction of m phenotypes as a function of a set of predictors.

Usage

model_and(formula, n, means, covs, predictors, ...)

Arguments

formula an object of class formula whose dependent variable is a combination of variables and logical & operators. All model terms must be accounted for in means and covs.
n sample size.
means named vector of predictor and response means.
covs named matrix of the covariance of all model predictors and the responses.
predictors named list of objects of class predictor.
... additional arguments

Value

an object of class "pcsslmlm".
An object of class "pcsslmlm" is a list containing at least the following components:

call the matched call
terms the terms object used
coefficients a p x 4 matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value.
sigma the square root of the estimated variance of the random error.
df degrees of freedom, a 3-vector p, n - p, p*, the first being the number of non-aliased coefficients, the last being the total number of coefficients.
fstatistic a 3-vector with the value of the F-statistic with its numerator and denominator degrees of freedom.
r.squared R^2, the ‘fraction of variance explained by the model’.
adj.r.squared the above R^2 statistic ‘adjusted’, penalizing for higher p.
cov.unscaled a p x p matrix of (unscaled) covariances of the coef[j], j = 1,...p.
Sum Sq a 3-vector with the model’s Sum of Squares Regression (SSR), Sum of Squares Error (SSE), and Sum of Squares Total (SST).
References


Examples

```r
ex_data <- pcsstools_example[c("g1", "x1", "y4", "y5")]
head(ex_data)
means <- colMeans(ex_data)
covs <- cov(ex_data)
n <- nrow(ex_data)
predictors <- list(
  g1 = new_predictor_snp(maf = mean(ex_data$g1) / 2),
  x1 = new_predictor_normal(mean = mean(ex_data$x1), sd = sd(ex_data$x1))
)
model_and(
  y4 & y5 ~ g1 + x1,
  means = means, covs = covs, n = n, predictors = predictors
)
summary(lm(y4 & y5 ~ g1 + x1, data = ex_data))
```

model_combo Model a linear combination of a set of phenotypes using PCSS

Description

model_combo calculates the linear model for a linear combination of phenotypes as a function of a set of predictors.

Usage

```r
model_combo(formula, phi, n, means, covs, ...)
```

Arguments

- `formula` an object of class formula whose dependent variable is a series of variables joined by + operators. `model_combo` will treat a principal component score of those variables as the actual dependent variable. All model terms must be accounted for in means and covs.
- `phi` named vector of linear weights for each variable in the dependent variable in formula.
- `n` sample size.
- `means` named vector of predictor and response means.
- `covs` named matrix of the covariance of all model predictors and the responses.
- `...` additional arguments
Value

an object of class "pcsslm".

An object of class "pcsslm" is a list containing at least the following components:

- **call**: the matched call
- **terms**: the terms object used
- **coefficients**: a $p \times 4$ matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value.
- **sigma**: the square root of the estimated variance of the random error.
- **df**: degrees of freedom, a 3-vector $p, n - p, p^*$, the first being the number of non-aliased coefficients, the last being the total number of coefficients.
- **fstatistic**: a 3-vector with the value of the F-statistic with its numerator and denominator degrees of freedom.
- **r.squared**: $R^2$, the 'fraction of variance explained by the model'.
- **adj.r.squared**: the above $R^2$ statistic 'adjusted', penalizing for higher $p$.
- **cov.unscaled**: a $p \times p$ matrix of (unscaled) covariances of the $\text{coef}[j], j = 1, \ldots, p$.
- **Sum Sq**: a 3-vector with the model’s Sum of Squares Regression (SSR), Sum of Squares Error (SSE), and Sum of Squares Total (SST).

References


Examples

```r
ex_data <- pcsstools_example[c("g1", "x1", "x2", "x3", "y1", "y2", "y3")]
head(ex_data)
means <- colMeans(ex_data)
covs <- cov(ex_data)
n <- nrow(ex_data)
phi <- c("y1" = 1, "y2" = -1, "y3" = 0.5)

model_combo(
  y1 + y2 + y3 ~ g1 + x1 + x2 + x3,
  phi = phi, n = n, means = means, covs = covs
)

summary(lm(y1 - y2 + 0.5 * y3 ~ g1 + x1 + x2 + x3, data = ex_data))
```
model_or

Approximate a linear model for a series of logical OR statements using PCSS

Description

model_or approximates the linear model for the a disjunction of m phenotypes as a function of a set of predictors.

Usage

model_or(formula, n, means, covs, predictors, ...)

Arguments

formula an object of class formula whose dependent variable is a combination of variables and logical | operators. All model terms must be accounted for in means and covs.
n sample size.
means named vector of predictor and response means.
covs named matrix of the covariance of all model predictors and the responses.
predictors named list of objects of class predictor.
... additional arguments

Value

an object of class "pcsslmlm".
An object of class "pcsslmlm" is a list containing at least the following components:
call the matched call
terms the terms object used
coefficients a p x 4 matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value.
sigma the square root of the estimated variance of the random error.
df degrees of freedom, a 3-vector \( p, n - p, p^* \), the first being the number of non-aliased coefficients, the last being the total number of coefficients.
fstatistic a 3-vector with the value of the F-statistic with its numerator and denominator degrees of freedom.
r.squared \( R^2 \), the 'fraction of variance explained by the model'.
adj.r.squared the above \( R^2 \) statistic 'adjusted', penalizing for higher \( p \).
cov.unscaled a \( p \times p \) matrix of (unscaled) covariances of the \( \text{coef}[j], j = 1, ... p \).
Sum Sq a 3-vector with the model’s Sum of Squares Regression (SSR), Sum of Squares Error (SSE), and Sum of Squares Total (SST).
References


Examples

```r
ex_data <- pcsstools_example[c("g1", "x1", "y4", "y5")]
head(ex_data)
means <- colMeans(ex_data)
covs <- cov(ex_data)
n <- nrow(ex_data)
predictors <- list(
g1 = new_predictor_snp(maf = mean(ex_data$g1) / 2),
x1 = new_predictor_normal(mean = mean(ex_data$x1), sd = sd(ex_data$x1))
)
model_or(
y4 | y5 ~ g1 + x1,
    means = means, covs = covs, n = n, predictors = predictors
)
summary(lm(y4 | y5 ~ g1 + x1, data = ex_data))
```

---

**model_prcomp**

*Model the principal component score of a set of phenotypes using PCSS*

**Description**

`model_prcomp` calculates the linear model for the mth principal component score of a set of phenotypes as a function of a set of predictors.

**Usage**

```r
model_prcomp(
    formula, 
    comp = 1, 
    n, 
    means, 
    covs, 
    center = FALSE, 
    standardize = FALSE, 
    ...
)
```
Arguments

**formula**
an object of class formula whose dependent variable is a series of variables joined by + operators. `model_prcomp` will treat a principal component score of those variables as the actual dependent variable. All model terms must be accounted for in `means` and `covs`.

**comp**
integer indicating which principal component score to analyze. Must be less than or equal to the total number of phenotypes.

**n**
sample size.

**means**
named vector of predictor and response means.

**covs**
named matrix of the covariance of all model predictors and the responses.

**center**
logical. Should the dependent variables be centered before principal components are calculated?

**standardize**
logical. Should the dependent variables be standardized before principal components are calculated?

... additional arguments

Value

an object of class "pcsslm".

An object of class "pcsslm" is a list containing at least the following components:

**call**
the matched call

**terms**
the terms object used

**coefficients**
a $px4$ matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value.

**sigma**
the square root of the estimated variance of the random error.

**df**
degrees of freedom, a 3-vector $p, n – p, p^*$, the first being the number of non-aliased coefficients, the last being the total number of coefficients.

**fstatistic**
a 3-vector with the value of the F-statistic with its numerator and denominator degrees of freedom.

**r.squared**
$R^2$, the 'fraction of variance explained by the model'.

**adj.r.squared**
the above $R^2$ statistic 'adjusted', penalizing for higher $p$.

**cov.unscaled**
a $pxp$ matrix of (unscaled) covariances of the $coef[j], j = 1,...p$.

**Sum Sq**
a 3-vector with the model's Sum of Squares Regression (SSR), Sum of Squares Error (SSE), and Sum of Squares Total (SST).

References

Examples

```r
ex_data <- pcsstools_example[c("g1", "x1", "x2", "y1", "y2", "y3")]
head(ex_data)
means <- colMeans(ex_data)
covs <- cov(ex_data)
n <- nrow(ex_data)

model_prcomp(
  y1 + y2 + y3 ~ g1 + x1 + x2,
  comp = 1, n = n, means = means, covs = covs
)
```

Description

model_product approximates the linear model for the product of m phenotypes as a function of a set of predictors.

Usage

```r
model_product(
  formula,
  n,
  means,
  covs,
  predictors,
  responses = NULL,
  response = "continuous",
  ...
)
```

Arguments

- **formula**: an object of class formula whose dependent variable is a combination of variables and * operators. All model terms must be accounted for in means and covs.
- **n**: sample size.
- **means**: named vector of predictor and response means.
- **covs**: named matrix of the covariance of all model predictors and the responses.
- **predictors**: named list of objects of class predictor
- **responses**: named list of objects of class predictor corresponding to all terms being multiplied in the response. Can be left NULL if only multiplying two terms
- **response**: character. Describe distribution of all product terms. Either "continuous" or "binary". If "binary" different approximations of product means and variances are used.
- **...**: additional arguments
**Value**

An object of class "pcsslm".

An object of class "pcsslm" is a list containing at least the following components:

- `call` the matched call
- `terms` the terms object used
- `coefficients` a $p \times 4$ matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value.
- `sigma` the square root of the estimated variance of the random error.
- `df` degrees of freedom, a 3-vector $n - p, p, p^*$, the first being the number of non-aliased coefficients, the last being the total number of coefficients.
- `fstatistic` a 3-vector with the value of the F-statistic with its numerator and denominator degrees of freedom.
- `r.squared` $R^2$, the 'fraction of variance explained by the model'.
- `adj.r.squared` the above $R^2$ statistic 'adjusted', penalizing for higher $p$.
- `cov.unscaled` a $p \times p$ matrix of (unscaled) covariances of the $\text{coef}_j, j = 1, \ldots, p$.
- `Sum Sq` a 3-vector with the model's Sum of Squares Regression (SSR), Sum of Squares Error (SSE), and Sum of Squares Total (SST).

**References**


**Examples**

```r
ex_data <- pcsstools_example[c("g1", "g2", "g3", "x1", "y4", "y5", "y6")]
head(ex_data)
means <- colMeans(ex_data)
covs <- cov(ex_data)
n <- nrow(ex_data)
predictors <- list(
  g1 = new_predictor_snp(maf = mean(ex_data$g1) / 2),
  g2 = new_predictor_snp(maf = mean(ex_data$g2) / 2),
  g3 = new_predictor_snp(maf = mean(ex_data$g3) / 2),
  x1 = new_predictor_normal(mean = mean(ex_data$x1), sd = sd(ex_data$x1))
)
responses <- lapply(means[c("y4", "y5", "y6")], new_predictor_binary)

model_product(
  y4 * y5 * y6 ~ g1 + g2 + g3 + x1,
  means = means, covs = covs, n = n,
  predictors = predictors, responses = responses, response = "binary"
)

summary(lm(y4 * y5 * y6 ~ g1 + g2 + g3 + x1, data = ex_data))
```
model\_singular

Description

model\_singular calculates the linear model for a singular phenotype as a function of a set of predictors.

Usage

model\_singular(formula, n, means, covs, ...)

Arguments

formula an object of class formula whose dependent variable is only variable. All model terms must be accounted for in means and covs.
n sample size.
means named vector of predictor and response means.
covs named matrix of the covariance of all model predictors and the responses.
... additional arguments

Value

an object of class "pcsslm". An object of class "pcsslm" is a list containing at least the following components:
call the matched call
terms the terms object used
coefficients a \( p \times 4 \) matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value.
sigma the square root of the estimated variance of the random error.
df degrees of freedom, a 3-vector \( p, n - p, p^* \), the first being the number of non-aliased coefficients, the last being the total number of coefficients.
fstatistic a 3-vector with the value of the \( F \)-statistic with its numerator and denominator degrees of freedom.
r.squared \( R^2 \), the 'fraction of variance explained by the model'.
adj.r.squared the above \( R^2 \) statistic 'adjusted', penalizing for higher \( p \).
cov.unscaled a \( p \times p \) matrix of (unscaled) covariances of the \( \text{coef}\[j]\), \( j = 1, \ldots, p \).
Sum Sq a 3-vector with the model’s Sum of Squares Regression (SSR), Sum of Squares Error (SSE), and Sum of Squares Total (SST).
References


Examples

```r
# Load example data
ex_data <- pcsstools_example[c("g1", "x1", "y1")]
means <- colMeans(ex_data)
covs <- cov(ex_data)
n <- nrow(ex_data)

# Model with singular coefficients
model_singular(
  y1 ~ g1 + x1,
  n = n, means = means, covs = covs
)
summary(lm(y1 ~ g1 + x1, data = ex_data))
```

new_predictor

Create an object of class "predictor"

Description

Create an object of class "predictor"

Usage

```r
new_predictor(
  f = function() { },
  predictor_type = character(),
  lb, ub,
  support
)
```

Arguments

- `f`: a function that gives the probability mass/distribution function of a random variable.
- `predictor_type`: a character describing the random variable. Either "discrete" or "continuous".
- `lb, ub`: if `predictor_type` == "continuous" double giving the lower/upper bound of the pdf `f`.
- `support`: if `predictor_type` == "discrete" vector of the support of the pmf for `f`.

Value

an object of class "predictor".
new_predictor_binary

See Also

new_predictor_normal, new_predictor_snp and new_predictor_binary.

Examples

call_new_predictor(
    f = function(x0) dnorm(x0, mean = 0, sd = 1),
    predictor_type = "continuous", lb = -Inf, ub = Inf
)

call_new_predictor_binary(p)

Description

new_predictor_binary calls new_predictor

Usage

new_predictor_binary(p)

Arguments

p probability of success (predictor mean)

Value

an object of class "predictor".

Examples

call_new_predictor_binary(p = 0.75)

new_predictor_normal

Shortcut to create a predictor object for a continuous variable

Description

new_predictor_normal calls new_predictor

Usage

new_predictor_normal(mean, sd)
new_predictor_snp

Arguments

mean  predictor mean (double).
sd    predictor standard deviation (double)

Value

an object of class "predictor".

Examples

new_predictor_normal(mean = 10, sd = 1)

new_predictor_snp  Shortcut to create a predictor object for a SNP’s minor allele counts

Description

new_predictor_snp calls new_predictor

Usage

new_predictor_snp(maf)

Arguments

maf    minor allele frequency

Value

an object of class "predictor".

Examples

new_predictor_snp(maf = 0.3)
**pcsslmlm**

*Approximate a linear model using PCSS*

**Description**

`pcsslmlm` approximates a linear model of a combination of variables using precomputed summary statistics.

**Usage**

`pcsslmlm(formula, pcss = list(), ...)`

**Arguments**

- `formula`: an object of class formula whose dependent variable is a combination of variables and logical operators. All model terms must have appropriate PCSS in `pcss`.
- `pcss`: a list of precomputed summary statistics. In all cases, this should include `n`: the sample size, `means`: a named vector of predictor and response means, and `covs`: a named covariance matrix including all predictors and responses. See Details for more information.
- `...`: additional arguments. See Details for more information.

**Details**

`pcsslmlm` parses the input `formula`'s dependent variable for functions such as sums (+), products (*), or logical operators (| and &). It then identifies models the combination of variables using one of `model_combo`, `model_product`, `model_or`, `model_and`, or `model_prcomp`.

Different precomputed summary statistics are needed inside `pcss` depending on the function that combines the dependent variable.

- For linear combinations (and principal component analysis), only `n`, `means`, and `covs` are required
- For products and logical combinations, the additional items `predictors` and `responses` are required. These are named lists of objects of class predictor generated by `new_predictor`, with a predictor object for each independent variable in `predictors` and each dependent variable in `responses`. However, if only modeling the product or logical combination of only two variables, `responses` can be `NULL` without consequence.

If modeling a principal component score of a set of variables, include the argument `comp` where `comp` is an integer indicating which principal component score to analyze. Optional logical arguments `center` and `standardize` determine if responses should be centered and standardized before principal components are calculated.

If modeling a linear combination, include the argument `phi`, a named vector of linear weights for each variable in the dependent variable in `formula`.

If modeling a product, include the argument `response`, a character equal to either "continuous" or "binary". If "binary", specialized approximations are performed to estimate means and variances.
Value

an object of class "pcsslm".

An object of class "pcsslm" is a list containing at least the following components:

call the matched call
terms the terms object used
coefficients a px4 matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value.
sigma the square root of the estimated variance of the random error.
df degrees of freedom, a 3-vector $p, n - p, p^*$, the first being the number of non-aliased coefficients, the last being the total number of coefficients.
fstatistic a 3-vector with the value of the F-statistic with its numerator and denominator degrees of freedom.
r.squared $R^2$, the ‘fraction of variance explained by the model’.
adj.r.squared the above $R^2$ statistic ‘adjusted’, penalizing for higher $p$.
cov.unscaled a pxp matrix of (unscaled) covariances of the $\text{coef}[j], j = 1, \ldots, p$.
Sum Sq a 3-vector with the model’s Sum of Squares Regression (SSR), Sum of Squares Error (SSE), and Sum of Squares Total (SST).

References


See Also

model_combo, model_product, model_or, model_and, and model_prcomp.

Examples

## Principal Component Analysis

ex_data <- pcsstools_example[c("g1", "x1", "y1", "y2", "y3")]
pcss <- list(
  means = colMeans(ex_data),
  covs = cov(ex_data),
  n = nrow(ex_data))
## Linear combination of variables

```
ex_data <- pcsstools_example[c("g1", "g2", "y1", "y2")]
pcss <- list(
  means = colMeans(ex_data),
  covs = cov(ex_data),
  n = nrow(ex_data)
)

pcssl(y1 + y2 + y3 ~ g1 + x1, pcss = pcss, comp = 1)

## Product of variables

ex_data <- pcsstools_example[c("g1", "x1", "y4", "y5", "y6")]

pcss <- list(
  means = colMeans(ex_data),
  covs = cov(ex_data),
  n = nrow(ex_data),
  predictors = list(
    g1 = new_predictor_snp(maf = mean(ex_data$g1) / 2),
    x1 = new_predictor_normal(mean = mean(ex_data$x1), sd = sd(ex_data$x1))
  ),
  responses = lapply(
    colMeans(ex_data)[3:length(colMeans(ex_data))],
    new_predictor_binary
  ),
)

pcssl(y4 * y5 * y6 ~ g1 + x1, pcss = pcss, response = "binary")
```

## Disjunct (OR statement) of variables

```
ex_data <- pcsstools_example[c("g1", "x1", "y4", "y5")]

pcss <- list(
  means = colMeans(ex_data),
  covs = cov(ex_data),
  n = nrow(ex_data),
  predictors = list(
    g1 = new_predictor_snp(maf = mean(ex_data$g1) / 2),
    x1 = new_predictor_normal(mean = mean(ex_data$x1), sd = sd(ex_data$x1))
  )
)

pcssl(y4 | y5 ~ g1 + x1, pcss = pcss)
```
Description

The pcsstools package provides functions to describe various regression models using only pre-computed summary statistics (PCSS) from genome-wide association studies (GWASs) and PCSS repositories.

Details

The main function of interest is `pcsslm`, which can be used to approximate linear models for various combinations of phenotypes using PCSS.

pcsstools_example

Simulated example data

Description

A dataset containing simulated genetic data with 3 SNPs, 3 continuous covariates, and 6 continuous phenotypes.

Usage

`pcsstools_example`

Format

A data frame with 1000 rows and 12 columns:

- `g1,g2,g3` Minor allele counts at three sites
- `x1,x2,x3` Continuous covariates
- `y1,y2,y3` Continuous phenotypes
- `y4,y5,y6` Binary phenotypes
print.pcsslm

Print an object of class pcsslml

Description

Prints a linear model fit through pre-computed summary statistics

Usage

```r
## S3 method for class 'pcsslml'
print(
  x,
  digits = max(3L,getOption("digits") - 3L),
  symbolic.cor = x$symbolic.cor,
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

- `x` an object of class "pcsslml"
- `digits` the number of significant digits to use when printing.
- `symbolic.cor` logical. If TRUE, print the correlations in a symbolic form (see symnum) rather than as numbers.
- `signif.stars` logical. If TRUE, 'significance stars' are printed for each coefficient.
- `...` further arguments passed to or from other methods.

Value

An object of class "pcsslml".

An object of class "pcsslml" is a list containing at least the following components:

- `call` the matched call
- `terms` the terms object used
- `coefficients` a px4 matrix with columns for the estimated coefficient, its standard error, t-statistic and corresponding (two-sided) p-value.
- `sigma` the square root of the estimated variance of the random error.
- `df` degrees of freedom, a 3-vector `p, n − p, p*`, the first being the number of non-aliased coefficients, the last being the total number of coefficients.
- `fstatistic` a 3-vector with the value of the F-statistic with its numerator and denominator degrees of freedom.
- `r.squared` $R^2$, the 'fraction of variance explained by the model'.
- `adj.r.squared` the above $R^2$ statistic 'adjusted', penalizing for higher p.
- `cov.unscaled` a pxp matrix of (unscaled) covariances of the coef[j], j = 1,...p.
- `Sum Sq` a 3-vector with the model’s Sum of Squares Regression (SSR), Sum of Squares Error (SSE), and Sum of Squares Total (SST).
Author(s)

R Core Team and contributors worldwide. Modified by Jack Wolf
Index

* datasets
  pcsstools_example, 30

anova.pcsslm, 2
anova.pcsslm.list(anova.pcsslm), 2
approx_and, 3
approx_conditional, 4
approx_mult_prod, 5
approx_or, 6
approx_prod_stats, 7
approx_response_cov_recursive, 8

calculate_lm, 9
calculate_lm_combo, 10
check_terms, 12

effect_strength, 12
extract_response, 13

get_pcor, 13
guess_response, 14

make_permutations, 14
model_and, 15, 27, 28
model_combo, 16, 27, 28
model_or, 18, 27, 28
model_prcomp, 19, 27, 28
model_product, 21, 27, 28
model_singular, 23

new_predictor, 24, 27
new_predictor_binary, 25, 25
new_predictor_normal, 25, 25
new_predictor_snp, 25, 26

pcsslm, 27, 30
pcsstools, 30
pcsstools_example, 30
print.pcsslm, 31

symnum, 31