

# Package ‘yhat’

January 2, 2012

**Type** Package

**Title** Interpreting Regression Effects

**Version** 1.0-3

**Date** 2009-12-09

**Author** Kim Nimon <kim.nimon@gmail.com>, J. Kyle Roberts <kylar@smu.edu>

**Maintainer** Kim Nimon <kim.nimon@gmail.com>, J. Kyle Roberts <kylar@smu.edu>

**Depends** R (>= 2.7.0), MBESS, yacca

**Description** The purpose of this package is to provide methods for variance partitioning for linear models and canonical correlation and methods for interpreting regression effects using beta weights, standardized beta weights, structure coefficients, and adjusted effect sizes.

**License** GPL (>= 2)

**URL** <http://www.r-project.org>

**LazyLoad** yes

**Repository** CRAN

**Date/Publication** 2009-12-10 19:56:43

## R topics documented:

yhat-package	2
canonCommonality	3
canonVariate	4
commonalityCoefficients	5
effect.size	7
genList	8
odd	8
regr	9
setBits	10

<b>Index</b>	<b>12</b>
--------------	-----------

**Description**

The purpose of this package is to provide methods for variance partitioning for linear models and canonical correlation and methods for interpreting regression effects using beta weights, standardized beta weights, structure coefficients, and adjusted effect sizes.

**Details**

Package: yhat  
Type: Package  
Version: 1.0  
Date: 2009-07-02  
License: GPL (>= 2)  
LazyLoad: yes

The `regr` reports beta weights, standardized beta weights, structure coefficients, adjusted effect sizes, and commonality coefficients.

The `commonalityCoefficients` function is based on Mood's (1969) procedure for computing commonality analysis formulas for any number of  $k$  predictor variables. The function outputs a list of two tables. The first table lists the commonality coefficients and the percentage of variance associated with each effect. The second table provides a total of the unique and common effects for each independent variable.

The `canonCommonality` function produces commonality data for both canonical variables sets. Variables in a given canonical set are used to partition the variance of the canonical variates produced from the *other* canonical set and vice versa. Commonality data is supplied for the number of canonical functions requested.

Canonical commonality analysis is not substantively different from regression commonality analysis developed by Mood (1969). In fact, Beaton (1973) recognized multivariate commonality as a simple generalization of univariate commonality, indicating that the two techniques provide identical results when one canonical set contains only a single variable.

**Author(s)**

Kim Nimon <kim.nimon@gmail.com>, J. Kyle Roberts <kylar@smu.edu>

**References**

Beaton, A. E. (1973) *Commonality*. (ERIC Document Reproduction Service No. ED111829)

Butts, C. T. (2009). *yacca: Yet Another Canonical Correlation Analysis Package*. R package version 1.1.

Mood, A. M. (1969) Macro-analysis of the American educational system. *Operations Research*, **17**, 770-784.

Nimon, K., Lewis, M., Kane, R. & Haynes, R. M. (2008) An R package to compute commonality coefficients in the multiple regression case: An introduction to the package and a practical example. *Behavior Research Methods*, **40**(2), 457-466.

**See Also**

[regr commonalityCoefficients canonCommonality](#)

canonCommonality      *Commonality Coefficients for Canonical Correlation*

**Description**

The canonCommonality function produces commonality data for both canonical variables sets. Variables in a given canonical set are used to partition the variance of the canonical variates produced from the *other* canonical set and vica versa. Commonality data is supplied for the number of canonical functions requested.

**Usage**

```
canonCommonality(A, B, nofns = 1)
```

**Arguments**

A	Matrix containing variable set A
B	Matrix containing variable set B
nofns	Number of canonical functions to analyze

**Details**

The function canonCommonality has two required arguments and one optional argument. The first two arguments contain the two variable sets. The third argument is optional and defines the number of canonical functions to analyze. Unless specified, the number of canonical functions defaults to 1.

The function canonCommonality calls a function canonVariate to decompose canonical varites twice: the first time for the variable set identified in the first argument, the second time for the variable set identified in the second argument.

**Value**

The function `canonCommonality` returns commonality data for both canonical variable sets. For the number of functions requested, both canonical variates are analyzed. For each canonical variate analyzed, two tables are returned. The first table lists the commonality coefficients and their contribution to the total effect, while the second table lists the unique and common effects for each regressor. The function returns the resulting output ordering the output according to the function's parameters.

**Author(s)**

Kim Nimon <kim.nimon@gmail.com>

**See Also**

[canonVariate](#)

**Examples**

```
## Example parallels the R builtin cancel and the
## yacca cca example
data(LifeCycleSavings)
pop <- LifeCycleSavings[, 2:3]
oec <- LifeCycleSavings[, -(2:3)]
## Perform Commonality Coefficient Analysis
canonCommonData<-canonCommonality(pop,oec,1)
print(canonCommonData)

## Uses HS dataset in MBESS
data(HS.data)
attach(HS.data)
## Create canonical variable sets
MATH_REASON<-HS.data[,c("deduct", "problemr")]
MATH_FUND<-HS.data[,c("numeric", "arithmet", "addition")]
## Perform Commonality Coefficient Analysis
canonCommonData<-canonCommonality(MATH_FUND,MATH_REASON,1)
print(canonCommonData)
```

---

canonVariate

*Canonical Commonality Analysis*

---

**Description**

The `canonCommonality` function produces commonality data for a given canonical variable set. Using the variables in a given canonical set to partition the variance of the canonical variates produced from the *other* canonical set, commonality data is supplied for the number of canonical functions requested.

**Usage**

```
canonVariate(A, B, nofns)
```

**Arguments**

A	Matrix containing variable set A
B	Matrix containing variable set B
nofns	Number of canonical functions to analyze

**Details**

For each canonical function, `canonVariate`: (a) creates a dataset that combines the matrix of variables for a given canonical set and the canonical variate for the *other* canonical set; (b) calls `commonalityCoefficients`, passing the dataset, the name of the canonical variate, and the names of the variates in a given canonical set; (c) saves resultant output.

**Value**

The function `canonVariate` returns commonality data for the canonical variable set input. For the number of functions requested, two tables are returned. The first table lists the commonality coefficients for each canonical function together with its contribution to the total effect, while the second table lists the unique and common effects for each regressor.

**Note**

This function is internal to `canonCommonality`, called during runtime and passed the appropriate parameters. This is not an end-user function.

**Author(s)**

Kim Nimon <kim.nimon@gmail.com>

**See Also**

[canonCommonality](#)

---

commonalityCoefficients

*Commonality Coefficients*

---

**Description**

Commonality Coefficients returns a list of two tables. The first table CC contains the list of commonality coefficients and the percent variance for each effect. The second CCTotByVar totals the unique and common effects for each independent variable.

**Usage**

```
commonalityCoefficients(dataMatrix, dv, ivlist, imat=FALSE)
```

**Arguments**

dataMatrix	Dataset containing the dependent and independent variables
dv	The dependent variable named in the dataset
ivlist	List of independent variables named in the dataset
imat	Echo flag, default to FALSE

**Details**

When echo flag is true, transitional matrices during commonality coefficient calculation are sent to output window. Default for this option is false. When set to true, the intermediate matrices for each commonality coefficient and regression combinations are printed in the output window.

**Value**

CC	Matrix containing commonality coefficients and percentage of variance for each effect.
CCTotalByVar	Table of unique and common effects for each independent variable.

**Author(s)**

Kim Nimon <kim.nimon@gmail.com>

**See Also**

[canonCommonality](#) [genList](#) [odd](#) [setBits](#)

**Examples**

```
## Predict miles per gallon based on vehicle weight, type of
## carbtorator, & number of engine cylinders
CCdata=commonalityCoefficients(mtcars,"mpg",list("wt","carb","cyl"))
print(CCdata)

## Predict paragraph comprehension based on four verbal
## tests: general info, sentence comprehension, word
## classification, & word type
## Uses HS dataset in MBESS
data(HS.data)
attach(HS.data)
## Commonality Coefficient Analysis
CCdata=commonalityCoefficients(HS.data,"paragrap",list("general",
"sentence","wordc","wordm"))
print(CCdata)
```

---

`effect.size`*Effect Size Computation for lm*

---

**Description**

Creates adjusted effect sizes for linear regression.

**Usage**

```
effect.size(lm.out)
```

**Arguments**

`lm.out`            Output from `lm` class object

**Details**

The function `effect.size` produces a family of effect size corrections for the R-squared metric produced from an `lm` class object. Suggestions for recommended correction are supplied, based on Yin and Fan (2001).

**Value**

Returns adjusted R-squared metric.

**Author(s)**

J. Kyle Roberts <kylerr@smu.edu>

**References**

Yin, P., & Fan, X. (2001) Estimated  $R^2$  shrinkage in multiple regression: A comparison of different analytical methods. *The Journal of Experimental Education*, **69**(2), 203-224.

**See Also**

[regr.yhat](#)

**Examples**

```
data(HS.data)
attach(HS.data)
lm.out<-lm(deduct~addition*arithmet)
effect.size(lm.out)
```

---

genList                      *Generate List R^2 Values*

---

**Description**

Use the bitmap matrix to generate the list of  $R^2$  values needed.

**Usage**

```
genList(ivlist, value)
```

**Arguments**

ivlist	List of independent variables in dataset
value	Number of variables

**Details**

Returns the number of  $R^2$  values that will be calculated in output tables.

**Value**

Returns newList from generate list function call.

**Note**

This function is internal to [commonalityCoefficients](#), called during runtime and passed the appropriate parameters. This is not an end-user function.

**Author(s)**

Kim Nimon <kim.nimon@gmail.com>

---

odd                              *isOdd Function*

---

**Description**

Function receives value and returns true if value is odd.

**Usage**

```
odd(val)
```

**Arguments**

val	Value to check
-----	----------------

**Details**

Determines value of parameter in argument.

**Value**

Returns true when value checked is odd. Otherwise, function returns a value false.

**Note**

This function is internal to `commonalityCoefficients`, called during runtime and passed the appropriate parameters. This is not an end-user function.

**Author(s)**

Kim Nimon <kim.nimon@gmail.com>

---

 regr

*Regression effect reporting for lm class objects*


---

**Description**

The `regr` reports beta weights, standardized beta weights, structure coefficients, adjusted effect sizes, and commonality coefficients for `lm` class objects.

**Usage**

```
regr(lm.out)
```

**Arguments**

`lm.out`            `lm` class object

**Details**

The function `regr` takes the `lm` class object and reports beta weights, standardized beta weights, structure coefficients, adjusted effect sizes, and commonality coefficients for `lm` class objects.

**Value**

```
return(list(LM_Output=summary(lm.out), Beta_Weights=beta.out, Structure_Coefficients=structure.coef,
Commonality_Data=CCdata[1], Effect_Size=es, Comment="The Effect Size recommendations are
based on Yin and Fan (2001). Your dataset may take on a different covariance structure, thus making
another effect size estimate more appropriate."))
```

`LM_Output`            The summary of the output from the `lm` class object  
`Beta_Weights`        Beta weights for the regression effects  
`Structure_Coefficients`  
                       Structure coefficients for the regression effects

Commonality\_Data      Commonality coefficients for the regression effects. The output only produces a parsed version of CCdata

Effect\_Size      Adjusted effect size computations based on R<sup>2</sup> adjustments

**Author(s)**

J. Kyle Roberts <kyle@smu.edu>, Kim Nimon <kim.nimon@gmail.com>

**References**

Nimon, K., Lewis, M., Kane, R. & Haynes, R. M. (2008) An R package to compute commonality coefficients in the multiple regression case: An introduction to the package and a practical example. *Behavior Research Methods*, **40**(2), 457-466.

Yin, P., & Fan. X. (2001) Estimated R<sup>2</sup> shrinkage in multiple regression: A comparison of different analytical methods. *The Journal of Experimental Education*, **69**(2), 203-224.

**See Also**

Functions in the yhat library [commonalityCoefficients](#), [effect.size](#)

**Examples**

```
data(HS.data)
attach(HS.data)
lm.out<-lm(deduct~addition*arithmet)
regr(lm.out)
```

---

setBits

*Decimal to Binary*

---

**Description**

Creates the binary representation of n and stores it in the nth column of the matrix.

**Usage**

```
setBits(col, effectBitMap)
```

**Arguments**

col      Column of matrix to represent in binary image

effectBitMap      Matrix of mean combinations in binary form

**Details**

Creates the binary representation of col and stores it in its associated column.

**Value**

Returns matrix effectBitMap of mean combinations in binary form.

**Note**

This function is internal to [commonalityCoefficients](#), called during runtime and passed the appropriate parameters. This is not an end-user function.

**Author(s)**

Kim Nimon <[kim.nimon@gmail.com](mailto:kim.nimon@gmail.com)>

# Index

- \*Topic **models**
  - commonalityCoefficients, 5
- \*Topic **multivariate**
  - canonCommonality, 3
  - canonVariate, 4
- \*Topic **package**
  - yhat-package, 2
- \*Topic **regression**
  - commonalityCoefficients, 5

canonCommonality, 3, 3, 5, 6  
canonVariate, 4, 4  
commonalityCoefficients, 3, 5, 8–11

effect.size, 7, 10

genList, 6, 8

odd, 6, 8

regr, 3, 7, 9

setBits, 6, 10

yhat, 7  
yhat (yhat-package), 2  
yhat-package, 2