Package ‘Simpsons’

February 19, 2015

Type Package

Title Detecting Simpson's Paradox

Version 0.1.0

Date 2012-08-17

Author Rogier Kievit & Sacha Epskamp

Depends R (>= 2.14.0), mclust

Maintainer Rogier Kievit <rogierkievit@gmail.com>

Description This package detects instances of Simpson's Paradox in datasets. It examines subpopulations in the data, either user-defined or by means of cluster analysis, to test whether a regression at the level of the group is in the opposite direction at the level of subpopulations.

License GPL-2

ByteCompile yes

Repository CRAN

Date/Publication 2012-08-23 14:41:48

NeedsCompilation no

R topics documented:

Simpsons-package *Detecting Simpson's Paradox*

Description

This package detects instances of Simpson's Paradox in datasets of bivariate continuous data . It examines subpopulations in the data, either user-defined or by means of cluster analysis, to test whether a regression at the level of the group is in the opposite direction at the level of subpopulations.

Details

Package:	Simpsons
Type:	Package
Version:	1.0
Date:	2012-08-17
License:	GPL-2

Author(s)

Rogier Kievit & Sacha Epskamp

Maintainer: Rogier Kievit <rogierkievit@gmail.com>

References

Fraley, C., & Raftery, A. E. (1998a) MCLUST: Software for model-based cluster and discriminant analysis. Department of Statistics, University of Washington: Technical Report No.342.

Fraley, C., & Raftery, A. E. (1998b). How many clusters? Which clustering method? - Answers via model-based cluster analysis. Department of Statistics, University of Washington: Technical Report no. 329.

Kievit, R.A., Frankenhuis, W. & Borsboom, D. (in preparation). Simpson's Paradox in Psychological Science: A Practical Guide.

Simpson, E. H. (1951). The interpretation of interaction in contingency tables. Journal of the Royal Statistical Society, Ser. B, 13, 238-241.

Examples

```
## Not run:
#example 1. Here, we want to regress 'Coffee' on 'Neuroticism',
#taking into account possible gender differences.
#Simulating 100 males
coffeem=rnorm(100,100,15)
neuroticismm=(coffeem*.8)+rnorm(100,15,8)
clusterid=rep(1,100)
```

```
males=cbind(coffeem,neuroticismm,clusterid)
coffeef=rnorm(100,100,15)
neuroticismf=160+((coffeef*-.8)+rnorm(100,15,8))
clusterid=rep(2,100)
females=cbind(coffeef,neuroticismf,clusterid)
data=data.frame(rbind(males,females))
colnames(data) <- c("Coffee","Neuroticism","gender")

example1=Simpsons(Coffee,Neuroticism,clusterid=gender, data=data)
example1


## End(Not run)
```

`cluster` *Extract clustered subsets*

Description

After running a Simpsons, the function ’cluster’ extracts desired clusters from "Simpsons" object.

Usage

```
cluster(x, clusters)
```

Arguments

`x`	Object of class Simpson
`clusters`	Define which clusters should be returned. Can range from 1 to maximum number of clusters detected in the Simpsons analysis.

Details

Returns list

Value

Returns list

Author(s)

Rogier Kievit <rogierkievit@gmail.com> & Sacha Epskamp <mail@sachaepskamp.com>

References

Kievit, R.A., Frankenhuis, W. E. , Waldorp, L. J. & Borsboom, D. (in preparation). Simpson’s Paradox in Psychological Science: A Practical Guide. http://rogierkievit.com/simpsonsparadox.html

Examples

```
## Not run:
#example 2. Here we estimate the relationship between 'Coffee' and #'Neuroticism'.
#As opposed to example 1, we have not measured any possible clustering #identifiers
#such as gender, so we want to estimate whether there is evidence for #clustering based
#only on the data we measured: Coffee and Neuroticism.

#generating data
Coffee1=rnorm(100,100,15)
Neuroticism1=(Coffee1*.8)+rnorm(100,15,8)
g1=cbind(Coffee1, Neuroticism1)
Coffee2=rnorm(100,170,15)
Neuroticism2=(300-(Coffee2*.8)+rnorm(100,15,8))
g2=cbind(Coffee2, Neuroticism2)
Coffee3=rnorm(100,140,15)
Neuroticism3=(200-(Coffee3*.8)+rnorm(100,15,8))
g3=cbind(Coffee3, Neuroticism3)
data2=data.frame(rbind(g1,g2,g3))
colnames(data2) <- c("Coffee","Neuroticism")

example2=Simpsons(Coffee,Neuroticism,data=data2)
cluster(example2,2) #extracts data belonging to cluster 2
cluster(example2,c(1,3) #extracts all datapoints belonging to clusters 1 and 3

## End(Not run)
```

---

coef.Simpson *Coef method for "Simpson"*

Description

Matrix of all regresssion results: Each individual cluster and the whole dataset (Alldata), their sample sizes, and regression estimates (beta, intercept and uncorrected p-value).

Usage

```
## S3 method for class 'Simpson'
coef(object, ...)
```

Arguments

`object`	Specify object of class 'Simpson'
`...`	Not used.

Value

Returns dataframe called 'Res' which contains clusters numbers, their sample size, and regression estimates (beta and intercept) for variables X and Y defined in the object.

Author(s)

Rogier Kievit <rogierkievit@gmail.com> & Sacha Epskamp <mail@sachaepskamp.com>

References

Kievit, R.A., Frankenhuis, W. E. , Waldorp, L. J. & Borsboom, D. (in preparation). Simpson's Paradox in Psychological Science: A Practical Guide. http://rogierkievit.com/simpsonsparadox.html

Examples

```
## Not run:
#Simulating 100 males
coffeem=rnorm(100,100,15)
neuroticismm=(coffeem*.8)+rnorm(100,15,8)
clusterid=rep(1,100)
males=cbind(coffeem,neuroticismm,clusterid)

#Simulating 100 females
coffeef=rnorm(100,100,15)
neuroticismf=160+((coffeef*-.8)+rnorm(100,15,8))
clusterid=rep(2,100)
females=cbind(coffeef,neuroticismf,clusterid)
data=data.frame(rbind(males,females))
colnames(data) <- c("Coffee","Neuroticism","gender")
example1=Simpsons(Coffee,Neuroticism,clusterid=gender, data=data)
coef(example1)



## End(Not run)
```

plot.Simpson *Plot method*

Description

Plots bivariate scatterplot of all data using colour coding for clusters. Also draws each individual regression and group regression.

Usage

```
## S3 method for class 'Simpson'
plot(x, ...)
```

Arguments

x

...

Author(s)

Rogier Kievit <rogierkievit@gmail.com> & Sacha Epskamp <mail@sachaepskamp.com>

print.Simpson *Print method*

Description

Print method.

Usage

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

Arguments

`x`

`...`

Author(s)

Rogier Kievit <rogierkievit@gmail.com> & Sacha Epskamp <mail@sachaepskamp.com>

Simpsons *Simpsons*

Description

This package detects instances of Simpson's Paradox in datasets of bivariate continuous data. It examines subpopulations in the data, either user-defined or by means of cluster analysis, to test whether a regression at the level of the group is in the opposite direction at the level of subpopulations.

Usage

```
Simpsons(X, Y, clusterid, clustervars, data, nreps = 5000)
```

Arguments

`X` The first continuous variable to be used in the regression analysis.

`Y` The second continuous variable to be used in the regression analysis.

`clusterid` If you have a vector describing group membership, such as gender, you can specify it here. This will then be used to test for possible instances of Simpson's Paradox. If left empty, a cluster analysis will attempt to discover clusters in the data. See example 1.

`clustervars` By default, the cluster analysis will be carried out on the X and Y variables. If you want to define the clusters on the basis of a different set of variables, such as a questionnaire, you can specify them using this command. See example 3.

`data` Describes the data matrix. Should be a dataframe.

`nreps` nreps specifies the number of permutations run for each cluster in the permutation significance test. The default is 5000. Each repetition is stored in the matrix 'permutationtest'.

Details

This package detects instances of Simpson's Paradox in datasets. That is, it tests whether some bivariate relationship found at the level of the whole dataset is consistent (in direction and strength) for possible subpopulations. It examines whether there is evidence for more than one cluster in the data in the data using cluster analysis, either user-defined or by means of cluster analysis. Then, it plots the data, using a different color for every cluster, plots the regression lines for each cluster, and estimates the regression of X on Y for each cluster. Finally, it tests whether the regression at the level of the whole dataset is different from the regression at the level of the subclusters.

Because clusters in the data are part of the whole dataset, and therefore create a dependency, a permutation test is used to test for significant differences. For each cluster, the cluster labels are permuted within the whole dataset, the regression is run within the cluster and the whole dataset, and the difference between these two betas is stored as 1 repetition of the null distribution and stored in the object 'permutationtest'. A regression is considered significantly different from the group if the difference in beta estimate exceeds the lower or upper 2,5 percent of the permuted null distribution. If this is the case, a warning is issued as follows: "Warning: Beta regression estimate in cluster X is significantly different compared to the group!". If the sign of the regression within a cluster is different (positive or negative) than the sign for the group and the beta estimate deviates significantly, a warning states "Sign reversal: Simpson's Paradox! Cluster X is significantly different and in the opposite direction compared to the group!"

Value

A list of class "Simpson" with the following elements:

`Nclusters` Number of clusters estimated in the data (or the number of different groups in the 'clusterid' column defined by the user).

`clustersize` the size of each estimated cluster

`alldata` the original dataset with the clusterid's appended as a new column

`Allbeta` A matrix of beta estimates for each cluster

`Allint` A matrix of intercepts for each cluster

permutationtest
: The matrix of all permutations. The columns define the clusters, the rows specify the difference in the beta of the group and the beta of that cluster for each iteration, thus generating the null distribution

namex
: The first variable used in the analysis

namey
: The second variable used in the analysis

pvalues
: The p-values for the significance of the regressions

mclustanalysis
: Object of class Mclust that contains all mclust results

Author(s)

Rogier Kievit <rogierkievit@gmail.com> & Sacha Epskamp <mail@sachaepskamp.com>

References

Kievit, R.A., Frankenhuis, W. E. , Waldorp, L. J. & Borsboom, D. (in preparation). Simpson's Paradox in Psychological Science: A Practical Guide. http://rogierkievit.com/simpsonsparadox.html

Examples

```
## Not run:
#This section contains three examples of the types of analyses you can run
#using the 'Simpsons' function, illustrating the commmands and the types of #output.

#Example 1. Here, we want to estimate the relationship between 'Coffee'
#and 'Neuroticism', taking into account possible gender differences.
#As we have measured gender, we supply this information using the #'clusterid' command.
#This means that the function runs the analysis both for
#the dataset as a whole and within the two subgroups.
#It then checks whether the subgroups deviate significantly
#from the regression at the level of the group.

#Simulating 100 males
coffeem=rnorm(100,100,15)
neuroticismm=(coffeem*.8)+rnorm(100,15,8)
clusterid=rep(1,100)
males=cbind(coffeem,neuroticismm,clusterid)

#Simulating 100 females
coffeef=rnorm(100,100,15)
neuroticismf=160+((coffeef*-.8)+rnorm(100,15,8))
clusterid=rep(2,100)
females=cbind(coffeef,neuroticismf,clusterid)

data=data.frame(rbind(males,females))
colnames(data) <- c("Coffee","Neuroticism","gender")

#'normal' data analysis: Plot & regression
plot(data[,1:2])
a=lm(data[,1]~data[,2])
abline(a)
```

```
summary(a) #A normal regression shows no effect

#Running a Simpsons Paradox analysis, using gender as known clustering #variable
example1=Simpsons(Coffee,Neuroticism,clusterid=gender, data=data)
# Analyze the relationship between coffee and neuroticism for both males
# and females.
example1



#example 2. Here we estimate the relationship between 'Coffee' and 'Neuroticism'.
#As opposed to example 1, we have not measured any possible clustering #identifiers
#such as gender, so we want to estimate whether there is evidence for #clustering based
#only on the data we measured: Coffee and Neuroticism.

#generating data
Coffee1=rnorm(100,100,15)
Neuroticism1=(Coffee1*.8)+rnorm(100,15,8)
g1=cbind(Coffee1, Neuroticism1)
Coffee2=rnorm(100,170,15)
Neuroticism2=(300-(Coffee2*.8)+rnorm(100,15,8))
g2=cbind(Coffee2, Neuroticism2)
Coffee3=rnorm(100,140,15)
Neuroticism3=(200-(Coffee3*.8)+rnorm(100,15,8))
g3=cbind(Coffee3, Neuroticism3)
data2=data.frame(rbind(g1,g2,g3))
colnames(data2) <- c("Coffee","Neuroticism")

#'normal' data analysis: Plot & regression
plot(data2)
b=lm(data2[,1]~data2[,2])
summary(b)
abline(b)

# Running the analysis tool identifies three clusters, and warns that the relationship
between alcohol and coffee is in the opposite direction in two of the subclusters.
example2=Simpsons(Coffee,Neuroticism,data=data2)
example2

#example3:

#In this final example, we want again want to analyse the relationship
# between 'Alcohol' and 'Mood'. However, this time
#we have reason to believe that responses to a questionnaire
#will fall into clusters of response types. Therefore, we want to
# estimate the clusters in the data on the basis of a different set
# of variables. In this case, we have simulate three types of responses
# to a questionnaire of nine questions, with continuous responses
#ranging between 1 and 7. We then first estimate the clusters on
#the basis of the questionnaire, and then examine the relationship
#between 'Alcohol' and 'Mood' based on these detected clusters.

#group 1
```

```
signal=matrix(rnorm(300,7,1),100,3)
noise=matrix(rnorm(600,3.5,1),100,6)
g1=cbind(signal,noise)

#group 2
signal=matrix(rnorm(300,1,1),100,3)
noise=matrix(rnorm(600,3.5,1),100,6)
g2=cbind(noise, signal)

#group 3
signal=matrix(rnorm(300,7,1),100,3)
noise1=matrix(rnorm(300,3.5,1),100,3)
noise2=matrix(rnorm(300,3.5,1),100,3)
g3=cbind(noise1,signal,noise2)

questionnaire=rbind(g1,g2,g3)
colnames(questionnaire)=c('q1','q2','q3','q4','q5','q6','q7','q8','q9')

Alc1=rnorm(100,10,8)
Mood1=(Alc1*.4)+rnorm(100,3,4)
A=cbind(Alc1, Mood1)
Alc2=rnorm(100,15,8)
Mood2=(Alc2*-.4)+rnorm(100,3,4)
B=cbind(Alc2,Mood2)
Alc3=rnorm(100,20,8)
Mood3=(Alc3*.8)+rnorm(100,3,4)
C=cbind(Alc3,Mood3)
data=data.frame(rbind(A,B,C))
colnames(data) <- c(”Alcohol”,”Mood”)
alldata=cbind(questionnaire,data)
alldata=as.data.frame(alldata)

#Run Simpsons Paradox detection algorithm, clustering on the basis of the questionnaire
example3=Simpsons(Alcohol,Mood,clustervars=c(”q1”,”q2”,'q3','q4',
'q5','q6','q7','q8','q9'),data=alldata)
example3

## End(Not run)
```

summary.Simpson *Summary method*

Description

Matrix of all regression results: Each individual cluster and the whole dataset (Alldata) of all clusters, their sample size, and regression estimates (beta and intercept).

Usage

```
## S3 method for class 'Simpson'
summary(object, ...)
```

Arguments

`object`	
`...`	Not used.

Value

Returns list called ’Res’. The first object contains clusters numbers, their sample size, and regression estimates (beta and intercept) for variables X and Y defined in the object. The second object is an object of class Mclust, and contains all diagnostics of the cluster analysis. For more details, see package Mclust by Fraley & Raftery.

Author(s)

Rogier Kievit <rogierkievit@gmail.com> & Sacha Epskamp <mail@sachaepskamp.com>

References

Kievit, R.A., Frankenhuis, W. E. , Waldorp, L. J. & Borsboom, D. (in preparation). Simpson’s Paradox in Psychological Science: A Practical Guide. http://rogierkievit.com/simpsonsparadox.html

Fraley, C., & Raftery, A. E. (1998a) MCLUST: Software for model-based cluster and discriminant analysis. Department of Statistics, University of Washington: Technical Report No.342.

Fraley, C., & Raftery, A. E. (1998b). How many clusters? Which clustering method? - Answers via model-based cluster analysis. Department of Statistics, University of Washington: Technical Report no. 329.

Examples

```
## Not run:
#Example
#Simulating 100 males
coffeem=rnorm(100,100,15)
neuroticismm=(coffeem*.8)+rnorm(100,15,8)
clusterid=rep(1,100)
males=cbind(coffeem,neuroticismm,clusterid)

#Simulating 100 females
coffeef=rnorm(100,100,15)
neuroticismf=160+((coffeef*-.8)+rnorm(100,15,8))
clusterid=rep(2,100)
females=cbind(coffeef,neuroticismf,clusterid)
data=data.frame(rbind(males,females))
colnames(data) <- c("Coffee","Neuroticism","gender")
example1=Simpsons(Coffee,Neuroticism,clusterid=gender, data=data)
summary(example1)

## End(Not run)
```

Index