CAMBRIDGE ASSESSMENT

An empirical assessment of Guttman's Lambda 4 reliability coefficient

Tom Benton

Paper presented at International Meeting of the Psychometric Society, Arnhem, July 2013.

Research Division
Cambridge Assessment
1, Regent Street
Cambridge
CB2 1GG

Benton.T@cambridgeassessment.org.uk

www.cambridgeassessment.org.uk

Abstract

Numerous alternative indices for test reliability have been proposed as being superior to Cronbach's alpha. One such alternative is Guttman's L4 (Guttman, 1945). This is calculated by dividing the items in a test into two halves such that the covariance between scores on the two halves is as high as possible. However, although simple to understand and intuitively appealing, the method can potentially be severely positively biased if the sample size is small or the number of items in the test is large (Ten Berge and Socan, 2004).

To begin with this paper compares a number of available algorithms for calculating L4. We then empirically evaluate the bias of L4 for 51 separate upper secondary school examinations taken in the UK in June 2012. For each of these tests we have evaluated the likely bias of L4 for a range of different sample sizes. The results show that the positive bias of L4 is likely to be small if the estimated reliability is larger than 0.85, if there are less than 25 items and if a sample size of more than 3000 is available. A sample of size of 1000 may be sufficient if the estimate of L4 is above 0.9. For lower reliabilities, greater numbers of items, or smaller sample sizes it will be necessary to evaluate bias on a case by case basis.

Introduction

The reliability of test is defined as the extent to which the result achieved by any pupil would be repeated if the entire exercise were replicated (Brennan, 2001). In particular we are often interested in the extent to which pupils' results would change had a different (but equally valid) set of items been used in the test rather than those that were actually included. Conceptually, the aim is to try to calculate the likely correlation between scores on the test actually sat by pupils and another (theoretical) test designed to the same specification.

Answering the above question has become a fairly routine task within psychometrics. Whilst the most commonly applied metric used to quantify reliability is Cronbach's alpha (Cronbach, 1951), research suggests that in many cases this may not be the most appropriate technique and will underestimate the true reliability of a test (Sijtsma, 2009, Revelle and Zinbarg, 2009).

An alternative method to calculate reliability is Guttman's[1] L4 (Guttman, 1945). The concept behind the method is quite simple. Reliability is calculated by first splitting a test into two halves. For example, this might be all the odd numbered versus all the even numbered questions, or all the questions in the first half of a test versus all the questions in the second half. Now the covariance between the scores pupils achieve on each half is calculated. The variance of the total test score (that is, including both halves) is also calculated. The overall test reliability can now be calculated by the formulae below.

$$Reliability = \frac{4Covariance(Half\ 1\ scores, Half\ 2\ scores)}{Variance(Total\ score\ on\ test)}$$

Although the above formula can be applied to any split half, L4 is generally taken to mean the reliability from the split that maximises this coefficient.

Although L4 is an appealing reliability coefficient in terms of being easy to understand and being less likely to underestimate reliability than Cronbach's alpha, it has two notable drawbacks. Firstly, routines to calculate L4 are not included in most standard statistical packages. Secondly, as has been noted by Ten Berge and Socan (2004) there is the danger that L4 may *overestimate* reliability if there are a large number of items or if the sample size is small.

[1] Although most subsequent literature refers to this reliability index as "Guttman's", the same coefficient appears in an earlier paper (Kelly, 1942) which in turn refers to the issue of calculating split-half reliability being "nicely taken care of by the following formula, due to Dr. John Flanagan".

This paper intends to address both of these drawbacks. The first issue will be addressed by evaluating the performance of two recently published R packages in terms of their ability to accurately identify L4. Furthermore, R code for two further methods to calculate L4 is provided in the appendix of this paper. To address to second issue we shall empirically evaluate the bias of L4 for a number of real assessments and examine how this varies dependent upon the sample size and the number of items in the test.

Algorithms to find L4

There are a number of possible algorithms that could be used to find the optimal split of the items into two halves.

1. An exhaustive search of all possible splits to identify the split leading to the highest reliability, although, such a method will be computationally demanding if our test has a large number of items.
2. A reduced exhaustive search, where, to begin with, pairs of items that are highly correlated are deemed to be in opposite halves. This method is applied in the R package *Lambda4* written by Tyler Hunt and published in 2012[2].
3. A cluster analysis based method drawing on the item correlation matrix. This method is applied in the R package *psych*[3] by William Revelle and first published in 2007[4].
4. The method of Callender and Osburn (1977) based on sequentially adding one item to each half so as to maximise the reliability coefficient at each step.
5. A method based on beginning with an initial split of the items into two groups and then iteratively improving the reliability by swapping items until no further improvements are possible. This procedure is relatively straightforward and only requires the item covariance matrix as an input. It works from the fact that if X is the total current score on half 1, Y is the total current score on half 2, and we wish to switch items Xi and Yj to opposite sides then the improvement in the covariance between the two halves that will be yielded by the switch is

$$\begin{aligned}&\mathrm{Cov}(X-Xi+Yj,Y+Xi-Yj)-\mathrm{Cov}(X,Y)\\ =&2\mathrm{Cov}(Xi,Yj)+\mathrm{Cov}(X,Xi)+\mathrm{Cov}(Y,Yj)-\mathrm{Cov}(X,Yj)-\mathrm{Cov}(Y,Xi)-V(Xi)-V(Yj).\end{aligned}$$

   All of these terms can be quickly calculated from the item covariance matrix. This allows us to identify the best possible swap and then recalculate possible improvements for subsequent swaps.

For this last method there are clearly a number of options for how to split items into two groups to start with. For many assessments, because items dealing with similar subjects are often placed consecutively in a test, a split into odd and even items may provide a sensible starting point. However, in order to increase our chances of identifying the best possible split, we may prefer to try several different starting splits and see which leads to the largest reliability coefficient overall. Hadamard matrices provide a possible effective method for trying numerous different starting splits as they can ensure that each new starting split is as different as possible from starting splits that have been tried before.

R code for methods 1 and 5 is provided in the appendix[5] along with code showing how method 5 can be applied from multiple different starting values derived from a Hadamard matrix[6].

[2] Available from http://cran.r-project.org/web/packages/Lambda4/index.html.
[3] Available from http://cran.r-project.org/web/packages/psych/index.html.
[4] Although the functions for finding L4 were not introduced until 2009.
[5] The code in the appendix also applies to adjustments proposed by Raju (1977) and Feldt (1975) for cases where the split halves may be of unequal length.

Evaluation of alternative algorithms

Each of the methods described in the previous section were evaluated against data from 51 separate upper secondary school examinations taken in the UK in June 2012. These tests each contain between 10 and 37 questions[7] and were each taken by a minimum of 5000 candidates. The number of available marks per question ranged between 1 and 22 with both the mean and the median number of available marks per question equal to 5. For each of these assessments, L4 was calculated using each of the algorithms described in the previous section. Because the *psych* package works from the correlation matrix rather than the covariance matrix, all item scores were standardised before applying any of the methods[8].

Table 1 shows the results of analysis in terms of how often each algorithm identifies the best possible split of those that were identified. The table also shows the mean and median L4s from each algorithm and well as the largest amount by which the algorithm underestimated the best L4. As can be seen, the algorithm used in the *psych* package failed to find the best possible split in any of the 51 assessments. In general the reliabilities estimated by this method were not too far below the optimum (0.02 on average) but could be as high as 0.05 below the actual maximum L4. The Callender-Osburn algorithm performed a little better, identifying the optimal split in 4 out of 51 cases. More importantly, the estimated reliability from this algorithm was never more than 0.02 below the maximum L4. The algorithm in the *Lambda4* package also failed to find the optimal split for the majority of assessments. Having said this, the differences between the L4 estimated by this algorithm and the maximum L4 tended to be extremely small; roughly 0.002 on average and never more than 0.01. The start-then-improve algorithm based on starting with odd and even question numbers identified the best split for over half the assessments (28 out of 51). Once again, where this algorithm failed to find the optimum split, the difference from the largest L4 tended to be very small. The start-then-improve algorithms tended to identify the best possible split for almost all assessments if an additional five random starting splits were used (46 out of 51), and for all assessments if a Hadamard matrix was used to provide additional starting splits.

[6] Hadamard matrices are generated using the *survey* package published by Thomas Lumley and available from http://cran.r-project.org/web/packages/survey/index.html (Lumley, 2004).

[7] Whole question scores were analysed for the purposes of calculating reliability rather than items from the same question stem. This was to avoid the possibility of irrelevant associations between item scores within the same question spuriously inflating the reliability estimate.

[8] The same analysis was also run with unstandardized item scores. The results were very similar.

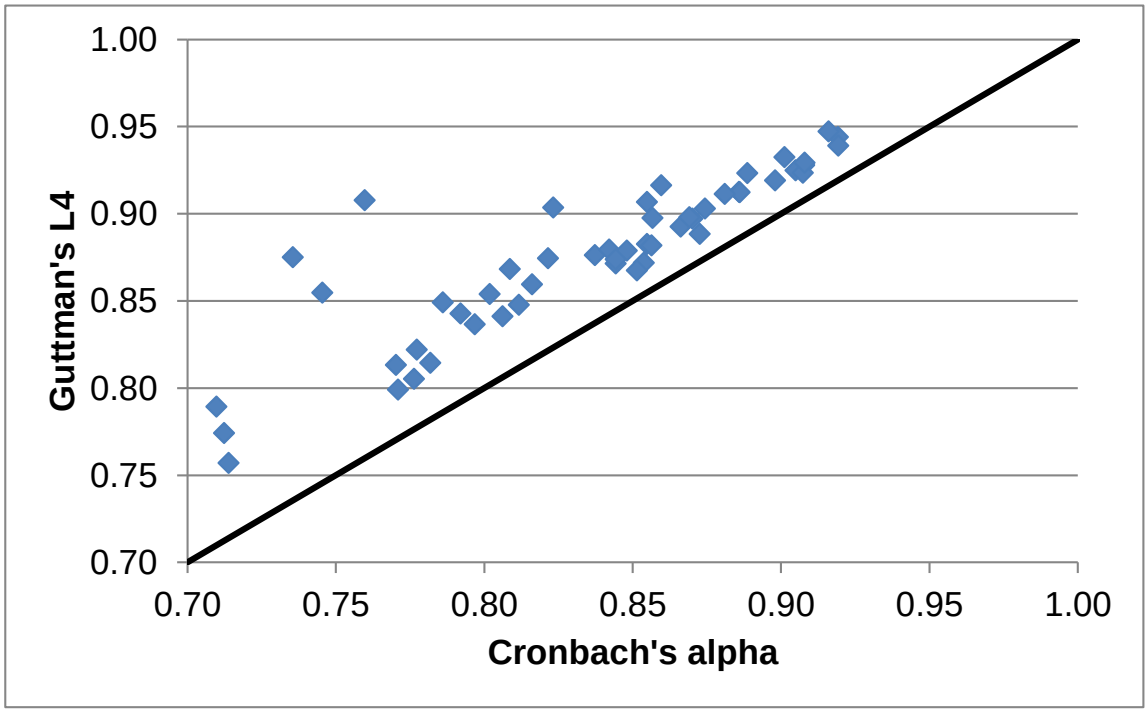
Table 1: Relative performance of different methods of optimising L4

Algorithm used to maximise L4	Number of times largest L4 identified (out of 51)	Mean L4	Median L4	Furthest distance below largest L4
R package *psych*	0	0.846	0.856	0.050
Callender-Osburn algorithm	4	0.861	0.867	0.020
R package *Lambda4*	13	0.863	0.868	0.010
Start-then-improve (Odd/Even start)	28	0.864	0.868	0.008
Start-then-improve (Odd/Even and 5 other random starts)	46	0.865	0.869	0.002
Start-then-improve (Odd/Even and 12 further starts from Hadamard matrix)	51	0.865	0.869	0.000

Thirty-nine of the 51 assessments contained 15 questions or fewer. For these assessments the algorithm based upon exhaustive search was also applied. In every case, the best split identified by exhaustive search matched the split identified by the start-then-improve algorithm using a Hadamard matrix.

A plot of Cronbach's alpha for each of these assessments against the maximised value of L4 is shown in figure 1. As can be seen the value of L4 is universally larger than the value of alpha (as we would expect). On average there was a difference of 0.04 between the two reliability indices, although, as can be seen, for some assessments the difference was somewhat larger than this.

Figure 1: The relationship between estimated Cronbach's alpha and L4.

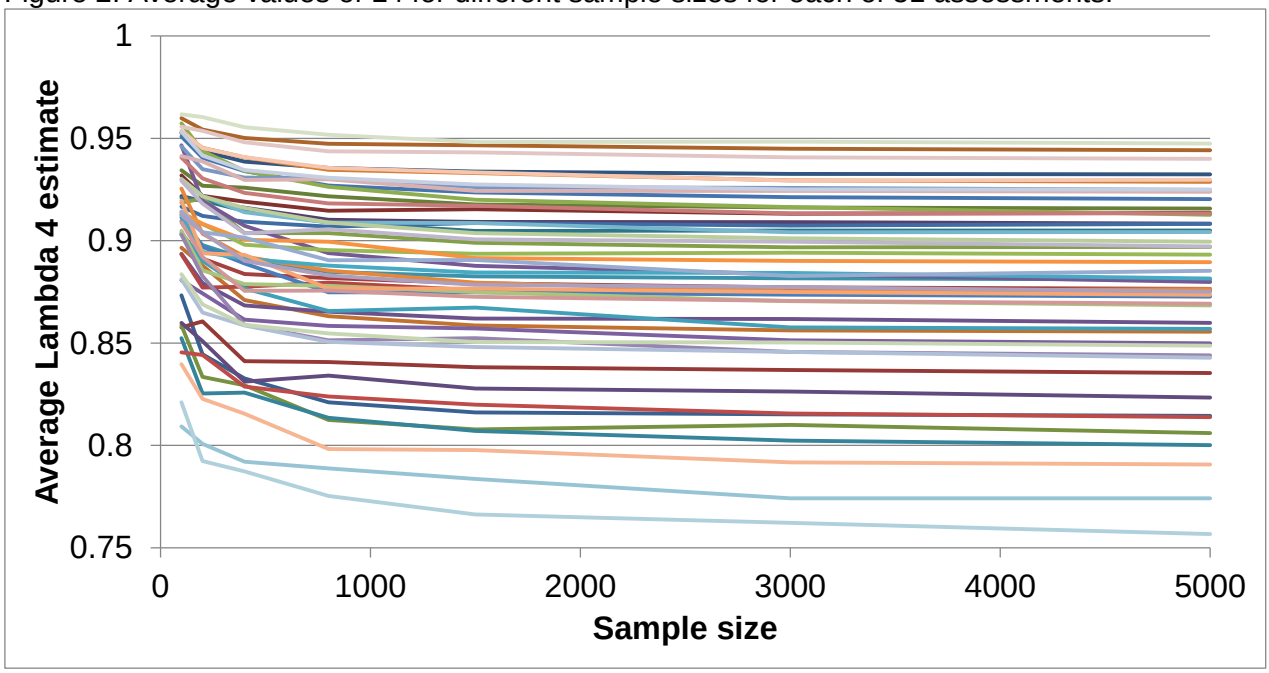


Examining the level of positive bias in L4

Having identified an efficient algorithm to calculate L4, we now turn our attention to the issue of how the likely positive bias of L4 changes dependent upon the sample size and the number of items.

For each of the 51 assessments, ten samples at each of sizes 100, 200, 400, 800, 1500, 3000 and 5000 were drawn from the available data. L4 was calculated[9] for each of the samples and the average reliability coefficient was computed for each sample size for each assessment. The results of this analysis are shown in figure 2. As can be seen, for each of the assessments, there is a tendency for the estimated value of L4 to decrease as the sample size increases. The rate of decrease is particularly evident for smaller sample sizes, indicating that in such cases L4 is likely to be severely positively biased.

Figure 2: Average values of L4 for different sample sizes for each of 51 assessments.



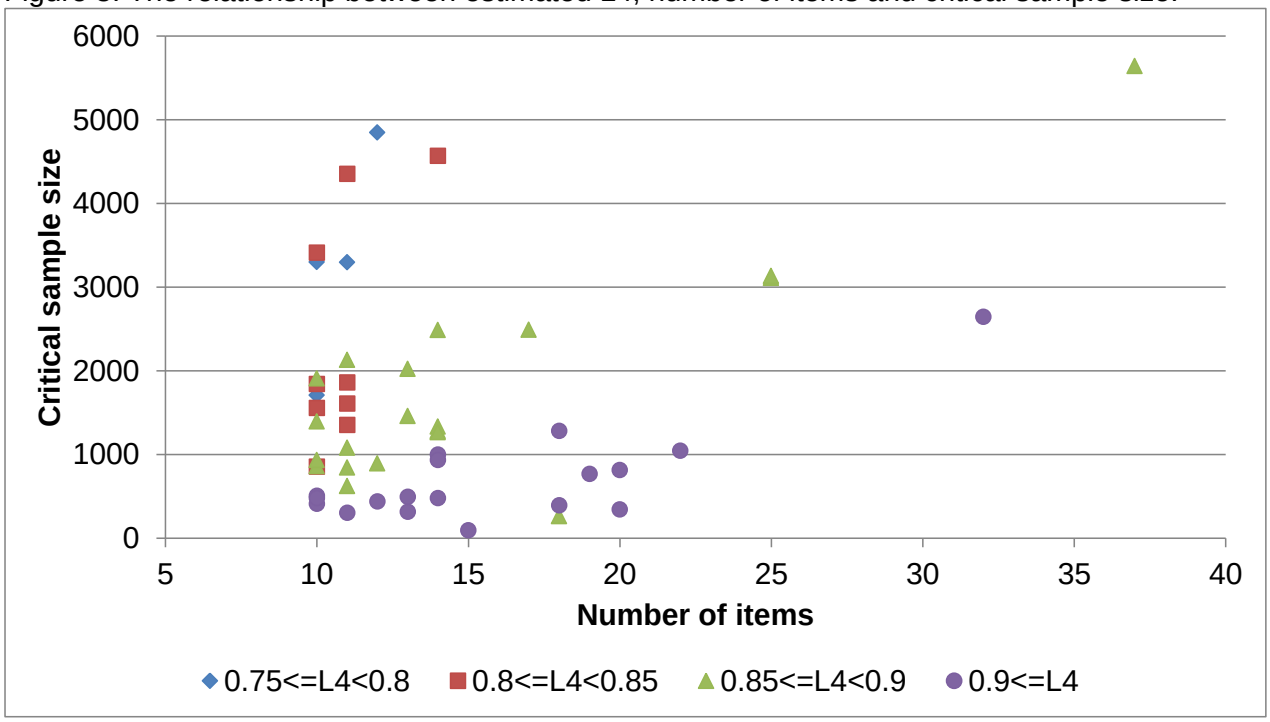
Using the results in figure 2, it is possible to estimate a bias corrected version of L4 based upon the method suggested by Verhelst (2000). This method involves a regression analysis of estimated L4 values on the reciprocal of the square root of the sample size. The intercept of this regression analysis (that is, the predicted value of L4 for an infinite sample size) is then a bias corrected estimated of L4. This procedure was applied for each of the 51 assessments allowing us to identify the estimated bias of L4 for each sample size. Of particular interest was identifying the sample size where the bias of L4 was likely to fall below 0.01 meaning that for most practical purposes the estimate could be treated as unbiased. These required sample sizes were termed the *critical sample size*.

The critical sample size required is plotted against the number of items for all 51 assessments in figure 3. Different coloured points are used to identify assessments with different levels of L4. For assessments with high levels of L4 (above 0.85) it can be seen that there is a fairly clear relationship between the number of items on the test and the critical sample size. On a practical note we can see that if we have less than 25 items then a sample size of 3000 appears to be always sufficient in these cases. Furthermore, if the estimated L4 is greater than 0.9 a sample size of 1000 appears to be usually sufficient. However, where the size of L4 is below 0.85, the relationship between the number of items and the required sample size is less clear cut. For the small number of such assessments included in this analysis, sample sizes between 1000 and 5000 were required with little evidence of the required sample size being closely determined by the

[9] This time without standardising item scores before beginning.

number of items. This indicates that, for assessments with lower estimated reliabilities, it is probably necessary to make an assessment of the likely positive bias of L4 on a case by case basis. This may prove particularly difficult for small sample sizes as it will require greater amount of extrapolation from the regression method used to generate bias corrected estimates.

Figure 3: The relationship between estimated L4, number of items and critical sample size.



Conclusion

Guttman's L4 provides a reliability coefficient that is relatively simple to understand and can be easily computed using code written in R. In addition to the code provided in the appendix of this paper, the *Lambda4* package by Tyler Hunt also appears to provide a robust estimation method.

Our analysis has confirmed that L4 can suffer from positive bias for small sample sizes. Positive bias is less likely to be an issue is the estimated value of L4 is above 0.85, if the number of items is below 25, and if the sample size is bigger than 3000. Potentially, a sample size of 1000 may be sufficient if the estimated value of L4 is greater than 0.9. However, our analysis shows that if the estimated value of L4 is below 0.85 it is difficult to identify the necessary sample size dependent upon the number of items. In such cases the likely bias of the method should be evaluated on a case by case basis.

References

Brennan, R. (2001). An Essay on the History and Future of Reliability from the Perspective of Replications, *Journal of Educational Measurement*, 38, 295-317.

Callender, J, and Osburn H (1977). A Method for Maximizing and Cross-Validating Split-Half Reliability Coefficients. *Educational and Psychological Measurement*, 37, 819-826.

Cronbach, L. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, 16, 297-334.

Feldt, L. (1975). Estimation of the reliability of a test divided into two parts of unequal length. *Psychometrika*, 40, 557-561.

Guttman, L. (1945). A basis for analysing test-retest reliability. *Psychometrika*, 10, 255-282.

Kelley, T. (1942). The reliability coefficient. *Psychometrika*, 7, 75-83.

Lumley, T. (2004) Analysis of complex survey samples. *Journal of Statistical Software*, 9, 1-19.

Revelle, W., and Zinbarg, R. (2009). Coefficients Alpha, Beta, Omega, and the glb: Comments on Sijtsma. *Psychometrika*, 74, 145-154.

Raju, N. (1977). A generalization of coefficient alpha. *Psychometrika*, 42, 549-565.

Sijtsma, K. (2009). On the use, the misuse and the very limited usefulness of Cronbach's alpha. *Psychometrika*, 74, 107-120.

Ten Berge, J., and Socan, G. (2004). The greatest lower bound to the reliability of a test and the hypothesis of unidimensionality. *Psychometrika*, 69, 613-625.

Verhelst, N. (2000). *Estimating the reliability of a test from a single test administration*. Arnhem: CITO.

Appendix: R code to find best split using the "start-then-improve" algorithm

```
#Function to find best split half from a given starting split
MaxSplitHalf<-function(data,xal){
#data – matrix of items scores (row=candidates,column=items)
#xal – vector of 0s and 1s specifying initial split
nite<-ncol(data)
cov1<-cov(data)
v<-diag(cov1)
yal<-1-xal
ones<-rep(1,nite)
covxy<-t(xal)%*%cov1%*%yal

#Code to examine all possible swaps
maxchg1=9;
while(maxchg1>0){
#Calculate change for swapping items in X and Y;
#This is equal to 2covxiyj+covxix+covyyj-vx-vy-covxiy-covxyj;
covxiyj<-cov1
covxix<-(cov1%*%xal)%*%t(ones)
covyyj<-ones%*%(yal%*%cov1)
vx<-v%*%t(ones)
vy<-t(vx)
covxiy<-(cov1%*%yal)%*%t(ones)
covxyj<-ones%*%(xal%*%cov1)
result<-2*covxiyj+covxix+covyyj-vx-vy-covxiy-covxyj
for (i in 1:nite){for (j in 1:nite){if (xal[i]==xal[j]){result[i,j]=0}}}
#Add bits for swapping with no other item
result<-cbind(result,as.vector(cov1%*%xal-cov1%*%yal-v)*xal)
result<-rbind(result,c(as.vector(cov1%*%yal-cov1%*%xal-v)*yal,0))
#find indices of maximum change;
maxchg=0
maxchgx=0
maxchgy=0
which1=which(result==max(result),arr.ind=TRUE)[1,]
if (result[which1[1],which1[2]]>0){maxchgx=which1[1]
maxchgy=which1[2]
maxchg=result[which1[1],which1[2]]}
maxchg1<-maxchg
if (maxchgx>0 & maxchgx<(nite+1)) {xal[maxchgx]=0}
if (maxchgy>0 & maxchgy<(nite+1)) {xal[maxchgy]=1}
if (maxchgx>0 & maxchgx<(nite+1)) {yal[maxchgx]=1}
if (maxchgy>0 & maxchgy<(nite+1)) {yal[maxchgy]=0}
covxy<-t(xal)%*%cov1%*%yal}

guttman<-4*covxy/sum(cov1)
pites<-sum(xal)/nite
raju<-covxy/(sum(cov1)*pites*(1-pites))

v1<-t(xal)%*%cov1%*%xal
v2<-t(yal)%*%cov1%*%yal
feldt<-4*covxy/(sum(cov1)-((v1-v2)/sqrt(sum(cov1)))**2);

res<-list(guttman=as.vector(guttman),
      raju=as.vector(raju),
      feldt=as.vector(feldt),
      xal=xal)
return(res)}
```

```
#Maximise L4 starting from odd/even and 12 splits from 12x12 Hadamard matrix
library(survey)
MaxSplitHalfHad12<-function(data){
#data - matrix of items scores (row=candidates,column=items)
#start with odd vs even
nite<-ncol(data)
sequence<-1:nite
xal<-(sequence%%2)
res1<-MaxSplitHalf(data,xal)
#now try 12 further splits based on 12*12 Hadamard matrix
had<-hadamard(11)
for (iz in 1:12){
nextra<-max(nite-12,0)
resrand<-MaxSplitHalf(data,c(had[,iz],rep(0,nextra))[1:nite])
if (resrand$guttman>res1$guttman){res1<-resrand}}
return(res1)}


#Maximise using exhaustive search
library(Lambda4)
MaxSplitExhaustive<-function(data){
#data - matrix of items scores (row=candidates,column=items)
cov1<-cov(data)
nite<-dim(data)[2]
mat1<-(bin.combs(nite)+1)/2
res1<-list(guttman=0,xal=rep(-99,nite))
for (jjz in 1:length(mat1[,1])){
xal<-mat1[jjz,]
gutt1<-4*(t(xal)%*%cov1%*%(1-xal))/sum(cov1)
resrand<-list(guttman=gutt1,xal=xal)
if (resrand$guttman>res1$guttman){res1<-resrand}}
return(res1)}

#Examples of use (using data from the Lambda4 package)
data(Rosenberg)
MaxSplitHalf(Rosenberg,c(0,1,0,1,0,1,0,1,0,1))
MaxSplitHalfHad12(Rosenberg)
MaxSplitExhaustive(Rosenberg)
```