A Function in R-code for Top-down Correlation

Darshi Arachige

05/11/2021

In many situations where several raters are involved, we compare how closely their ratings agree. Following are two possible scenarios.

1. Credit Risk Analytics:

Proportions of new customers rateded by a credit rating system over the year can be ranked by the magnitudes of those proportions and see whether these rankings agree with the ones assigned over several years. If the rankings do not agree the stability of the ranking system or the population can be further investigated. Rating providers or credit models can also be compared to see their concordance.

2. Text-mining:

In a bibliometric setting, a set of historical figures based on the ranks by the number of mentions in the books of comparable histories written by different authours can be compared to see their place in history in the eyes of authour's generations.

The test that pops up in one's mind for more than 2 raters is Kendall's test for concordance (Sheskin, 2004). However, in the situations like above, someone may prefer to use a weighted method that emphasises the higher ranks. Top-down correlation proposed by Iman and Conover (1987) is a technique suitable for such applications. Even though there are many other methods, Top-down approach was selected here for its exponential weighting scheme on the ranks. The R function described below has been extended to calculate not only correlation and test for concordance based on Iman and Conover (1987) but also Spearman correlation and Kendall's coefficient. As I could not find any publicly available program to do the top-down method, it was decided to write a new function to check some manual calculations. Hence, this post is created to share it with anyone who would like it too.

This function reads in data as a set of rankings in columns and the objects that are ranked in rows. No row identifiers are used in the function. It considers that the number 1 denotes the highest rank, and the first column represents the first set of rankings. To select the top-down methods or the standard methods (no weighting) it uses a switch that should be set to 1 for the standard approaches. Function call can be like "output<-concord(Topdown,6,1)" where "Topdown" is a dataset with 6 columns of rankings that would have been imported as R dataset. By default, data will be sorted in ascending order of the first column of the dataset and printed in the output as Scores. The output provides the correlation or concordance as well as the distribution-based probabilities for significance testing (Iman and Conover, 1987; Sheskin, 2004).

References

Iman R L and Conover W J (1987). A measure of Top - Down correlation. Technometrics, 29(3), 351-357

Sheskin D J (2004) Handbook of Parametric and Nonparametric Statistical Procedures. Chapman & Hall/CRC, Third Edition, pp 1093-1107

R- Function

R – packages required are:

```
library(readxl)

library(dplyr)
```

library(plyr)

```
concord<-function (data,n,s=0){
  'This function includes the tie corrections'
  m0<-as.matrix(data)
  nc<-ncol(m0)
  if(n<2){stop("Incorrect input for n")}
  if(nc>n){
    ncg<-rep(n+1:nc)
    m<-m0[,-ncg]
  }
  else{ if(nc==n){
    m<-m0
  }
    else{stop("Please check the number of variables and the entered value for n and re-run.")}

  }

  m2<-m[order(m[,1], decreasing = FALSE),]

  m1<-matrix(1:1,dim(data)[1],dim(data)[1])
  v<-array(0,dim(data)[1])
  v1<-array(0,dim(data)[1])
  v2<-array(0,dim(data)[1])
  v3<-array(0,dim(data)[1])
  Ti<-array(0,dim(data)[1])
  m1[lower.tri(m1)]<-0

  v2<-c(1:dim(data)[1])
  m3<-1/v2
  savage<-matrix(0:0,dim(data)[1],n)
  m4<-matrix(0:0,dim(data)[1],2*n)
  m4[,1:n]<-m2
  for (i in 1:n){
    if (s == 0) {
      v2<-m3
      v3<-sort(v2,decreasing=TRUE)
      v<-m1%*%v3
      od<-order(m2[,i])
      v1<-v[order(od)]
      m4[,i+n]<-v1}

    if (s ==1){
      v3<-v2
      v<-v3
      od<-order(m2[,i])
      v1<-v[order(od)]
      m4[,i+n]<-v1
    }
    kmpbind<-data.frame(cbind(group=m4[,i],cc=m4[,i+n]))
    kmpcount<-data.frame(ddply(kmpbind,.(group),summarize,mean=mean(cc)))
    group<-data.frame(group=m4[,i])
```

```
    kmp<-left_join(group,kmpcount, by= 'group')
    savage[,i]<-kmp$mean
    savagesum<-apply(savage,1,sum)
    savagesumsq<-(savagesum^2)
    savagesumsqr<-sum(savagesumsq)
    counter<-data.frame(ddply(kmpbind,.(group),summarize,N=length(cc)))
    Ti[i]<- sum(((counter[,2])^3-counter[,2]))
    Ticor<-sum(Ti)
    c1<-1/(n^2*(dim(data)[1]-savage[1,1]))
    c2<-(savagesumsqr-n^2*dim(data)[1])
    df<-as.numeric(as.numeric(length(savagesum))-1)
    if (n >2) { if(s==0){rt<-"Not defined for more than 2 variables"
    Ct<-as.numeric(c1*c2)
    Ctt<-n*(dim(data)[1]-1)*as.numeric(c1*c2)
    pr<-pchisq(Ctt, df,lower.tail=FALSE)}}
    if(s==1){rt<-"Not defined for more than 2 variables"
    U<-12*(savagesumsqr)-3*n^2*(dim(data)[1])*((dim(data)[1])+1)^2
    Ct<-U/((n^2*(dim(data)[1])*(((dim(data)[1])^2)-1))-(n*Ticor))
    Ctt<-n*(dim(data)[1]-1)*Ct
    pr<-pchisq(Ctt, df,lower.tail=FALSE)}
    if (n ==2) {
      if(s==0) {
        Ct<-"Not defined for 2 variables"
        rt<-(sum(savage[,1]*savage[,2])-dim(data)[1])/(dim(data)[1]-savage[1,1])
        if(dim(data)[1]<200){rz<-rt*(dim(data)[1]-2)^0.5/(1-rt^2)^0.5
        df<-dim(data)[1]-2
        pr<-pt(rz, df=df, lower.tail = FALSE)}
        if(dim(data)[1]>=200){rz<-rt*(dim(data)[1]-1)^0.5
        pr<-pnorm(rz, mean = 0, sd = 1, lower.tail = FALSE)}
      }
      else{ Ct<-"Not defined for 2 variables"
      d2<-sum((savage[,1]-savage[,2])^2)
      x2<-(((dim(data)[1]^3-(dim(data)[1])))) -Ti[1])/12
      y2<-(((dim(data)[1]^3-(dim(data)[1])))) -Ti[2])/12
      rt<-(x2+y2-d2)/(2*sqrt(x2*y2))
      if(dim(data)[1]<200){rz<-rt*(dim(data)[1]-2)^0.5/(1-rt^2)^0.5
      df<-dim(data)[1]-2
      pr<-pt(rz, df=df, lower.tail = FALSE)}
      if(dim(data)[1]>=200){rz<-rt*(dim(data)[1]-1)^0.5
      pr<-pnorm(rz, mean = 0, sd = 1, lower.tail = FALSE)}
      }
    }
  }
  print("The following list outputs the data used, correlation or Tau value and the associated one tail
  print(list(Scores=savage,Tau=Ct, Corr= rt,df=df,Probabilty=pr))
}
```