対応分析

Sample dataは折を見てアップします。

|  |
| --- |
| install.packages('Rcmdr')  library(Rcmdr)  install.packages('MASS')  #簡単に説明すると、コレスポンデンス分析（対応分析）とは、質問項目ごとの主成得点と個々のデータの主成分分析の散布図を重ね書きして、その対応関係を視覚的に把握する分析だと思えば良い。MCAというコマンドは、多重コレスポンデンス分析をするコマンドで、データは例として示したデータのよう数値ではなくて、ラベル化した文字で書かれている必要がある。ある質問に対して５件法（「そう思う。ややそう思う。どちらでもない。ややそうは思わない。そうは思わない。」のように程度を表す回答を選択させる。）や７件法のような形で答えさせるアンケート調査を、プロビット変換などを使って連続関数にするという操作をせずに、質的データのまま分析したいときに使うと便利である。必要があれば、プロビット変換など何らかの方法で量的データに変換して主成分分析・因子分析などに持って行けば良いが、多くの場合、変換の妥当性については検討にしようがない。そのようなことを考えると、アンケート調査の分析などでは、とりあえず、質的データのままコレスポンデンス分析、あるいは、多重コレスポンデンス分析をとりあえずやるのは無駄がない。  library(pillar)  library(MASS)  library(ggplot2)  library(FactoMineR)  qdma<-MCA(data2,graph = TRUE)  qdma  #各主成分のの固有値を表示（第一列が固有値、第二列が分散比（100%)、第三列が累積分散比）  qdma$eig  #項目の主成分得点  qdma$var$coord  #観測データの主成分得点  qdma$ind$coord  #項目について第五主成分までの散布図行列表示  pairs(qdma$var$coord)  #観測データについて第五主成分までの散布図行列を表示  pairs(qdma$ind$coord)  #累積分散比と散布図行列を参考に、どの要因を使って対応分析をするか決める（この例では第一、第二成分を選択）  #データフレーム作成。  qdma\_var\_df<-qdma$var$coord  qdma\_obs\_df<-qdma$ind$coord  df1<-data.frame(x=qdma\_var\_df[,1],y=qdma\_var\_df[,2])  df2<-data.frame(x=qdma\_obs\_df[,1],y=qdma\_obs\_df[,2])  #ggplot2で重ね合わせた散布図を作成  g<-ggplot(NULL)  g<-g+geom\_point(data=df1,aes(x,y,color="red"))+geom\_point(data=df2,aes(x,y),colour="black")  print(g)  #一応、、ここれで目的は達成されるが、対応分析だから、ドットにラベルを付ける。ラベルをつけるときに注意しなければならないのは、データフレームの形が一致していないと重ね書きができないことである。そこで、ラベルをつけない観測データの方のフレームには、ｚ="null"を入れておく。layerという機能を使えば、列の数が一致していなくてもかさねがきができるが、その場合、図の位置やわくのおおきさをしていするひつようがありめんどうなので、列の数を一致させた方が楽である。  df3<-data.frame(x=qdma\_var\_df[,1],y=qdma\_var\_df[,2],z=rownames(qdma\_var\_df))  df2<-data.frame(x=qdma\_obs\_df[,1],y=qdma\_obs\_df[,2],z="null")  f<-ggplot(df3,aes(x,y,label=z))+geom\_point(data=df3,aes(x,y,colour="answer"))+geom\_text(size=3,hjust=0,vjust=0,colour="red")+geom\_point(data=df2,aes(x,y),colour="black")  print(f)  #以下はグラフの装飾。。背景を白にする。  f<-f+theme\_bw()  print(f)  #グラフタイトルをつける。  f<-f+ggtitle("corespondence relationship in PC1 and PC")  print(f)  #縦軸横軸の説明  f<-f+xlab("PC1")+ylab("PC2")  print(f)  write.table(qdma$var$coor,sep=",","obs.csv")  #普通のcorrespondence analysis１形式にすれば、多重対応分析（multiple correspondence analysis　MCA)として使える。  library(MASS)  library(ggplot2)  library(FactoMineR)  ca<-corresp(Pinpricordata,nf=7)  eig<-ca$cor^2  round(eig,3)  biplot(ca)  write.table(eig,"Pinpricoreig7.csv",sep=",")  write.table(ca$cscore,"Pinpricorcolumn7.csv",sep=",")  write.table(ca$rscore,"Pinpricorrow7.csv",sep=",")    #5件法の平行分析:correspondence analysis  library(MASS)  library(ggplot2)  library(FactoMineR)  library(psych)  #準備  rm(Mateig0)  #O1データのランダムなdatasetを作る。  #j=1として、固有値の集計表の第一列を作る。  #シミュレーションの元になる、布全ての乱数表を導入  #データセットのitemsu数をni,組み合わせの数をnc,必要なセット数をnsとする。  #条件を読み込む  dr1<-alt  ni1<-13  nc1=5  ns<-99  Mateig0<-matrix(nrow=1,ncol=ni1)  rm(ds1)  rm(ds2)  #ランダムなデータセットを作る。  #初めの一組  j=1  o<-sample(1:nc1,1)  ds1<-dr1[o,]  i=2  while(i<=ns){  o<-sample(1:nc1,1)  ds1<-rbind(ds1,dr1[o,])  i<-i+1  }  ds2<-ds1  #繰り返して横につなぐ  j=2  for(j in 2:ni1){  o<-sample(1:nc1,1)  ds1<-dr1[o,]  i=2  while(i<=ns){  o<-sample(1:nc1,1)  ds1<-rbind(ds1,dr1[o,])  i<-i+1  }  ds2<-cbind(ds2,ds1)  j<-j+1  }  ds2  #correspondenc analysisの実施  ca<-corresp(ds2,nf=ni1)  ca$eig<-ca$cor^2  round(ca$eig,3)  Mateig<-t(ca$eig)  Mateig0<-Mateig  #以下繰り返し  k<-1  while(k<=99){  j=1  o<-sample(1:nc1,1)  ds1<-dr1[o,]  i=2  while(i<=ns){  o<-sample(1:nc1,1)  ds1<-rbind(ds1,dr1[o,])  i<-i+1  }  ds2<-ds1  #繰り返して横につなぐ２と  j=2  for(j in 2:ni1){  o<-sample(1:nc1,1)  ds1<-dr1[o,]  i=2  while(i<=ns){  o<-sample(1:nc1,1)  ds1<-rbind(ds1,dr1[o,])  i<-i+1  }  ds2<-cbind(ds2,ds1)  j<-j+1  }  #correspondenc analysisの実施  ca<-corresp(ds2,nf=ni1)  ca$eig<-ca$cor^2  round(ca$eig,3)  Mateig<-t(ca$eig)  Mateig0<-rbind(Mateig0,Mateig)  rm(Mateig)  k<-k+1  }  Mateig0  write.table(Mateig0,"Capara.csv",sep=",")  rm(Mateig0)  #選択型回答の平行分析correspondence analysis  library(MASS)  library(ggplot2)  library(FactoMineR)  library(psych)  #準備  rm(Mateig0)  rm(para)  inum<-20  Mateig0<-matrix(nrow=1,ncol=inum)  j=1  #O1データのランダムなdatasetを作る。  #j=1として、固有値の集計表の第一列を作る。  #シミュレーションの元になる、布全ての乱数表を導入  #データセットのitemsu数をni,組み合わせの数をnc,必要なセット数をnsとする。  #第一セット  dr1<-item9  ni1<-9  nc1=84  ns<-99  #第二セット  dr2<-item10  ni2<-10  nc2<-120  #第三セット  dr3<-item10  ni3<-10  nc3<-120  n<-ni1+ni2+ni3  Mateig0<-matrix(nrow=1,ncol=n)  #第一セット  o<-sample(1:nc1,1)  ds1<-dr1[o,]  i=2  while(i<=ns){  o<-sample(1:nc1,1)  ds1<-rbind(ds1,dr1[o,])  i<-i+1  }  #第二セット  i=1  o<-sample(1:nc2,1)  ds2<-dr2[o,]  i=2  while(i<=ns){  o<-sample(1:nc2,1)  ds2<-rbind(ds2,dr2[o,])  i<-i+1  }  #第三セット  i=1  o<-sample(1:nc3,1)  ds3<-dr3[o,]  i=2  while(i<=ns){  o<-sample(1:nc3,1)  ds3<-rbind(ds3,dr2[o,])  i<-i+1  }  #乱数表の結合乱数  rand<-cbind(ds1,ds2,ds3)  write.table(rand,"count4.csv",sep=",")  rand  #dorrespondence analysisの実施  ca<-corresp(rand,nf=inum)  ca$eig<-ca$cor^2  round(ca$eig,3)  Mateig<-t(ca$eig)  Mateig0<-Mateig  Mateig0  j<-2  #以下繰り返し計算  j=1  for(j in 1:100){  para<-matrix(nrow=ns,ncol=1)  #data数（ns）の乱数セットを変数個（inum)個発生させてdata.frameをつくる。  #第一セット  o<-sample(1:nc1,1)  ds1<-dr1[o,]  i=2  while(i<=ns){  o<-sample(1:nc1,1)  ds1<-rbind(ds1,dr1[o,])  i<-i+1  }    #第二セット  i=1  o<-sample(1:nc2,1)  ds2<-dr2[o,]  i=2  while(i<=ns){  o<-sample(1:nc2,1)  ds2<-rbind(ds2,dr2[o,])  i<-i+1  }  #第三セット  i=1  o<-sample(1:nc3,1)  ds3<-dr3[o,]  i=2  while(i<=ns){  o<-sample(1:nc3,1)  ds3<-rbind(ds3,dr2[o,])  i<-i+1  }  #乱数表の結合乱数  rand<-cbind(ds1,ds2,ds3)  write.table(rand,"count4.csv",sep=",")  rand    #correpondence anlysisの実施  ca<-corresp(rand,nf=inum)  ca$eig<-ca$cor^2  round(ca$eig,3)  Mateig<-t(ca$eig)  Mateig0<-rbind(Mateig0,Mateig)  rm(para)  rm(Mateig)  j<-j+1  }  Mateig0  Mateig0<-Mateig0[-1,]  write.table(Mateig0,"PinpricaPara.csv",sep=",")  rm(Mateig0)  warning() |