因子分析

注釈の一部が文字化けしています。折を見て修正します。Sample dataも見つかり次第掲載します。

|  |
| --- |
| #matirixとして作られたデータシートから、因子分析をします。データは量的なデータだけでなく、順位やyes-noの0－1データも#分析できるようにしました。#通常のPsychでは、順位や０－１データだｔ、回転が出来ませんが、ここではGPArotationをinstallして、それを可能にしています。#はじめに、平行分析をして、妥当な因子数を決めますが。おそらく、デフォルトでは、正規分布する数値をランダムに与えて、#項目数と#データ数が同じデータセットを何組かつくって、シミレーションして、確率的に、偶発的に発生する固有値の分布と比較し#ているはずです。#これは、順位やO－1の時のランダムな固有値の発生の分布とは異なるはずです。ですから、（（、Graphのスクリプト#に示した方法で、ランダムなデータ分布を作り、主成分分析によって、有意なデータ分布のデ＜（。#必要なパッケージのインストール#必要なパッケージinstall(MASS)install(ggplot2)install(pshych)install(GPArotation)# 開始library(MASS)library(ggplot2)library(psych)library(GPArotation)#データ導入。つづいて平行分析（主成分と因子について、各主成分・各因子とシミュレーションの結果が、#固有値の折れ線グラフで示される。fdata<-Poloprilogitfan<-fa.parallel(fdata)#上記の結果を参考に因子数を決め、因子分析を行うが、因子数（nfactor)を6以上にしないと、うまく動かない。#また、計量的なデータでない場合には,上記の結果は参考にならない。#因子数を決めるnf<-5#因子分析。抽出法「最尤法」"ml"、「最小二乗法」"minres"fa1<-fa(r=fdata,nfactor=nf,rotate="promax",fm="minres")summary(fa1)#各因子の因子付加量の表示print(fa1,score=TRUE)eig<-as.matrix.data.frame(fa1$loading)eig<-eig^2SSeig<-colSums(eig)SSeigpar(mfrow=c(2,2))barplot(fa1$loadings[,1])barplot(fa1$loadings[,2])barplot(fa1$loadings[,3])barplot(fa1$loadings[,4])barplot(fa1$loadings[,5])biplot(fa1)#各因子の固有値(分散)SSeig#各項目の因子負荷量fa1$loadings#各データの因子得点fa1$scores#各項目の因子負荷量の散布行列pairs(fa1$loadings)#各でデータの因子得点の散布行列pairs(fa1$scores)write.table(fa1$loadings,"Polopritem5.csv",sep=",")write.table(fa1$scores,"Polopriind5.csv",sep=",")write.table(SSeig,"Poloprieig5.csv",sep=",")#この結果を因子負荷量と因子得点について、重ね合わせた図をつくる。#このスクリプトのデフォルトでは、第一因子と第二因子でプロット。#df1とdf2の呼び込みの数字を変えれば、他の因子の関係も示せる#ただし、グラフのスクリプトで、各項目のベクトルと比較した方が、わかりやすいかもしれない。内容は同じ。fa1\_var\_df<-fa1$loadingsfa1\_obs\_df<-fa1$scoresdf1<-data.frame(x=fa1\_var\_df[,1],y=fa1\_var\_df[,2])df2<-data.frame(x=fa1\_obs\_df[,1],y=fa1\_obs\_df[,2])df3<-data.frame(x=fa1\_var\_df[,1],y=fa1\_var\_df[,2],z=rownames(fa1\_var\_df))#まず、ggplot2で、測定項目の因子負荷用の分布を示す。g<-ggplot(NULL)g<-g+geom\_segment(data=df1,aes(x=0,xend=x,y=0,yend=y,color="red"))+geom\_point(data=df2,aes(x,y),colour="black")+ geom\_text(df3,aes(x=x,y=y,label=z),color ="red",size =3,hjust=0)print(g)#荳蠢懊√√％縺薙ｌ縺ｧ逶ｮ逧?縺ｯ驕疲?舌＆繧後ｋ縺後∬ｵ､縺ｮ繝峨ャ繝医↓繝ｩ繝吶Ν繧剃ｻ倥◆縺ｻ縺?縺後ｏ縺九ｊ繧?縺吶＞縲ゅΛ繝吶Ν繧偵▽縺代ｋ縺ｨ縺阪↓豕ｨ諢上＠縺ｪ縺代ｌ縺ｰ縺ｪ繧峨↑縺?縺ｮ縺ｯ縲√ョ繝ｼ繧ｿ繝輔Ξ繝ｼ繝?縺ｮ蠖｢縺御ｸ閾ｴ縺励※縺?縺ｪ縺?縺ｨ驥阪?ｭ譖ｸ縺阪′縺ｧ縺阪↑縺?縺薙→縺ｧ縺ゅｋ縲ゅ◎縺薙〒縲√Λ繝吶Ν繧偵▽縺代↑縺?隕ｳ貂ｬ繝?繝ｼ繧ｿ縺ｮ譁ｹ縺ｮ繝輔Ξ繝ｼ繝?縺ｫ縺ｯ縲??ｽ?="null"繧貞?･繧後※縺翫￥縲Ｍayer縺ｨ縺?縺?讖溯?ｽ繧剃ｽｿ縺医?ｰ縲∝?励?ｮ謨ｰ縺御ｸ閾ｴ縺励※縺?縺ｪ縺上※繧ゅ°縺輔?ｭ縺後″縺後〒縺阪ｋ縺後√◎縺ｮ蝣ｴ蜷医∝峙縺ｮ菴咲ｽｮ繧?繧上￥縺ｮ縺翫♀縺阪＆繧偵＠縺ｦ縺?縺吶ｋ縺ｲ縺､繧医≧縺後≠繧翫ａ繧薙←縺?縺ｪ縺ｮ縺ｧ縲∝?励?ｮ謨ｰ繧剃ｸ閾ｴ縺輔○縺滓婿縺梧･ｽ縺ｧ縺ゅｋ縲?df3<-data.frame(x=fa1\_var\_df[,1],y=fa1\_var\_df[,2],z=rownames(fa1\_var\_df))df2<-data.frame(x=fa1\_obs\_df[,1],y=fa1\_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)#莉･荳九?ｯ繧ｰ繝ｩ繝輔?ｮ陬?鬟ｾ縲ゅりレ譎ｯ繧堤區縺ｫ縺吶ｋ縲?f<-f+theme\_bw()print(f)#繧ｰ繝ｩ繝輔ち繧､繝医Ν繧偵▽縺代ｋ縲?f<-f+ggtitle("factor relationship between FA1 and FA2")print(f)#邵ｦ霆ｸ讓ｪ霆ｸ縺ｮ隱ｬ譏?f<-f+xlab("FA1")+ylab("FA2")print(f)#蝗?蟄舌↓蜷?鬆?逶ｮ縺後←縺ｮ繧医≧縺ｫ縺九°繧上▲縺溘＞繧九?ｮ縺九ｒ遏･繧句ｿ?隕√′縺ゅｋ縺後∝屏蟄舌→縺ｯ縺､縺ｾ繧九→縺薙ｍ蜷?鬆?逶ｮ繧偵?吶け繝医Ν縺ｨ縺ｨ繧峨∴繧九→縲?#隍?謨ｰ縺ｮ鬆?逶ｮ縺ｮ繝吶け繝医Ν縺悟酔荳縺ｮ譁ｹ蜷代ｒ繧縺?縺ｦ縺?繧九→縺?縺?縺薙→縺ｧ縺ゅｋ縺昴ｌ縺槭ｌ縺ｮ雉ｪ蝠城??逶ｮ縺ｫ蟇ｾ縺吶ｋ蝗樒ｭ斐?ｮ繝吶け繝医Ν縺後?#蜷後§譁ｹ蜷代↓蜷代°縺｣縺ｦ縺ｾ縺ｨ縺ｾ縺｣縺ｦ縺?繧九°縺ｮ讀懆ｨ弱′蠢?隕√〒縺ゅｋ縲ゅ◎縺薙〒縲√け繝ｭ繝ｳ繝舌ャ繧ｯ縺ｮﾎｱ菫よ焚縺ｨﾏ峨→菫よ焚繧堤ｮ怜?ｺ縺吶ｋ縲?#縺ｩ縺ｮ鬆?逶ｮ繧帝∈繧薙〒ﾎｱ繧定ｨ育ｮ励☆繧九°縺ｯ縲?鬆?逶ｮ縺ｮ蝗?蟄仙ｾ礼せ縺ｧ驕ｸ縺ｶ縲?library(psych)alpha(money2)af1<-data.frame(chrom[,20],chrom[,22],chrom[,23],chrom[,25])af2<-data.frame(chrom[,27],chrom[,28],chrom[,29])af3<-data.frame(chrom[,15],chrom[,16],chrom[,17],chrom[,21])af4<-data.frame(chrom[,10],chrom[,12],chrom[,25])af5<-data.frame(chrom[,5],chrom[,6],chrom[,12],chrom[,17],chrom[25],chrom[26])af6<-data.frame(chrom[,18],chrom[,19])af7<-data.frame(chrom[,1],chrom[,12],chrom[,13],chrom[,21])af8<-data.frame(chrom[,2],chrom[,4],chrom[,10])alpha(af1)alpha(af2)alpha(af3)alpha(af4)alpha(af5)alpha(af6)alpha(af7)alpha(af8)#Mirtによるカテゴリカル因子分析。データをロジット変換して、MCMCで因子分析する。library(mirt)data<-pincate(mod1<-mirt(data,5))call:mirt(data=data, nfactor=5)coef(mod1)residuals(mod1,restype="exp")residuals(mod1)#カテゴリカルな順序説明変数を、本来連続的な変数を閾値によって切り分けた結果だと考える。#連続変数の分布を考えて、パラメータを最尤法的に決める。#これによって決まった相関係数を、ポリコリック相関係数うという。#サンプルデータの導入library(psych)library(polycor)individualdata<-data.frame(Polopsycat)ans<-hetcor(individualdata, ML=TRUE)ans$correlationsfa1<-factanal(individualdata,6,covmat=ans$correlations)eig<-as.matrix.data.frame(fa1$loading)eig<-eig^2SSeig<-colSums(eig)write.table(fa1$loadings,"Polopsycaitem6.csv",sep=",")write.table(SSeig,"Polopsycateig6.csv",sep=",")summary(fa1)#各因子の因子付加量の表示print(fa1,score=TRUE)eig<-as.matrix.data.frame(fa1$loading)eig<-eig^2SSeig<-colSums(eig)SSeigpar(mfrow=c(2,2))barplot(fa1$loadings[,1])barplot(fa1$loadings[,2])barplot(fa1$loadings[,3])barplot(fa1$loadings[,4])biplot(fa1)#各因子の固有値(分散)SSeig#各項目の因子負荷量fa1$loadings#各データの因子得点fa1$scores#各項目の因子負荷量の散布行列pairs(fa1$loadings)#各でデータの因子得点の散布行列pairs(fa1$scores)write.table(fa1$loadings,"Pinpricaload.csv",sep=",")write.table(fa1$scores,"Pinpricaind.csv",sep=",")write.table(SSeig,"Pinrricaeig.csv",sep=",") |