xyz <- matrix(0,12,5);

xyz[1,1]<-5700; xyz[1,2]<-12.8; xyz[1,3]<-2500;

xyz[1,4]<-270; xyz[1,5]<-25000;

xyz[2,1]<-1000; xyz[2,2]<-10.9; xyz[2,3]<-600;

xyz[2,4]<-10; xyz[2,5]<-10000;

xyz[3,1]<-3400; xyz[3,2]<-8.8; xyz[3,3]<-1000;

xyz[3,4]<-10; xyz[3,5]<-9000;

xyz[4,1]<-3800; xyz[4,2]<-13.6; xyz[4,3]<-1700;

xyz[4,4]<-140; xyz[4,5]<-25000;

xyz[5,1]<-4000; xyz[5,2]<-12.8; xyz[5,3]<-1600;

xyz[5,4]<-140; xyz[5,5]<-25000;

xyz[6,1]<-8200; xyz[6,2]<-8.3; xyz[6,3]<-2600;

xyz[6,4]<-60; xyz[6,5]<-12000;

xyz[7,1]<-1200; xyz[7,2]<-11.4; xyz[7,3]<-400;

xyz[7,4]<-10; xyz[7,5]<-16000;

xyz[8,1]<-9100; xyz[8,2]<-11.5; xyz[8,3]<-3300;

xyz[8,4]<-60; xyz[8,5]<-14000;

xyz[9,1]<-9900; xyz[9,2]<-12.5; xyz[9,3]<-3400;

xyz[9,4]<-180; xyz[9,5]<-18000;

xyz[10,1]<-9600; xyz[10,2]<-13.7; xyz[10,3]<-3600;

xyz[10,4]<-390; xyz[10,5]<-25000;

xyz[11,1]<-9600; xyz[11,2]<-9.6; xyz[11,3]<-3300;

xyz[11,4]<-80; xyz[11,5]<-12000;

xyz[12,1]<-9400; xyz[12,2]<-11.4; xyz[12,3]<-4000;

xyz[12,4]<-100; xyz[12,5]<-13000;

xyzz <- as.data.frame(xyz)

xyzz<-as.data.frame(read.csv("c:/Users/sigur/Documents/perfzk.csv"));

f <-5;

noitn<-50;

deltar<-0.0001;

numbcolsatrightpartialled<-42;

factoranalysis<-function(xyz, f, deltar, noitn,

numbcolsatrightpartialled){

#Program name is factoranalysis. Full facilities for factor analysing #data by a new method. Copyright (C) 2022 Dr Sigurd Johnsen This program is #free software. You can redistribute it and/or modify it under the terms of the #GNU General Public License as published by the Free Software Foundation. #Either version 2 of the License, or (at your option) any later version. This #program is distributed in the hope that it will be useful, but WITHOUT ANY #WARRANTY. Without even the implied warranty of MERCHANTABILITY or #FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for #more details. You should have received a copy of the GNU General Public License #along with this program. If not, write to the Free Software Foundation, Inc., 51 #Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA. #sigurdjohnsen7@gmail.com 6 Byron Avenue, Winchester, SO22 5AT, UK;

# The function needs the input data matrix, xyz, to be a data frame object;

# columns correspond to variables, rows to vectors of observations;

# dichotomous variables can be included in input data;

# f is number of factors;

# deltar (the differentiation step) is usually 0.0001;

# noitn (number of estimation iterations) is usually about 100;

# the number of rightmost variables (columns) in the input data set

# to be partialled out is given by numbcolsatrightpartialled.

# nosamp is the number of jackknifing samples, 8 recommended

# to start. A big enough value is when results are stabilised

# for different random start seeds.

# lis[1] in output gives start & final values of objective

# function – noitn should be big enough to stabilise

# final value.

# lis[2] to lis[4] in output are factor patterns resp. for

# no rotation,varimax (orthogonal factors)

# & promax (oblique factors).

# lis[5] to lis[7] in output are the corresponding factor scores.

noitn<-abs(as.integer(noitn));

if (noitn==0){noitn<-1};

numbcolsatrightpartialled<-abs(as.integer(numbcolsatrightpartialled));

f<-abs(as.integer(f));

deltar<-abs(deltar);

if(deltar<0.0001){deltar<-0.0001};

n <- nrow(xyz)

nam <- names(xyz)

qq<-ncol(xyz)- numbcolsatrightpartialled;

if (qq<2){numbcolsatrightpartialled<-0};

if((ncol(xyz)- numbcolsatrightpartialled)<2){a<-0;b<-1;c<-b/a};

if(numbcolsatrightpartialled>0 &

numbcolsatrightpartialled<ncol(xyz)){

q<-ncol(xyz)- numbcolsatrightpartialled;

coo <- numbcolsatrightpartialled;

y <- matrix(0,n,q) ;

xyz2 <- matrix(0,n,coo)

for (k in 1:n ){

for (kk in 1:coo){xyz2[k,kk] <- xyz[k,kk+q]}}

for (kkx in 1:q){

a <- lm(xyz[,kkx] ~ xyz2,na.action=na.exclude)

y[,kkx] <- residuals(a)}

for (rex in 1:q){

d <- 0;

dd <- 0

ddd <- 0

for (hhj in 1:n){if (is.na(y[hhj,rex])==F){dd <- dd+y[hhj,rex];ddd<-ddd+1}}

if(ddd>0){d<-dd/ddd;}

for (hj in 1:n){if (is.na(y[hj,rex])==T){y[hj,rex]<-d}}}}else

{y<-matrix(0,n,ncol(xyz)); y<-xyz;q<-ncol(y)}

for(lzy in 1:q){

sumz<-0;sumzz<-0;

for (lzzy in 1:n){

sumz<-sumz+y[lzzy,lzy]

sumzz<-sumzz+y[lzzy,lzy]^2;};

me<-sumz/n;

sd<-sqrt((sumzz-me\*sumz)/(n-1));

for(lzzzy in 1:n){

y[lzzzy,lzy]<-(y[lzzzy,lzy]-me)/sd};};

kkkk<-0;

check<-matrix(0,100,10);

epsi<-matrix(0,f,f)

for (ira in 1:f){epsi[ira,ira]<-0.00001}

if(f >= q){f<-(q-1)};

if(f==0){f<-1};

ff<-f;

nam1<-rep(q);

for (gmh in 1:q){nam1[gmh]<-nam[gmh]};

tempp<-matrix(0,1,q);

discrep<-rep(0,2);

yyy<-matrix(0,n,q);

yyy <- y;

scores<-matrix(0,n,f);

nmiss<-0;

means<-rep(0,q);

for (uht in 1:q){means[uht] <- 0};

m1<-matrix(0,q,f);

m2<-m1;

mat1<-matrix(0,q,q);

for (re in 1:q){mat1[re,re]<-1};

prk2<-rep(runif(q\*(f+2)+nmiss+1),q\*(f+2)+nmiss+1);

prk3<-rep(1,q\*(f+2)+nmiss+1);

prk<-rep(1,q\*(f+2)+nmiss+1);

mat2<-matrix(0,q,q);

orthog1<-matrix(0,q,q);

orthog<-matrix(0,q-f,q);

orthog2<-matrix(0,q-f,q);

lg<-log(2\*3.1416);

obj<-0;

mov<-0;

obj1<-0;

ina<-0;

ra<-0;

prev = 0;

fff<-f;

countlimit<-4;

count <- -countlimit-1;

initia<-0;

initial<-0;

jsaa1 <- q\*f+nmiss+1;

jsaa <- q\*(f+1)+nmiss;

for(jsa in jsaa1:jsaa){prk2[jsa] <- 0};

jsaa1 <- q\*f+f+nmiss+1;

jsaa <- q\*(f+2)+nmiss;

for(jsa in jsaa1:jsaa){prk2[jsa] <- 0.03};

jsaa1 <- 1;

jsaa <- q\*f;

trystart<-1;

try<-1;

centralobj <- 0;

xi<-rep(0, q\*(f+2)+nmiss+1);

g<-rep(0, q\*(f+2)+nmiss+1);

h<-rep(1, q\*(f+2)+nmiss+1);

sta<-0;

finalobj<-0;

for(it in 1:noitn){

count<-count+1;

if(count>countlimit+1){count<- -countlimit;};

mini<-1;

if (count<1){ novar<-q\*f+q;} else {novar<- q+nmiss+1;};

lim<-2+4\*novar+2;

indi<-0;

if(count == -countlimit || count == 1){

indi <- 1; conv <- 0;prev = -1000; xxx<-0; prk4<-prk2;

base<-0; centralobj<-0;};

if(indi == 1){

lim <- 1+2\*novar;}

indcm<-0;

if(count == -countlimit+1 || count == 2){

indcm<-1;};

if (indcm > 0){

g <- -xi; h <- -xi;};

if(conv == 0){

indico<-0; cz<-0;

for (xx in 1:lim){

xxx<-xxx+1;

if(xxx == 1){prk<-prk2;};

if (xxx > 1 & xxx < 2\*novar+2 & indico == 1

& abs(count)>0 & abs(count-countlimit-1)>0){

prkti<-prkti+1;

if (prkti == 3){prkti<-1;};

if (prkti == 1){ sta<-sta + 1;

prk[sta]<- (prk[sta]+deltar);}else{

prk[sta]<-prk[sta]-2\*deltar};};

if (xx == 2 + 2 \* novar){

cz<-0; limyy<-10; con<-0; limyyy<-3;}else{

cz<-0; limyy<-1; con<-0; limyyy<-1;};

initialobj <- obj; base <- obj; rati <- 1;

for (xyy in 1:limyy){

fai<-0;

for (xyyy in 1:limyyy){

if(limyy>1 & fai == 0 & con == 0){

if(xyy == 1 & xyyy == 1){usemov<-0;

prk2<-prk4; prk3<-prk2;rati<-1;

cz<-0;chg<-0;goodmov<-0;prevmov<-0;

};prk<-prk3;

cz<-cz+1; if(cz>3){cz<-1;}

if(cz == 2 & con == 0){

s1<-1; s2<-q\*f+q; if(count>0){s1<-q\*f+q+1;

s2<-q\*(f+2)+nmiss+1;};

for (il in s1:s2){

prk[il]<-prk3[il]+xi[il]\*deltar;};};

if(cz == 3 & con == 0){

s1<-1; s2<-q\*f+q; if(count>0){s1<-q\*f+q+1;

s2<-q\*(f+2)+nmiss+1;};

for (il in s1:s2){

prk[il]<-prk3[il]-xi[il]\*deltar;};};};

indk <- 1;

if(count == -countlimit || count == 1 ||

xx < 2 || xx > 1+2\*novar){indk <- 0};

if(xxx > 1 & xxx < 2+2\*novar &

indico == 1){indk <- 0};

if(con > 0 || xx == 4+4\*novar ||

fai > 0){indk <- 1};

if(indk == 0){

jj<-q\*f+q;

logs <- **0**;

axjk <- abs(prk[q\*(f+**2**)+nmiss+**1**])+**0.0001**;

for (hhilx in 1:q){

jj<-jj + **1**;

ax <- (abs(prk[jj])+**1**)\*axjk;

logs<-logs+log(ax);

  mat1[hhilx, hhilx] <- ax;}

jj<-0;

for (jjj in 1:f){

for (jjjj in 1:q){

jj<-jj + 1;

m2[jjjj,jjj] <- prk[jj]

orthog1[jjjj,jjj] <- prk[jj]};};

orthog1<-mat1%\*%orthog1;

start<-2; try<-1;

aasz<-sqrt(c(t(orthog1[,1])%\*%orthog1[,1]));

orthog1[,1]<-orthog1[,1]/aasz;

for(iikx in start:q){

iiijxlim<-q;

if(try>q){try<-1};

   succe<-**0;**

**iiijx<-0;**

**if(iikx<f+1){iiijxlim<-1; succe<-1};**

while(iiijx<iiijxlim & (iiijxlim==1 || succe==0)){

iiijx<-iiijx+1;

**if (iikx > f){for(jj in 1:q){orthog1[jj,iikx]<-0};**

   orthog1[try,iikx]<-**1**};

mzzz<-iikx-1;

for (iiimx in 1:mzzz){

ask1<-0;

for (ixc in 1:q){

askl<-ask1+orthog1[ixc,iiimx]\*orthog1[ixc,iikx];};

orthog1[,iikx]<-orthog1[,

iikx]-orthog1[,iiimx]\*askl;}

aasz<-sqrt(c(t(orthog1[,

iikx])%\*% orthog1[,iikx]));

try<-try+**1**;

if (aasz > **0.000000001 || try==q || iikx<f+1){**

orthog1[,iikx]<-orthog1[,iikx]/aasz;

succe<-**1**}}};

for(iinx in (f+1):q){for (iikxb in 1:q){

orthog[(iinx-f),iikxb]<-t(orthog1[iikxb,iinx])};}

for (hgt in (q\*f+1):(q\*(f+1))){

means[(hgt-q\*f)] <- prk[hgt];};

  logs = -**2**\*logs;

  orthog2 <- (orthog%\*%mat1);

  mat2<-t(orthog2)%\*%orthog2;

  logdet <- logs;

  obj<-n\*(logdet+(q-f)\*lg);

for (gggjk in 1:n){

for(ik in 1:q){

tempp[1,ik]<-(y[gggjk,ik]-means[ik])};

  obj=obj+tempp%\*%mat2%\*%t(tempp)};};

                        if (xxx == **1){**

                initialobj <- obj;

                indico <- **0**;

if (count == -countlimit || count == 1 ||

xx > 1 + 2 \* novar){indico <- **1**;};

sta <- **0**; if (count > **0)** {sta = q\*f+q;};

prkti <- **0**;

if (xx == **1){**

centralobj <- obj;} else { indico <- **1**};};

if (xx == **2** \* novar + **2** & con == **0** & fai == **0**){

if (cz == **1){finalobj <- obj;if(discrep[1]==0){**

**discrep[1]<-obj};if(is.na(base)){base<-0;};**

**if(is.na(centralobj)){centralobj<-0;};**

**if(base>centralobj){base <- centralobj;}**

  centralobj <- obj;};

                if (cz == **2){**

                        highobj <- obj; };

                        if (cz == **3){**

                        lowobj <- obj;

                        deno <- (highobj+lowobj-**2**\*centralobj);

if(is.na(deno)){deno<-0};

                        if (deno == **0){**

       if (xyy > **1){** mini <- mini/**2**;}; deno <- deltar;};

prevmov<-mov;

       mov <- -((highobj-lowobj)\*deltar)/(**2**\*(deno))+0.00001;

if(is.na(mov)){mov<-0};

if (xyy==1){usemov<-mov};

chg <- (centralobj - base);

if(is.na(chg)){chg<- -0.0005};

#if (xyy==1){chg<- -1}

if((chg > -0.001) & (chg < 0 & xyy > 1)){con<-1};

if (chg>0){prk3 <- prk2;if(goodmov==0){if(xyy>1){

usemov<- -usemov}}else{usemov<-goodmov};

rati <- 2 \* rati;}else{

prk2 <- prk3;prk4<-prk3;base<-centralobj; usemov<-mov;

goodmov<-prevmov;}

s1 <- **1**; s2 <- q\*f+q;

if (count > **0){** s1 <- q\*f+q+**1**;

s2 <- q\*(f+**2**)+nmiss+**1**; };

for(klpp in s1:s2){ prk3[klpp] <-

prk3[klpp] + (xi[klpp]\*usemov)/rati;};};

if(kkkk<100

){kkkk<-kkkk+1; check[kkkk,1]<-centralobj; check[kkkk,2]<-base;

check[kkkk,3]<-xyy;check[kkkk,4]<-chg;

check[kkkk,5]<-usemov;check[kkkk,6]<-rati;

check[kkkk,7]<-prk2[2]}};};};

if (xxx > **1** & xxx < **2** \* novar + **2** & indico == **1**

& abs(count)>0 & abs(count-countlimit-1)>0){

if (prkti == **1){**obj1 <- obj; } else

{prk[sta] <- prk[sta] + deltar;

xi[sta] <- (obj1-obj)/(**2**\*deltar);};};

if (xx == **4** + **4** \* novar){

if (abs(prev-centralobj) < 0.001**){**conv <- **1**;};

prev = centralobj;

discrep[2]<-prev;

gg <-**0**;

dgg <- **0**;

s1 <- **1**; s2 <- q\*f+q; if (count > **0){**

s1 <- q\*f+q+**1**; s2 <- q\*(f+**2**)+nmiss+**1**;};

for (jjj in s1:s2){

gg <- gg + g[jjj]^**2**;

dgg <- dgg + (xi[jjj]+g[jjj])\*xi[jjj];};

gam <- (dgg+**0.0001**)/(gg+**0.0001**);

for (kjj in s1:s2){

g[kjj] <- -xi[kjj];

h[kjj] <- g[kjj] + gam \* h[kjj];

xi[kjj]<-h[kjj];}};};};};

for(pls in 1:f){

orthog1[,pls]<-m2[,pls]}

start<-2;

aasz<-sqrt(c(t(orthog1[,1])%\*%orthog1[,1]));

orthog1[,1]<-orthog1[,1]/aasz;

for(iikx in start:f){

**m1[,iikx-1]<-orthog1[,iikx-1];**

**mzzz<-iikx-1;**

for (iiimx in 1:mzzz){

ask1<-0;

for (ixc in 1:q){

askl<-ask1+orthog1[ixc,iiimx]\*orthog1[ixc,iikx];};

orthog1[,iikx]<-orthog1[,

iikx]-orthog1[,iiimx]\*askl;}

aasz<-sqrt(c(t(orthog1[,

iikx])%\*% orthog1[,iikx]));

orthog1[,iikx]<-orthog1[,iikx]/aasz;};

m1[,f]<-orthog1[,f];

m1<-m2;

m <- matrix(0,f,f);

epsilon<-matrix(0,f,f);

for (ii in 1:f){m[ii,ii] <- 1};

for (ii in 1:f){epsilon[ii,ii] <- 0.00001};

b <- matrix(0,f,f);

bb <- matrix(0,f,f);

lambda <- matrix(0,q,f);

d <- matrix(0,f,f);

c <- matrix(0,q,f);

a <- matrix(0,f,f);

ev <- matrix(0,f,f);

egv <- matrix(0,f,f);

lambda <- m1;

for (iii in 1:5){

for (iv in 1:f){

d[iv,iv] <- 0;

for (iiv in 1:q){

c[iiv,iv] <- c(m1[iiv,]%\*%m[,iv])^3;

d[iv,iv] <- d[iv,iv] + c(m1[iiv,]%\*%m[,iv])^2;};};

b <- m%\*%(t(lambda)%\*%(c-(lambda%\*%d)/q));

bb <- t(b)%\*%b;

temp <- svd(bb);

svdd <- temp$d

svdu <- temp$u

svdv <- temp$v

for(tsf in 1:f){egv[tsf,tsf] <- sqrt(svdd[tsf]);};

a <- svdv%\*%egv%\*%t(svdu);

m <- b%\*%solve(a+epsilon);

lambda <- m1%\*%m;};

ppx <- matrix(0,q,f);

for(jv in 1:f){

for(jjv in 1:q){

ppx[jjv,jv] <- ((abs(lambda[jjv,jv]))^3)\*lambda[jjv ,jv];};};

u <- matrix(0,f,f);

u <- solve((t(lambda)%\*%lambda)+epsilon)%\*%t(lambda)%\*%ppx;

dsq <- solve((t(u)%\*%u)+epsilon);

dd <- matrix(0,f,f);

for (lv in 1:f){dd[lv,lv] <- sqrt(dsq[lv,lv])};

lambdap <- lambda%\*%u%\*%dd;

store<-matrix(0,q,f);

for(yjt in 1:3){

store<-m1;

if(yjt==2){store<-lambda};

if(yjt==3){store<-lambdap};

for(yjtt in 1:f){

su<-0;

for(yjttt in 1:q){if(su<abs(store[yjttt,yjtt])){su<- abs(store[yjttt,yjtt])}}

store[,yjtt]<-store[,yjtt]/su;}

if(yjt==1){m1<-store};

if (yjt==2){lambda<-store};

if(yjt==3){lambdap<-store}}

scoresunrotated1 <- matrix(0,n,f);

scoresvarimax1 <- matrix(0,n,f);

scorespromax1 <- matrix(0,n,f);

matzv <- solve((t(lambda)%\*%mat1%\*%mat1%\*%lambda)+epsi)%\*%t(lambda)%\*%mat1;

matzp <- solve((t(lambdap)%\*%mat1%\*%mat1%\*%lambdap)+epsi)%\*%t(lambdap)%\*%mat1;

matzu <- solve((t(m1)%\*%mat1%\*%mat1%\*%m1)+epsi)%\*%t(m1)%\*%mat1;

for (ii in 1:n){

for (j in 1:q){tempp[1,j] <- yyy[ii,j]-means[j];};

temp <- t(matzu%\*%t(tempp));

for(jj in 1:f){scoresunrotated1[ii,jj] <- temp[jj]};};

for (ii in 1:n){

for (j in 1:q){tempp[1,j] <- yyy[ii,j]-means[j];};

temp <- t(matzv%\*%t(tempp));

for(jj in 1:f){scoresvarimax1[ii,jj] <- temp[jj]};};

for (ii in 1:n){

for (j in 1:q){tempp[1,j] <- yyy[ii,j]-means[j];};

temp <- t(matzp%\*%t(tempp));

for(jj in 1:f){scorespromax1[ii,jj] <- temp[jj]};};

store<-matrix(0,n,f);

for(yjt in 1:3){

store<-scoresunrotated1;

if(yjt==2){store<-scoresvarimax1};

if(yjt==3){store<-scorespromax1};

for(yjtt in 1:f){

su<-0;

for(yjttt in 1:n){if(is.na(store[yjttt,yjtt])==F){

{if(su<abs(store[yjttt,yjtt])){su<- abs(store[yjttt,yjtt])}}}};

store[,yjtt]<-store[,yjtt]/su;}

if(yjt==1){scoresunrotated1<-store};

if (yjt==2){scoresvarimax1<-store};

if(yjt==3){scorespromax1<-store}};

scoresunrotated<-as.data.frame(scoresunrotated1);

scoresvarimax<-as.data.frame(scoresvarimax1);

scorespromax<-as.data.frame(scorespromax1);

dimnames(m1) <- list(nam1,NULL);

dimnames(lambda) <- list(nam1,NULL);

dimnames(lambdap) <- list(nam1,NULL);

unrotatedfactorpattern<-as.data.frame(m1);

varimaxfactorpattern<-as.data.frame(lambda);

promaxfactorpattern<-as.data.frame(lambdap);

lis<-list(discrep,

unrotatedfactorpattern,

varimaxfactorpattern,promaxfactorpattern,

scoresunrotated,scoresvarimax,scorespromax);

return(lis)}

lis<-factoranalysis(xyzz, f, deltar, noitn, numbcolsatrightpartialled);

lis