Program “facan” for Factor Analysis
with TI Voyage 200 or TI 89 Titanium
Feb. 17th 2021

Factor Analysis is a method for detecting variables that are behind a set of correlated variables. Principal component analysis is a method for reducing the dimensionality of a set of variables.

facan(if1,if2), whereas
if1: Input for computation (0: Data matrix, 1: Correlation matrix)
if2: Method (0: Principal component analysis, 1: Factor analysis)

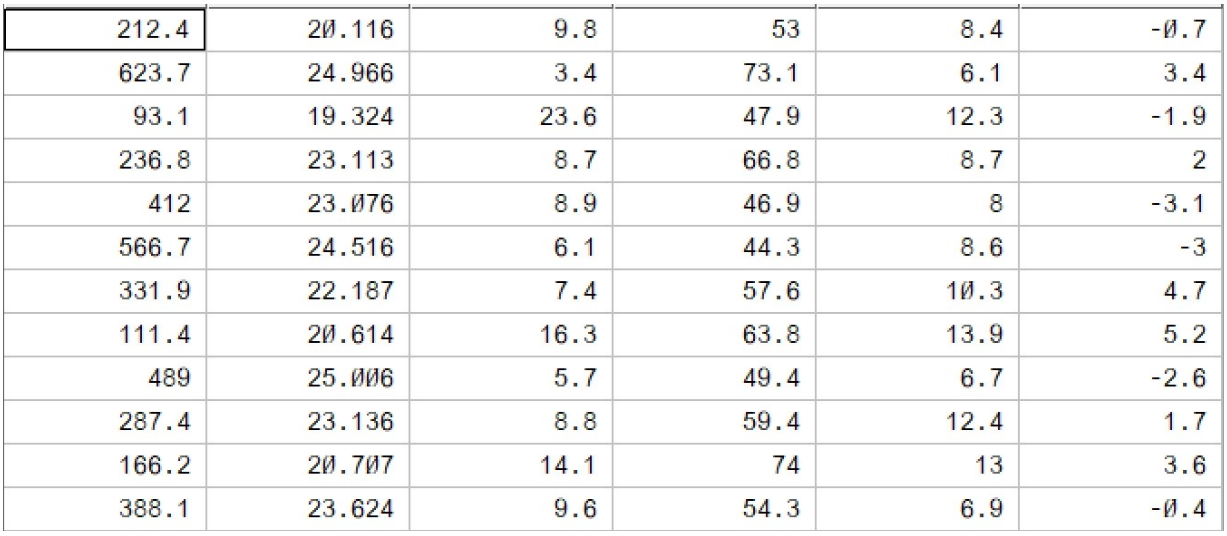
Precondition:
- Angle mode is set to “rad”
- If if1=0 Data matrix mdo (cases: lines, variables: columns) is in same directory as program or
If if1=1 Correlation matrix mr is in same directory as program

Variables:
a) Global variables
if1, if2: Control variables (Input, explanation see above)
mdo: Data matrix (input)
mds: Data matrix standardized (output, if if1=0)
mr: Correlation matrix (Input(if1=1) or output (if1=1)), later: Reduced Correlation matrix (output)
ma: Matrix A of unrotated factor loadings (output)
mar: Matrix of rotated factor loadings (output)
maa: Matrix AA’ of reproduced correlation matrix (output)
mrot: Rotation matrix (output)
mfs: Matrix of factor scores (lines: cases; columns: factors)
lews: List of sorted Eigenvalues

b) Local variables
i1..i3: Loop indices
zf: Number of cases
zv: Number of variables
zfa: Number of common factors (input during program run)
lew: List of eigenvalues (unsorted)
lix: List of order of eigenvalues
zf0,zb0,zc0,zd0,zn0,zz0,zu0,zv0,zt4,za4,za0,za1,zw0,zwi: Used for rotation
lve: List of variances of rotated factor loadings (unsorted)
lvs: List of sorted variances of factor loadings

Example:

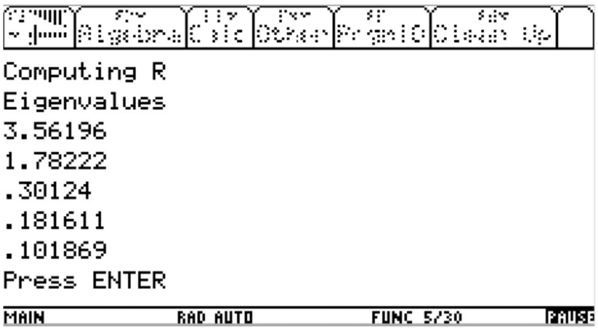
Data matrix mdo:



212.4	20.116	9.8	53	8.4	-0.7
623.7	24.966	3.4	73.1	6.1	3.4
93.1	19.324	23.6	47.9	12.3	-1.9
236.8	23.113	8.7	66.8	8.7	2
412	23.076	8.9	46.9	8	-3.1
566.7	24.516	6.1	44.3	8.6	-3
331.9	22.187	7.4	57.6	10.3	4.7
111.4	20.614	16.3	63.8	13.9	5.2
489	25.006	5.7	49.4	6.7	-2.6
287.4	23.136	8.8	59.4	12.4	1.7
166.2	20.707	14.1	74	13	3.6
388.1	23.624	9.6	54.3	6.9	-0.4

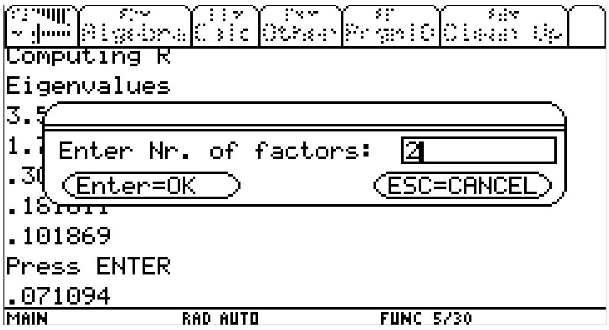
facan(0,1)- Enter

The first „0‟ means: Starting with date matrix (alternative: 1 = starting with correlation matrix), the second number “1” means, that we want to have a factor analysis (alternative: 0 = principal component analysis)



```
Computing R
Eigenvalues
3.56196
1.78222
.30124
.181611
.101869
Press ENTER
```

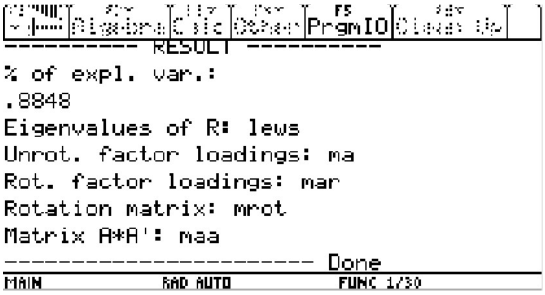
Pressing Enter shows the last eigenvalue and dialog box for entering of number of factors. Because of 2 big eigenvalues > 1 we chose 2.



```
Computing R
Eigenvalues
3.5
1.
.3
.18
.101869
Press ENTER
.071094
```

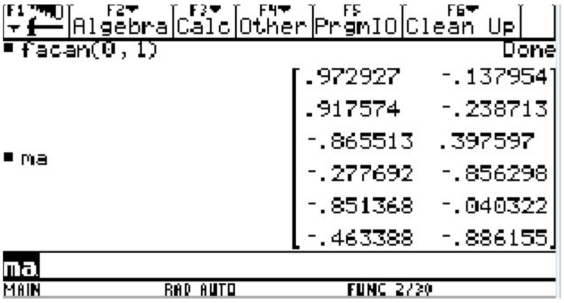
Enter Nr. of factors: 2

Then there is the message “Iterating communalities” and the development of the termination criterion is shown. If this is below 1E-4 there is the message “Varimax rotation”



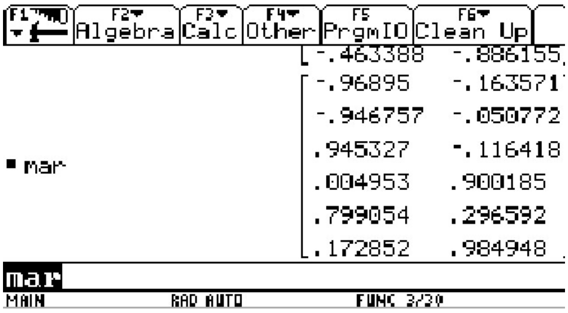
```
---------- RESULT ----------
% of expl. var.:
.8848
Eigenvalues of R: lews
Unrot. factor loadings: ma
Rot. factor loadings: mar
Rotation matrix: mrot
Matrix A*A': maa
----------------------- Done
```

Results are:
Matrix of unrotated factor loadings ma



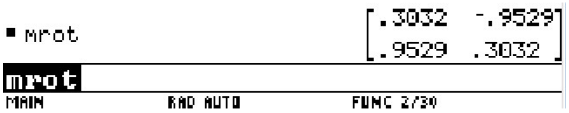
```
■ facan(0,1)                    Done
                  [.972927   -.137954]
                  [.917574   -.238713]
■ ma              [-.865513  .397597 ]
                  [-.277692  -.856298]
                  [-.851368  -.040322]
                  [-.463388  -.886155]
ma
```

Matrix of rotated factor loadings mar



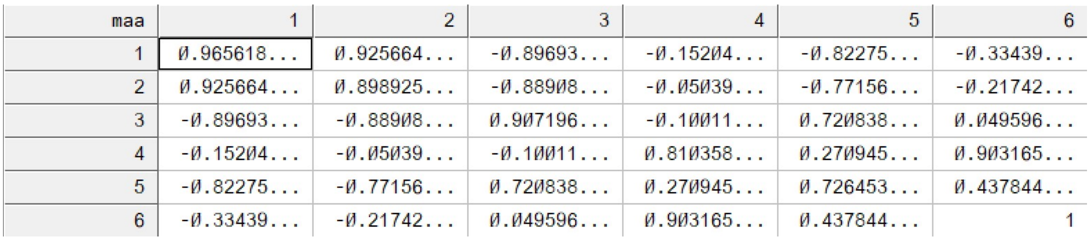
```
                  [-.463388  -.886155]
                  [-.96895   -.163571]
                  [-.946757  -.050772]
■ mar             [.945327   -.116418]
                  [.004953   .900185 ]
                  [.799054   .296592 ]
                  [.172852   .984948 ]
mar
```

Rotation matrix mrot



```
■ mrot            [.3032  -.9529]
                  [.9529  .3032 ]
mrot
```

Reproduced correlation matrix maa:



maa	1	2	3	4	5	6
1	0.965618...	0.925664...	-0.89693...	-0.15204...	-0.82275...	-0.33439...
2	0.925664...	0.898925...	-0.88908...	-0.05039...	-0.77156...	-0.21742...
3	-0.89693...	-0.88908...	0.907196...	-0.10011...	0.720838...	0.049596...
4	-0.15204...	-0.05039...	-0.10011...	0.810358...	0.270945...	0.903165...
5	-0.82275...	-0.77156...	0.720838...	0.270945...	0.726453...	0.437844...
6	-0.33439...	-0.21742...	0.049596...	0.903165...	0.437844...	1

Program listing:

```
facan(if1,if2)
Prgm
Local i1,i2,i3,lmw,lsd,zf,zfa,zv,zdim
Local lew,lix,mev,zak,zf0,zb0,zc0,zd0
Local zn0,zz0,zu0,zv0,zt4,za4,za0
Local za1,zw0,zw1,zw2,lve,lvs,lvi
If if1=1
 Goto korr
Disp "Computing R"
dim(mdo)->zdim
zdim[1]->zf
zdim[2]->zv
newList(zv)->lmw
newList(zv)->lsd
newMat(zf,zv)->mds
For i1,1,zv
 For i2,1,zf
  lmw[i1]+mdo[i2,i1]->lmw[i1]
  lsd[i1]+mdo[i2,i1]^2->lsd[i1]
 EndFor
 lmw[i1]/zf->lmw[i1]
 SQRT((lsd[i1]-zf*lmw[i1]^2)/(zf-1))->lsd[i1]
EndFor
For i1,1,zv
 For i2,1,zf
  (mdo[i2,i1]-lmw[i1])/(lsd[i1])->mds[i2,i1]
 EndFor
EndFor
mds™*mds/(zf-1.)->mr
Lbl korr
eigVl(mr)->lew
dim(lew)->zv
lew->lews
SortD lews
Disp "Eigenvalues"
For i1,1,zv
 Disp lews[i1]
 If mod(i1,5)=0 Then
  Disp "Press ENTER"
  Pause
 EndIf
```

```
EndFor
Request "Nr. of factors?",zfa
expr(zfa)->zfa
Disp "Iterating communalities"
newList(zv)->lix
newMat(zv,zfa)->ma
1->zak
While zak>1.•4
 eigVc(mr)->mev
 For i1,1,zfa
  For i2,1,zv
   If lews[i1]=lew[i2] Then
    i2->lix[i1]
   EndIf
  EndFor
 EndFor
 For i1,1,zfa
  For i2,1,zv
   mev[i2,lix[i1]]*SQRT(lews[i1])->ma[i2,i1]
  EndFor
 EndFor
 ma*ma™->maa
 For i1,1,zv
  If maa[i1,i1]>1 Then
   For i2,1,zfa
   ma[i1,i2]/(SQRT(maa[i1,i1]))->ma[i1,i2]
   EndFor
   1->maa[i1,i1]
  EndIf
 EndFor
 0->zak
 For i1,1,zv
  zak+(mr[i1,i1]-maa[i1,i1])^2->zak
  maa[i1,i1]->mr[i1,i1]
 EndFor
 SQRT(zak)->zak
 If if2=0 Then
  0->zak
 EndIf
 Disp zak
EndWhile
If zfa=1
 Goto fin
Disp "Varimax rotation"
0->zw0
1->zak
ma->mar
While zak>1.•4
 For i1,1,zfa-1
  For i2,i1+1,zfa
   0->zf0
   0->zb0
   0->zc0
   0->zd0
   For i3,1,zv
    (mar[i3,i1]^(2.)-mar[i3,i2]^(2.))/(mr[i3,i3])->zu0
    2.*mar[i3,i1]*mar[i3,i2]/(mr[i3,i3])->zv0
    zf0+zu0->zf0
    zb0+zv0->zb0
    zc0+zu0^2-zv0^2->zc0
    zd0+2.*zu0*zv0->zd0
   EndFor
   zd0-2.*zf0*zb0/zv->zz0
   zc0-(zf0^2-zb0^2)/zv->zn0
   If zn0=0 Then
    100.->zt4
   Else
    zz0/zn0->zt4
   EndIf
   tan´(zt4)->za4
   0->za0
   If zz0<0
    za0+Œ->za0
   If zt4<0
    za0+Œ->za0
   (za4+za0)/(4.)->za1
   For i3,1,zv
    mar[i3,i1]*cos(za1)+mar[i3,i2]*sin(za1)->zw2
    mar[i3,i1]*sin(za1)+mar[i3,i2]*cos(za1)->mar[i3,i2]
    zw2->mar[i3,i1]
   EndFor
  EndFor
 EndFor
 0->zf0
 0->zb0
 For i1,1,zv
  For i2,1,zfa
   zf0+mar[i1,i2]^2/(mr[i1,i1])->zf0
   zb0+(mar[i1,i2]^2/(mr[i1,i1]))^2->zb0
  EndFor
 EndFor
 (zb0-zf0^2/(zv*zfa))/(zv*zfa-1)->zw1
 Disp zw1
 zw1-zw0->zak
 If zfa=2
  0->zak
 zw1->zw0
EndWhile
© Computing rotation matrix
(ma™*ma)^(1)*ma™*mar->mrot
Lbl fin
Disp "Final computations"
© Sum of factor loadings per factor> 0
For i1,1,zfa
```

```
 0->zw0
 For i2,1,zv
  zw0+mar[i2,i1]->zw0
 EndFor
 If zw0<0 Then
  For i2,1,zv
   mar[i2,i1]->mar[i2,i1]
  EndFor
 EndIf
EndFor
© Ordering fact. accord. to var. expl.
newList(zfa)->lve
newList(zfa)->lvi
For i1,1,zfa
 For i2,1,zv
  lve[i1]+mar[i2,i1]^2->lve[i1]
 EndFor
EndFor
lve->lves
SortD lves
For i1,1,zfa
 For i2,1,zfa
  If lves[i1]=lve[i2] Then
   i2->lvi[i1]
  EndIf
 EndFor
EndFor
For i1,1,zv
 For i2,1,zfa
  mar[i1,i2]->lve[i2]
 EndFor
 For i2,1,zfa
  lve[lvi[i2]]->mar[i1,i2]
 EndFor
 1->mr[i1,i1]
EndFor
If if1=1
 Goto result
© Computing factor scores
(mar™*mr^(1)*mds™)™->mfs
Lbl result
Disp "---------- RESULT ----------"
Disp "% of expl. var.:",sum(lves)/zv*100
Disp "Eigenvalues of R: lews"
Disp "Unrot. factor loadings: ma"
Disp "Rot. factor loadings: mar"
Disp "Rotation matrix: mrot"
Disp "Matrix A*A': maa"
If if1=0
 Disp "Matrix of factor scores: mfs"
Disp "----------------------- Done"
```