Practical: Phenotypic Factor Analysis

Big 5 dimensions Neuroticism & Extraversion in 361 female UvA students

- Exploratory Factor Analysis (**EFA**) using R (factanal) with Varimax and Promax rotation
- Confirmatory Factor Analysis (**CFA**) using OpenMx

Neuroticism: identifies individuals who are prone to psychological distress

n1 - Anxiety: level of free floating anxiety

n2 - Angry Hostility: tendency to experience anger and related states such as frustration and bitterness

n3 - Depression: tendency to experience feelings of guilt, sadness, despondency and loneliness

n4 - Self-Consciousness: shyness or social anxiety

n5 - Impulsiveness: tendency to act on cravings and urges rather than delaying gratification

n6 - Vulnerability: general susceptibility to stress

Extraversion: quantity and intensity of energy directed outwards into the social world

e1 - Warmth: interest in and friendliness towards others

e2 - Gregariousness: preference for the company of others

e3 - Assertiveness: social ascendancy and forcefulness of expression

e4 - Activity: pace of living

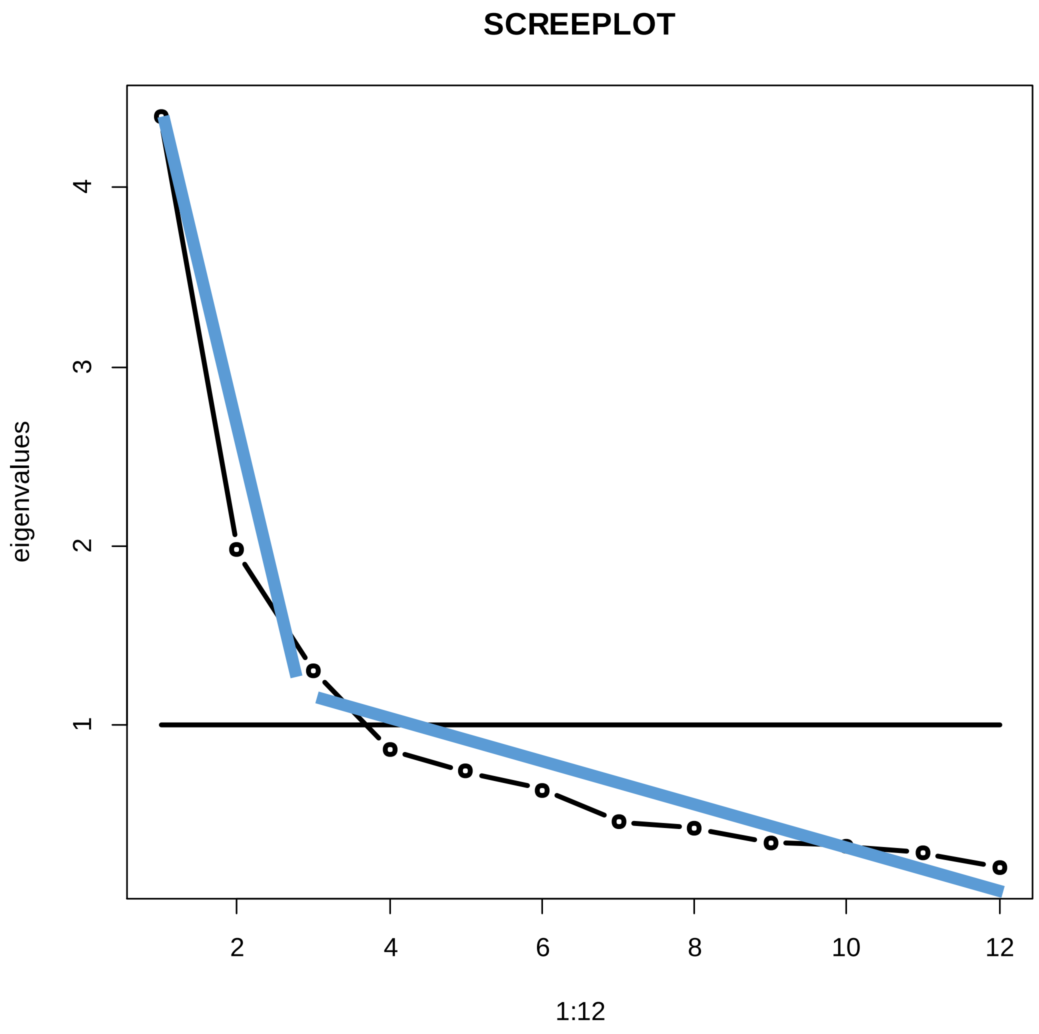
e5 - Excitement Seeking: need for environmental stimulation

e6 - Positive Emotions: tendency to experience positive emotions

```
        # Part 1: read the data

# clear the memory
rm(list=ls(all=TRUE))

# load OpenMx
library(OpenMx)

# set workingdirectory
setwd("YOUR_WORKING_DIRECTORY")

# read the data
datb5=read.table('rdataf')    # read the female data

# assign variable names
varlabs=c('sex',
 'n1', 'n2', 'n3', 'n4', 'n5', 'n6',
 'e1', 'e2', 'e3', 'e4', 'e5', 'e6',
 'o1', 'o2', 'o3', 'o4', 'o5', 'o6',
 'a1', 'a2', 'a3', 'a4', 'a5', 'a6',
 'c1', 'c2', 'c3', 'c4', 'c5', 'c6')
colnames(datb5)=varlabs

# select the variables of interest
isel=c(2:13)   # selection of variables n1-n6, e1-e6
datb2=datb5[,isel] # the data frame that we'll use below.
```

```
# Part 2: summary statistics

Ss1=cov(datb2[,1:12]) # calculate the covariance matrix in females
print(round(Ss1,1))

Rs1=cov2cor(Ss1)       # convert to correlation matrix
print(round(Rs1,2))

Ms1=apply(datb2[,1:12],2,mean)    # females means
print(round(Ms1,2))

# End of part 2
```

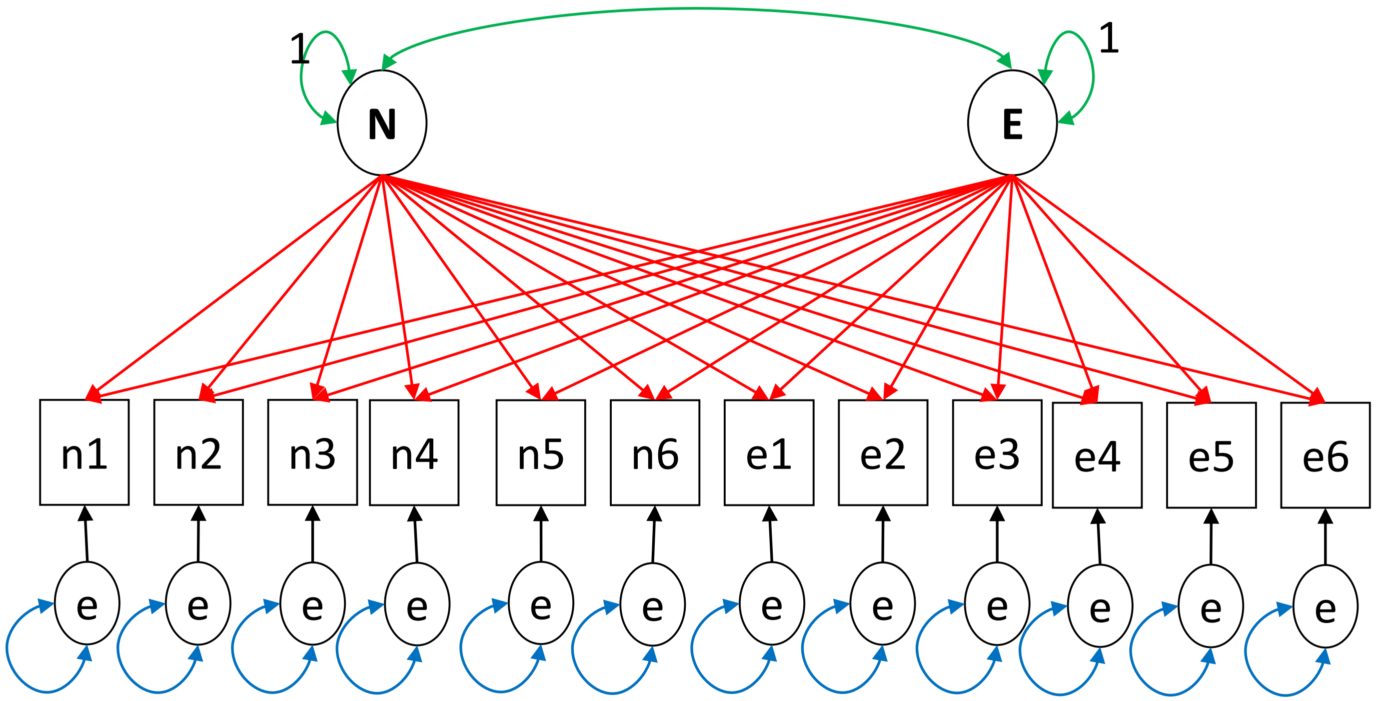
```
> print(round(Ms1,2))
   n1    n2    n3    n4    n5    n6    e1    e2    e3    e4    e5    e6 
23.22 20.38 23.35 23.99 27.62 20.00 31.23 28.81 23.36 25.43 27.78 31.13 
> print(round(Rs1,2))
      n1    n2    n3    n4   n5    n6    e1    e2    e3    e4    e5    e6
n1  1.00  0.44  0.77  0.59 0.18  0.72 -0.25 -0.20 -0.29 -0.24 -0.14 -0.40
n2  0.44  1.00  0.45  0.30 0.27  0.44 -0.36 -0.22  0.07 -0.03  0.02 -0.33
n3  0.77  0.45  1.00  0.62 0.20  0.69 -0.28 -0.24 -0.36 -0.29 -0.11 -0.47
n4  0.59  0.30  0.62  1.00 0.15  0.59 -0.34 -0.24 -0.42 -0.19 -0.11 -0.39
n5  0.18  0.27  0.20  0.15 1.00  0.22  0.06  0.15  0.06  0.13  0.37  0.19
n6  0.72  0.44  0.69  0.59 0.22  1.00 -0.28 -0.14 -0.35 -0.27 -0.08 -0.43
e1 -0.25 -0.36 -0.28 -0.34 0.06 -0.28  1.00  0.46  0.13  0.28  0.15  0.59
e2 -0.20 -0.22 -0.24 -0.24 0.15 -0.14  0.46  1.00  0.14  0.19  0.37  0.38
e3 -0.29  0.07 -0.36 -0.42 0.06 -0.35  0.13  0.14  1.00  0.38  0.13  0.26
e4 -0.24 -0.03 -0.29 -0.19 0.13 -0.27  0.28  0.19  0.38  1.00  0.27  0.45
e5 -0.14  0.02 -0.11 -0.11 0.37 -0.08  0.15  0.37  0.13  0.27  1.00  0.30
e6 -0.40 -0.33 -0.47 -0.39 0.19 -0.43  0.59  0.38  0.26  0.45  0.30  1.00
```



How many factors?

Ambiguous....
- Eigenvalues > 1 suggests 3 factors
- Elbow criterion suggests 2 factors

$$\Sigma_y = \Lambda \Psi \Lambda^t + \Theta$$

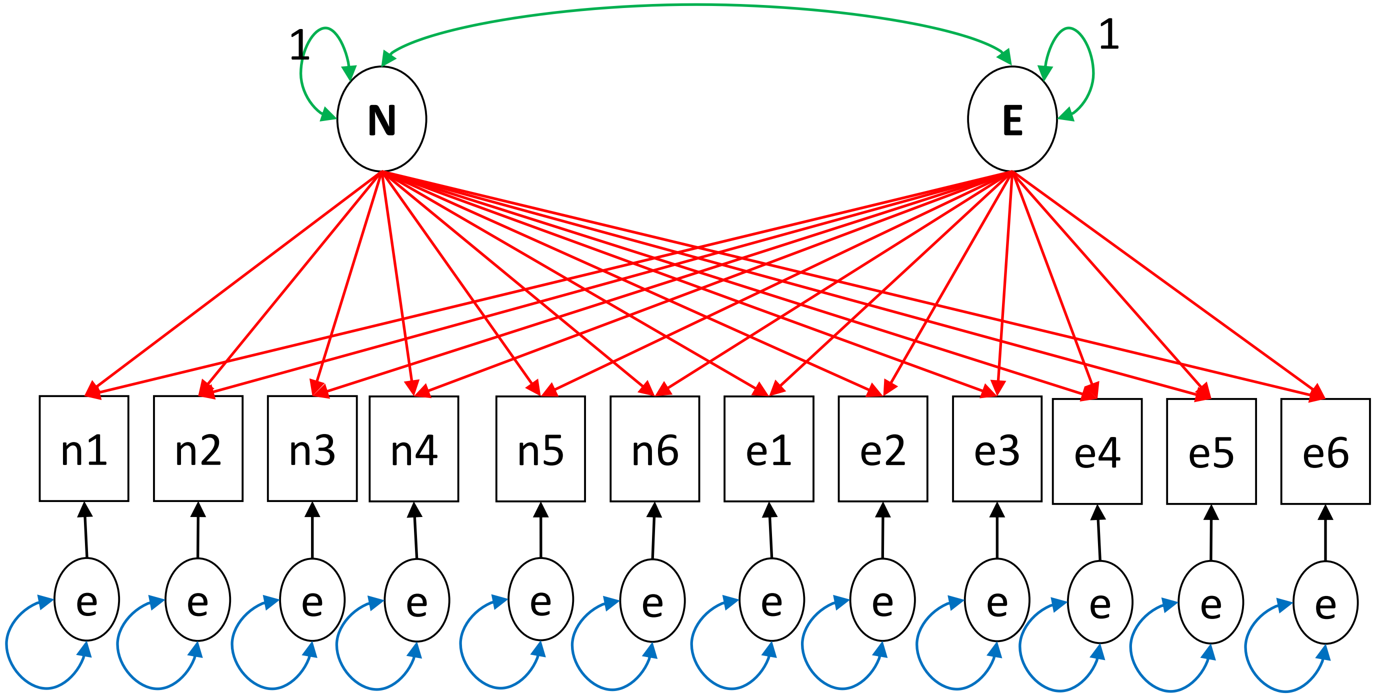


EFA

$$\Sigma_y = \Lambda \Psi \Lambda^t + \Theta$$

Matrix (12x2) of factor loadings Λ (note: |loading| < .1 not shown):

	Factor1	Factor2
n1	0.851	
n2	0.486	
n3	0.838	
n4	0.647	-0.140
n5	0.464	0.501
n6	0.801	
e1		0.614
e2		0.537
e3	-0.294	0.199
e4		0.466
e5	0.102	0.497
e6	-0.198	0.731



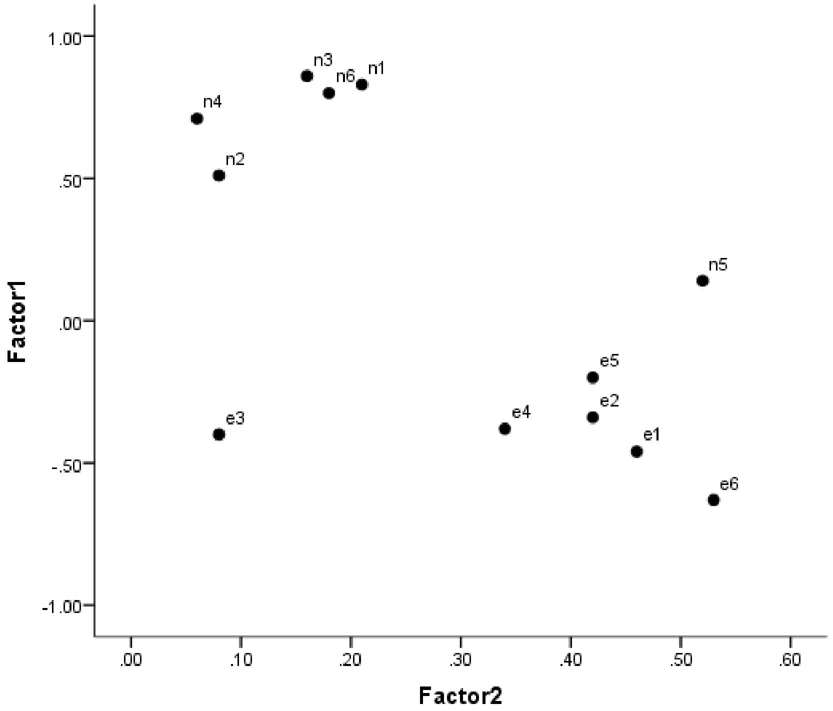
Factor Correlation matrix (2x2) Ψ:

	Factor1	Factor2
Factor1	1.000	-0.368
Factor2	-0.368	1.000

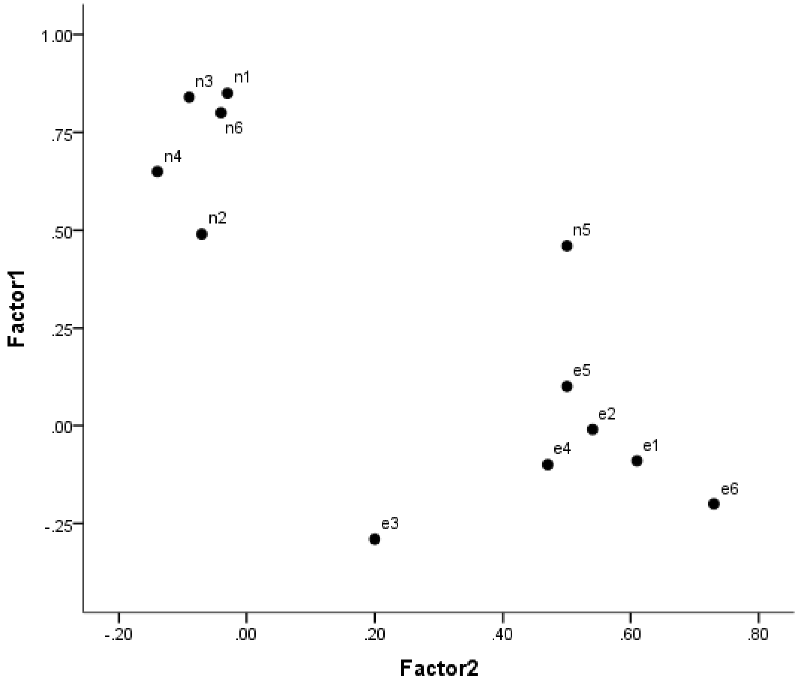
Diagonal covariance matrix (12x12) of residuals (Θ):

n1	n2	n3	n4	n5	n6	e1	e2	e3	e4	e5	e6
0.259	0.735	0.237	0.496	0.705	0.330	0.574	0.707	0.831	0.738	0.780	0.321

unrotated



rotated



Goodness of fit of EFA 2 factor model.

Test of the hypothesis that 2 factors are sufficient. The chi square statistic is 289.76 on 43 degrees of freedom. The p-value is **2.52e-38**

By this statistical criterion the model is judged to be acceptable if the p-value is greater than the chosen alpha (e.g. alpha=.05).

By the statistical criterion, we'd **reject** the model!

```
# Part 4A: saturated model

ny=12     # number of indicators
ne=2      # expected number of common factors
varnames=colnames(datb2) # var names

### fit the saturated model ###
```

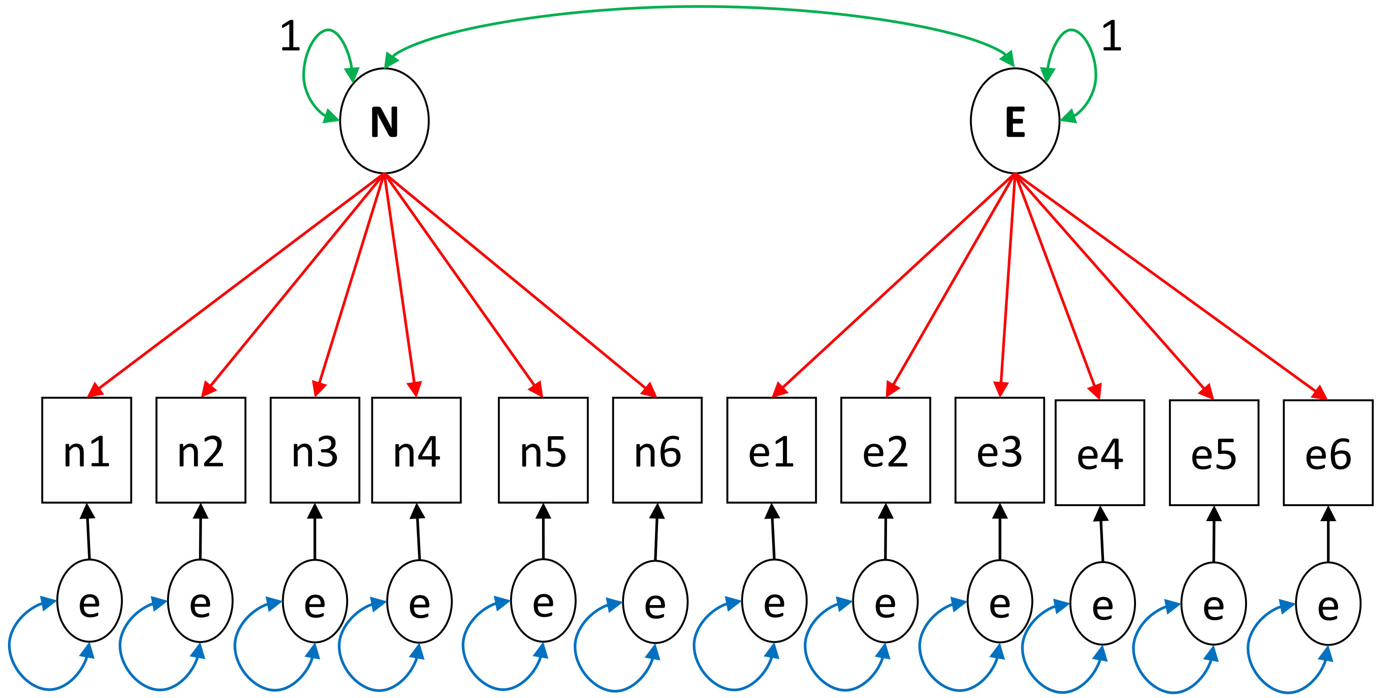
```
# define the means and covariance matrix in OpenMx to obtain the saturated model loglikelihood

Rs1=mxMatrix(type='Stand',nrow=ny,ncol=ny,free=TRUE,value=.05,
                          lbound=-.9,ubound=.9,name='cor')                    # 12x12 correlation matrix
Sds1=mxMatrix(type='Diag',nrow=ny,ncol=ny,free=TRUE,value=5,name='sds')       # 12x12 diagonal matrix (st devs)
Mean1=mxMatrix(type='Full',nrow=1,ncol=ny,free=TRUE,value=25,name='mean1')    # 1x12 vector means
MkS1=mxAlgebra(expression=sds%*%cor%*%sds,name='Ssat1')                       # expected covariance matrix

satmodels1=mxModel('part1',Rs1, Sds1, Mean1,MkS1) # assemble the model
```

```
# data + estimation function

satdats1=mxModel("part2",
   mxData( observed=datb2, type="raw"),   # the data
   mxExpectationNormal( covariance="part1.Ssat1", means="part1.mean1",
   dimnames=varnames),  # the fit function
   mxFitFunctionML()
                                     )  # data & expected cov/means
```

```
# fit the saturated model....

Models1 <-  mxModel("models1", satmodels1, satdats1,
                 mxAlgebra(part2.objective, name="minus2loglikelihood"),
                 mxFitFunctionAlgebra("minus2loglikelihood"))
Models1_out <- mxRun(Models1)
```

```
> summary(Models1_out)

observed statistics:  4332
estimated parameters:  90
degrees of freedom:  4242
-2 log likelihood:  23578.09
number of observations:  361
```

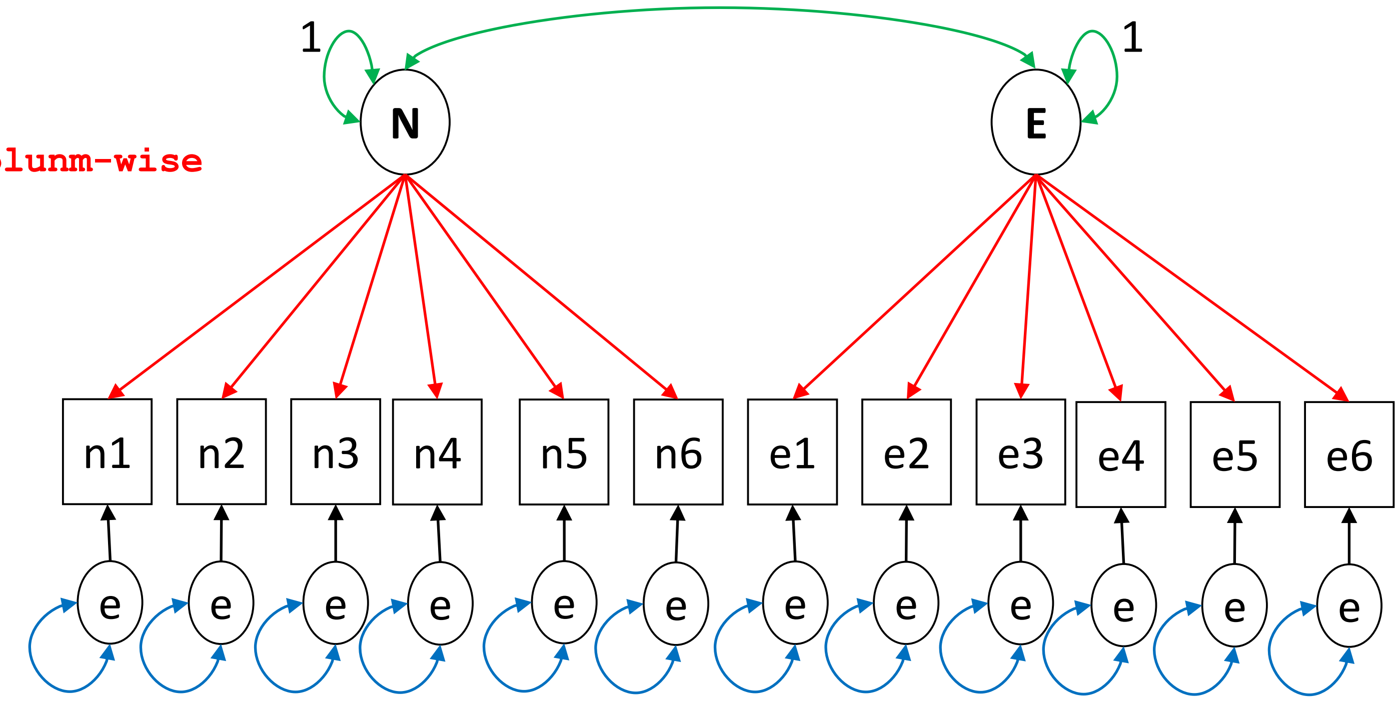
$$\Sigma_y = \Lambda \Psi \Lambda^t + \Theta$$



Define factor loading matrix Λ

$$\Sigma_y = \Lambda \Psi \Lambda^t + \Theta$$

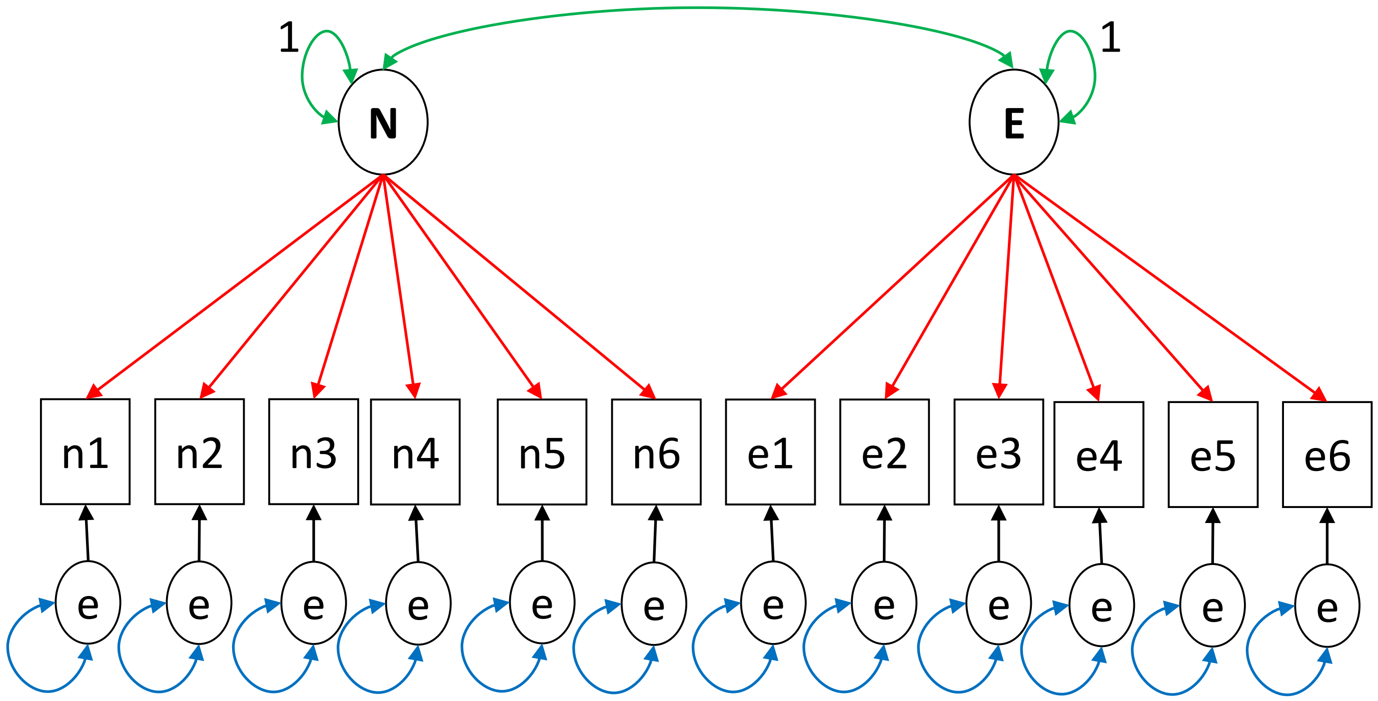
```
Ly=mxMatrix(type='Full',nrow=ny,ncol=ne,
 free=matrix(c(
     T,F,
     T,F,
     T,F,
     T,F,
     T,F,
     T,F,
     F,T,
     F,T,
     F,T,
     F,T,
     F,T,
     F,T),ny,ne,byrow=T),
 values=c(4,4,4,4,4,4,0,0,0,0,0,0,
            0,0,0,0,0,0,4,4,4,4,4,4), # read colunm-wise
 labels=matrix(c(
    'f1_1','f1_2',
    'f2_1','f2_2',
    'f3_1','f3_2',
    'f4_1','f4_2',
    'f5_1','f5_2',
    'f6_1','f6_2',
    'f7_1','f7_2',
    'f8_1','f8_2',
    'f9_1','f9_2',
    'f10_1','f10_2',
    'f11_1','f11_2',
    'f12_1','f12_2'),ny,ne,byrow=T),name='Ly'),
```



Define covariance matrix of residuals Θ

```
Te=mxMatrix(type='Diag',nrow=ny,ncol=ny,
              labels=c('rn1','rn2','rn3','rn4','rn5','rn6',
                       're1','re2','re3','re4','re5','re6'),
              free=TRUE,value=10,name='Te')
```

$$\Sigma_y = \Lambda \Psi \Lambda^t + \Theta$$

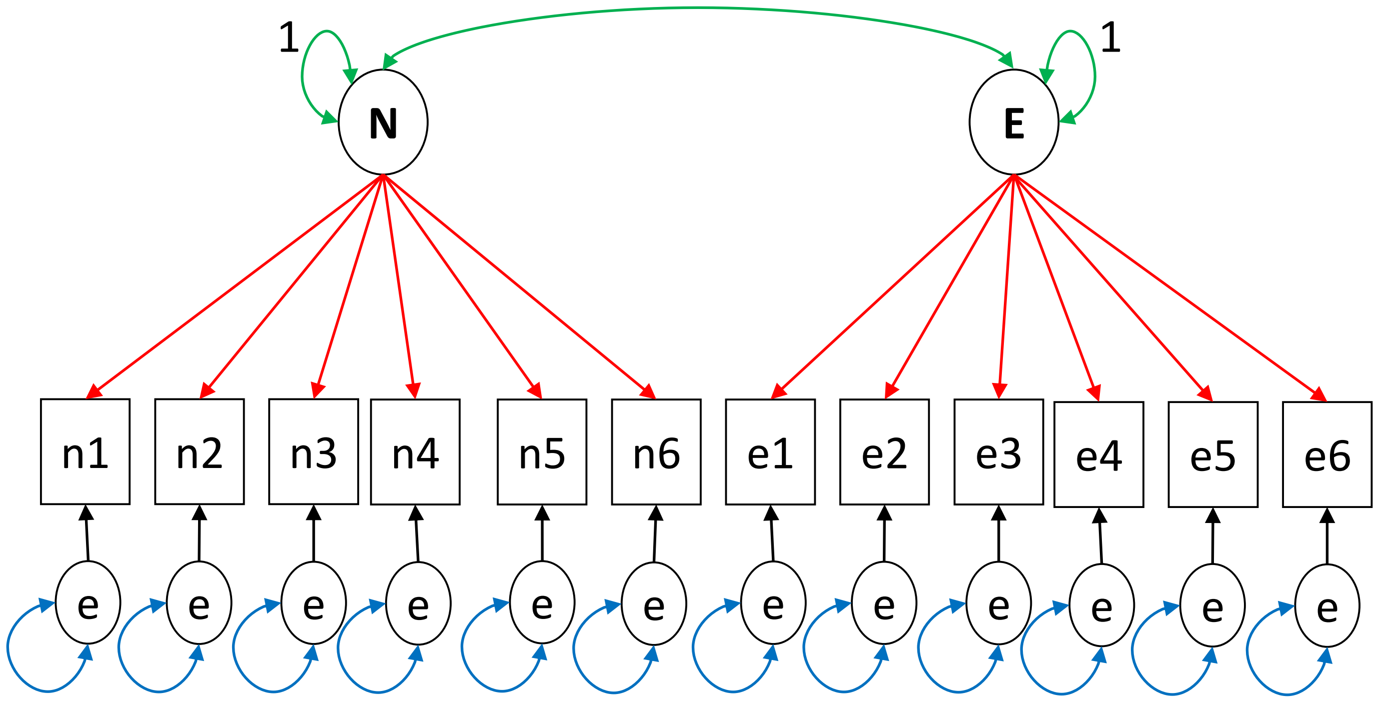


Define covariance matrix of factors Ψ

```
## latent correlation matrix
Ps=mxMatrix(type='Symm',nrow=ne,ncol=ne,
                 free=c(FALSE,TRUE,FALSE),
                 labels=c('v1_0','r12_0', 'v2_0'),
                 values=c(1,.0,1),name='Ps')
```

$$\Sigma_y = \Lambda\Psi\Lambda^t + \Theta$$

NOTE: scaling of the common factors by fixing the variances to equal 1.
Ψ is a correlation matrix!



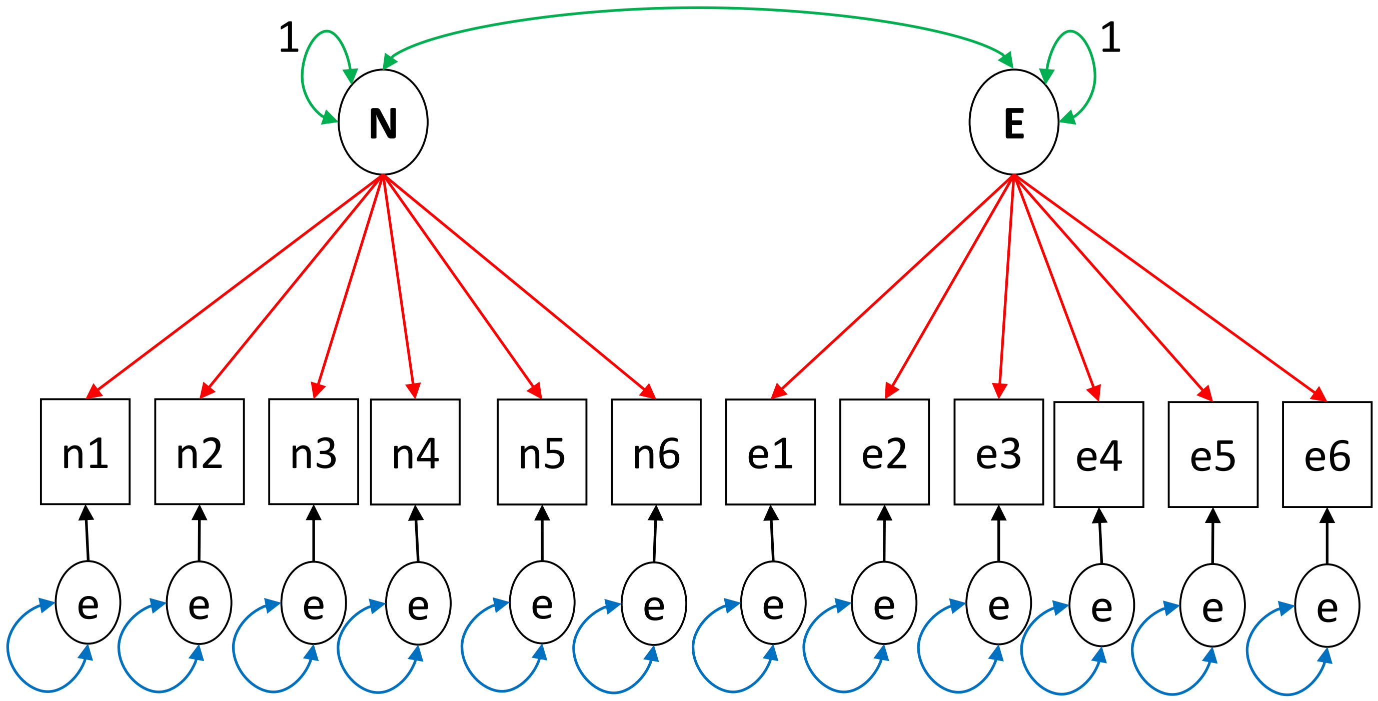
```
# means
Tau=mxMatrix(type='Full',nrow=1,ncol=ny,free=TRUE,value=25,
                         labels=c('mn1','mn2','mn3','mn4','mn5','mn6',
                                  'me1','me2','me3','me4','me5','me6'),
                         name='Means')
```

$$\Sigma_y = \Lambda \Psi \Lambda^t + \Theta$$

```
MKS=mxAlgebra(expression=Ly%*%(Ps)%*%t(Ly)+ Te%*%t(Te),name='Sigma'),
MKM=mxAlgebra(expression=Means,name='means')
```

.... assemble the model and run

```
CFM1_out = mxRun(cfamodel1)
```

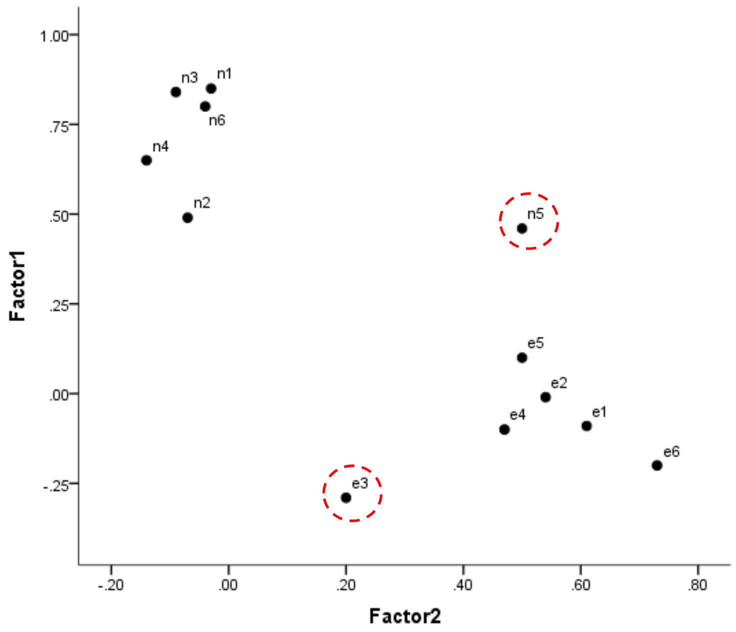


```
> mxCompare(Models1_out,CFM1_out)
     base comparison ep minus2LL   df      AIC   diffLL diffdf            p
1 models1       <NA> 90 23578.09 4242 15094.09       NA     NA           NA
2 models1       CFM1 37 23975.63 4295 15385.63 397.5483     53 2.785297e-54
```

This model does not fit (but we already know this from the EFA results).

```
> print(round(fa1s1pr$loadings[,1:2],4))
   Factor1 Factor2
n1  0.8515 -0.0252
n2  0.4858 -0.0682
n3  0.8377 -0.0870
n4  0.6465 -0.1404
n5  0.4638  0.5005
n6  0.8014 -0.0444
e1 -0.0907  0.6142
e2 -0.0126  0.5367
e3 -0.2943  0.1988
e4 -0.0996  0.4664
e5  0.1024  0.4973
e6 -0.1978  0.7308
```

Λ

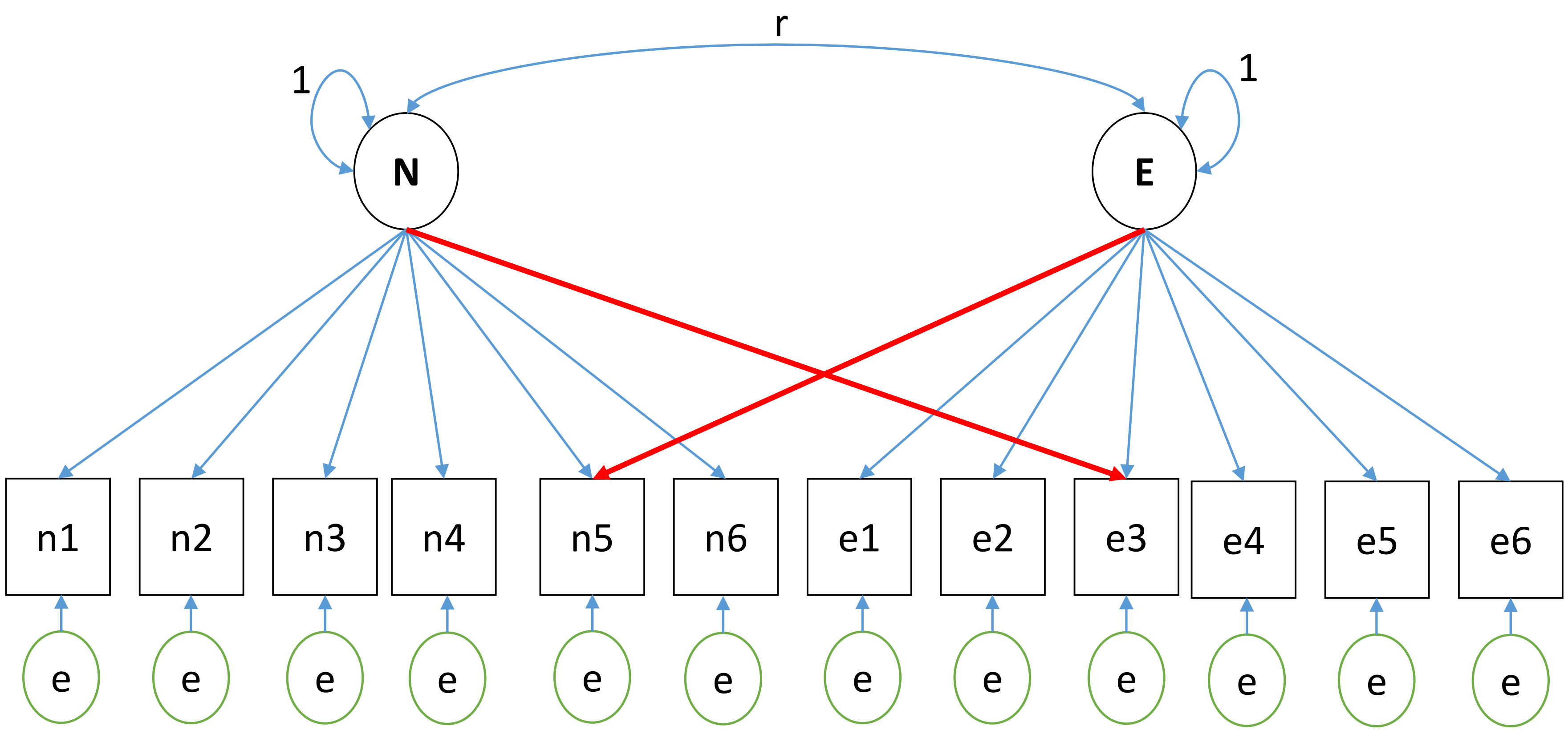



```
> round(est_Ly,3)
      [,1]  [,2]
 [1,] 4.923 0.000
 [2,] 2.383 0.000
 [3,] 5.197 0.000
 [4,] 2.993 0.000
 [5,] 0.889 0.000
 [6,] 3.785 0.000
 [7,] 0.000 2.565
 [8,] 0.000 2.225
 [9,] 0.000 1.822
[10,] 0.000 1.873
[11,] 0.000 1.576
[12,] 0.000 3.691
```

Λ

To do: free the cross loadings Ly[5,2] and Ly[9,1]

To do: free the cross loadings Ly[5,2] and Ly[9,1]



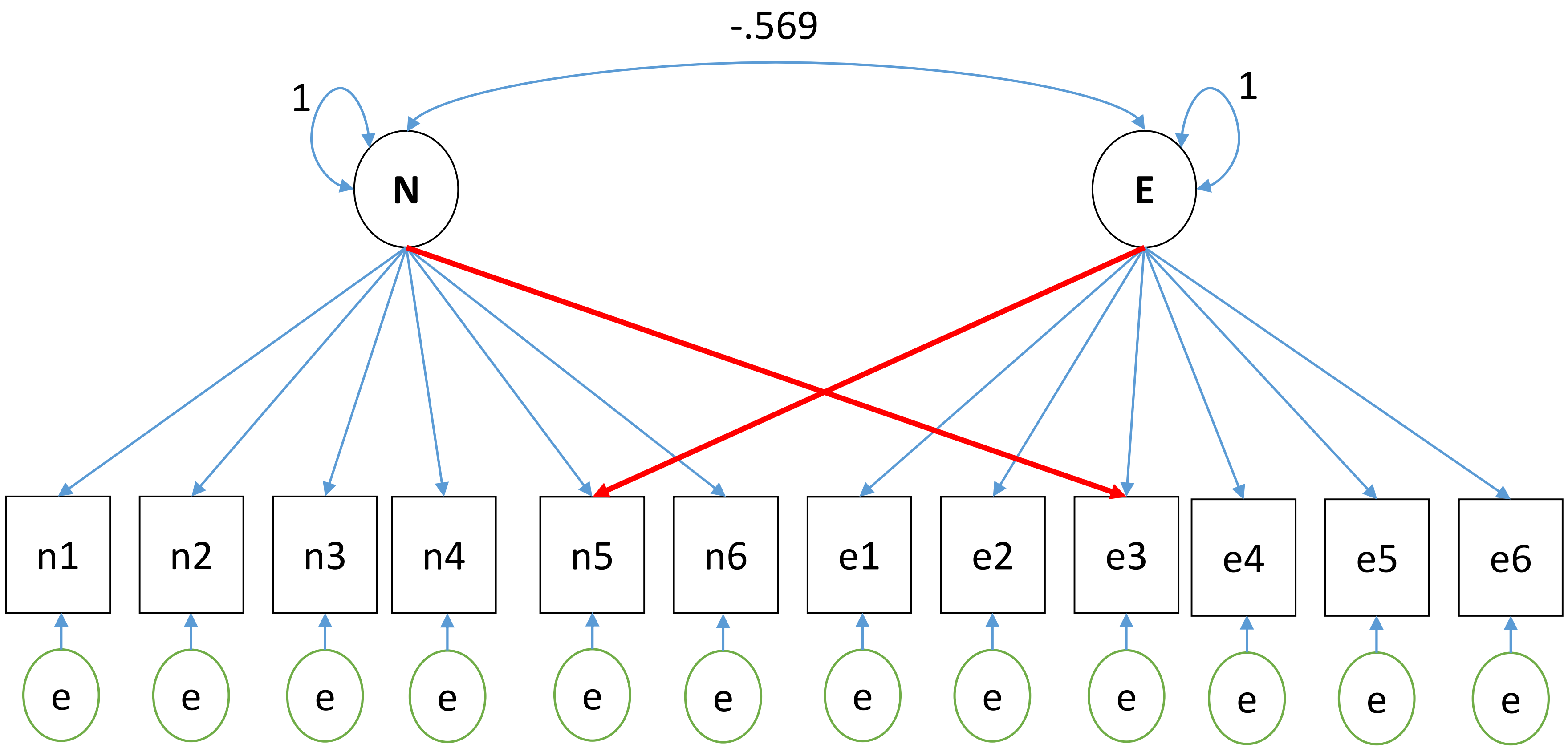
Test whether the cross loadings are unequal zero using the likelihood ratio test

```
base comparison ep minus2LL   df      AIC   diffLL diffdf            p
1 CFM1       <NA> 39 23891.96 4293 15305.96       NA     NA           NA
2 CFM1       CFM1 37 23975.63 4295 15385.63 83.67033      2 6.779838e-19
```

Given alpha=.05, we reject the hypothesis Ly[5,2] = Ly[9,1] = 0
Therefore either one or both are not equal to zero.

```
print(round(est_Ly,2))
       [,1] [,2]
 [1,]  4.88 0.00
 [2,]  2.38 0.00
 [3,]  5.20 0.00
 [4,]  3.02 0.00
 [5,]  2.30 2.32
 [6,]  3.80 0.00
 [7,]  0.00 2.50
 [8,]  0.00 2.23
 [9,] -1.47 0.89
[10,]  0.00 1.85
[11,]  0.00 1.74
[12,]  0.00 3.73
```

Inspect the correlation matrix Ψ

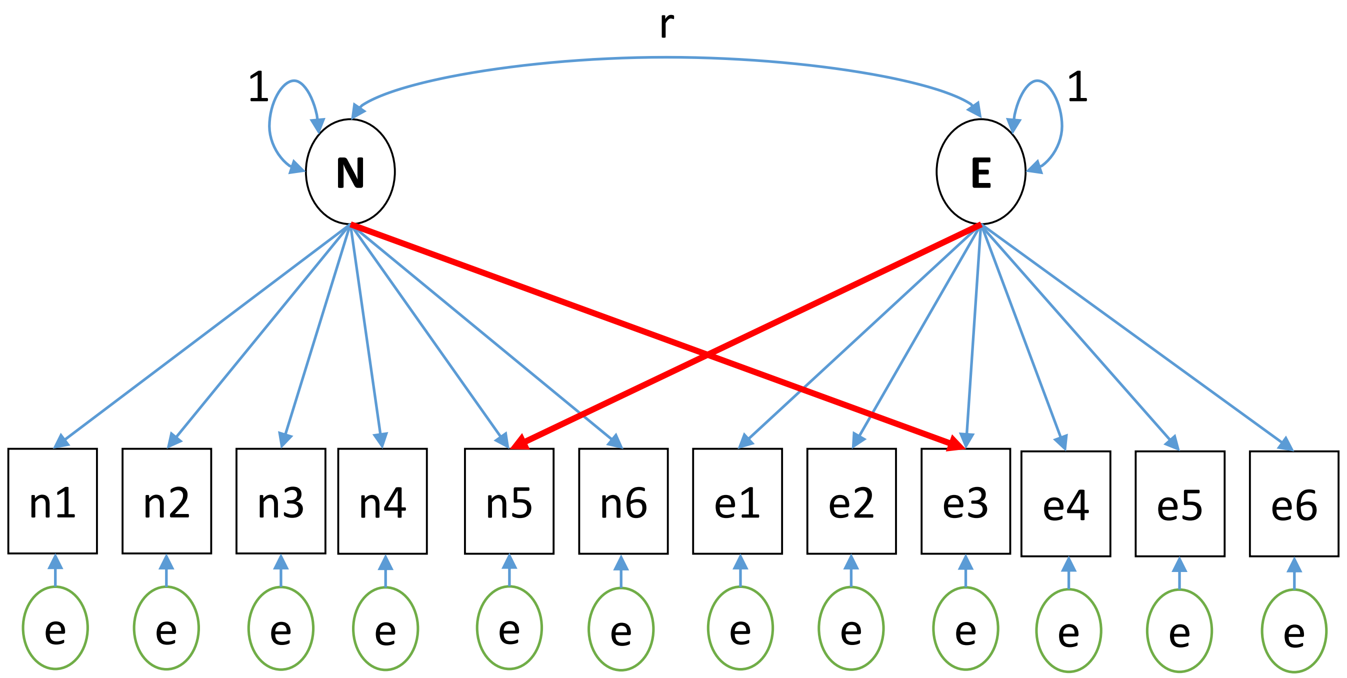



```
 1.000 -0.569
-0.569  1.000
```

Reliability of the indicators

```
> round(diag(Sfit1)/diag(Sfit),2)
 [1] 0.74 0.26 0.77 0.50 0.28 0.66 0.42 0.25 0.17 0.26 0.16 0.74
```

```
est_Ly=mxEval(CFAmodel1.Ly,CFM2_out)
est_Ps=mxEval(CFAmodel1.Ps,CFM2_out)
est_Te=mxEval(CFAmodel1.Te,CFM2_out)
est_Te=est_Te^2
Sfit1=est_Ly%*%est_Ps%*%t(est_Ly)
Sfit=Sfit1+est_Te
rel=diag(Sfit1)/diag(Sfit)
print(round(rel,3))
```



Variance of n1 due to N divided by the total variance of n1:

23.77 / 32.24 = .74.

The common factor N explains 74% of the variance in item 1.