

Today: Sections 9.3-9.5 Clarifications and Applications

Estimation (Section 9.3) : Use standardized variables (means 0 and SDs = 1)

The objective is to use  $\hat{\Sigma}$  to estimate L. Recall  $\Sigma = LL' + \Psi$ ,  
 $p \times p = p \times m + m \times p + p \times p$

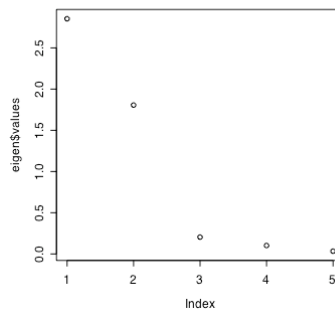
1. a) Principal Component Method (use first m PCs, assign residual to  $\varepsilon$  - see box p 486)
- b) Principal Factor Modification – (iterate between communalities and adjusted correlation matrix – see pp 490-491)

Example 9.3 First we use Principal Components approach 1a).

```
eigen=eigen(E9.3)
> loadings=eigen$vector%*%diag(eigen$values^.5)
> loadings
      [,1] [,2] [,3] [,4]
[1,] 0.5598618 -0.8160981 0.04453561 0.04437464
[2,] 0.7772594 0.5242021 0.33579190 -0.09028161
[3,] 0.6453364 -0.7479464 0.07615553 0.03744292
[4,] 0.9391057 0.1049187 -0.27203901 -0.18184384
[5,] 0.7982069 0.5432281 -0.09972820 0.24045880
      [,5]
[1,] 0.128787099
[2,] 0.013153309
[3,] -0.130059531
[4,] 0.000303964
[5,] 0.001653934
```

Eigenvalues suggest m=2 factors should do.

```
> plot(eigen$values)
```



```

> commies=loadings[,1]^2 + loadings[,2]^2
> commies
[1] 0.9794614 0.8789200 0.9758829 0.8929275 0.9322311
> ones=rep(1,5)
> specvar=ones-commies
> specvar
[1] 0.02053865 0.12107998 0.02411712 0.10707250
[5] 0.06776888

```

1a) Factor Loadings are coefficients of the first  $m$  PCs. Specific variances are diagonal elements of  $\hat{\Sigma} - LL' = \hat{\Psi}$  ( $L$  is the matrix of  $m$  factor loadings.) Note that the communalities, computed from subtracting the diagonal elements of  $LL'$  from the original data covariances (or from 1 with standardized data), are useful descriptive statistics for the researcher: How much do the original variables depend only on the common factors?

1b) Suppose we guess the specific variances – to make this short, lets guess that they are what we learned from 1a)

```

> specvar
[1] 0.02053865 0.12107998 0.02411712 0.10707250
[5] 0.06776888

```

```

> LL=E9.3-diag(specvar)
> eigen.LL=eigen(LL)
> loadings=eigen.LL$vectors%*%diag(eigen.LL$values^.5)
> loadings
      [,1]      [,2]      [,3]      [,4]
[1,] 0.5827236 0.7960398 0.01517805 0.040769478
[2,] 0.7427103 -0.5226413 0.22950638 -0.034131641
[3,] 0.6662649 0.7232420 0.06183476 -0.006177827
[4,] 0.9137868 -0.1345592 -0.18259148 -0.079848756
[5,] 0.7774556 -0.5590191 -0.06900716 0.101193413
      [,5]
[1,] 0.065746336
[2,] 0.017590270
[3,] -0.070945595
[4,] 0.009978874
[5,] -0.017012451
> eigen.LL
$values
[1] 2.77453778 1.76052088 0.09482873 0.01948122
[5] 0.01005428

```

```

$vectors

```

```

      [,1] [,2] [,3] [,4]
[1,] 0.3498383 0.5999488 0.04928854 0.29209702
[2,] 0.4458863 -0.3938974 0.74528916 -0.24453957
[3,] 0.3999923 0.5450835 0.20079954 -0.04426166
[4,] 0.5485921 -0.1014128 -0.59293973 -0.57208443
[5,] 0.4667457 -0.4213142 -0.22409088 0.72501037
      [,5]
[1,] 0.65568633
[2,] 0.17542726
[3,] -0.70753839
[4,] 0.09951903
[5,] -0.16966468

```

The eigenanalysis is quite similar and so are the loadings, though not identical, to what we got with method 1a). But we can improve 1b) by an iterative method. Since the initial solution provides, amongst other things, specific variance estimates, we can repeat the method 1b) on the original correlation matrix using these new specific variances estimates instead of our initial guess.

```

LL=E9.3-diag(specvar)
> eigen.LL=eigen(LL)
> eigen.LL
$values
[1] 2.77453778 1.76052088 0.09482873 0.01948122
[5] 0.01005428

$vectors
      [,1] [,2] [,3] [,4]
[1,] 0.3498383 -0.5999488 0.04928854 0.29209696
[2,] 0.4458863 0.3938974 0.74528916 -0.24453959
[3,] 0.3999923 -0.5450835 0.20079954 -0.04426159
[4,] 0.5485921 0.1014128 -0.59293973 -0.57208444
[5,] 0.4667457 0.4213142 -0.22409088 0.72501039
      [,5]
[1,] 0.65568636
[2,] 0.17542724
[3,] -0.70753839
[4,] 0.09951897
[5,] -0.16966461

> evals=c(2.774,1.7605)
> evecs=cbind(eigen.LL$vectors[,1],eigen.LL$vectors[,2])
> loadings=evecs%*%diag(evals^.5)
> loadings
      [,1] [,2]
[1,] 0.5826672 -0.7960351

```

```
[2,] 0.7426383 0.5226382
[3,] 0.6662003 -0.7232378
[4,] 0.9136982 0.1345584
[5,] 0.7773803 0.5590158
```

These are very close to the initial solution, because we had a good "guess" of the specific variances. Notice that the sign is  $F_2$  is reversed, but this is inconsequential.

Next we look at the

2. Max Likelihood Method (assume normality – impose "harmless" constraint to obtain uniqueness)

We get a completely different answer from method 1.

```
> factanal(factors=2,covmat=E9.3,rotation="none")
```

Call:

```
factanal(factors = 2, covmat = E9.3, rotation = "none")
```

Uniquenesses:

```
[1] 0.028 0.237 0.040 0.168
[5] 0.052
```

Loadings:

```
      Factor1 Factor2
[1,] 0.976 -0.139
[2,] 0.150 0.860
[3,] 0.979
[4,] 0.535 0.738
[5,] 0.146 0.963
```

```
      Factor1
SS loadings 2.241
Proportion Var 0.448
Cumulative Var 0.448
      Factor2
SS loadings 2.232
Proportion Var 0.446
Cumulative Var 0.895
```

The degrees of freedom for the model is 1 and the fit was 0.0233

????????????????????

For details of mle See p 492-493. Gives same answer for S and R (up to an orthog transformation). Is this surprising? Worrying!

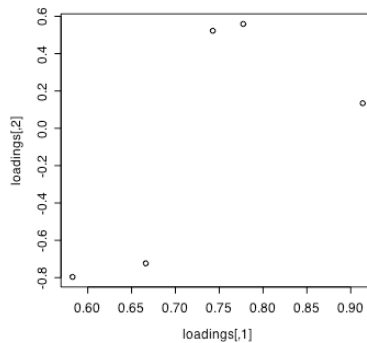
Example comparing 1. and 2. See Table 9.3 p 494. Also Table 9.4 p 496.

We will see that Factor Rotation can improve the MLE solution for interpretability. Section 9.4 discusses Factor Rotation.

```
> loadings
      [,1] [,2]
[1,] 0.5826672 -0.7960351
[2,] 0.7426383  0.5226382
[3,] 0.6662003 -0.7232378
[4,] 0.9136982  0.1345584
[5,] 0.7773803  0.5590158
```

<-----original loadings

```
plot(loadings)
>
```



```
> promax(loadings)
$loadings
```

```
Loadings:
      [,1] [,2]
[1,]      -1.004
[2,]  0.928  0.115
[3,]     -0.978
[4,]  0.796 -0.320
[5,]  0.979  0.131
```

<-----rotated loadings

```
      [,1] [,2]
SS loadings  2.462 2.096
Proportion Var 0.492 0.419
Cumulative Var 0.492 0.912
```

```
$rotmat
```

```

      [,1] [,2]
[1,] 0.7708126 -0.4836453
[2,] 0.6797138 0.9067801 <-----orthonormal rotation transformation

```

```

> varimax(loadings)
$loadings

```

```

Loadings:
      [,1] [,2]
[1,] -0.986
[2,] 0.908
[3,] 0.135 -0.974
[4,] 0.827 -0.410 <-----rotated loadings
[5,] 0.957

```

```

      [,1] [,2]
SS loadings 2.445 2.090
Proportion Var 0.489 0.418
Cumulative Var 0.489 0.907

```

```

$rotmat
      [,1] [,2]
[1,] 0.8215549 -0.5701294
[2,] 0.5701294 0.8215549 <-----orthonormal rotation transformation

```

## 9.5 Factor Score Estimation

We can estimate "factor scores" which give us  $m$ -tuples of estimated data for the unobserved factors. The  $m$  factor scores are realizations of the random variables  $F_1, F_2, F_3, \dots, F_m$ , and we get a different set of these random variable realizations for each row of our data matrix. So the vector of factor scores is comparable to the original data vector, but with fewer coordinates. We can then plot these like we did for PCs.

Note that the loading matrix  $L$  does not give these factor scores directly, except when the estimation of  $L$  is done by the principal component method which I labeled 1a). (comment p 513 top). The formula for factor scores using mle estimates is given on p 512 in the box there for the Bartlett method, and on p 514 (box) for the regression method. In the former, the matrix  $\hat{\Lambda}$  is given by the arbitrary constraint on the mle solution, and it is defined on bottom of page 511.

Two methods:

1. Regression Method (pp 513-514) "Thomson's method"
2. Weighted Least Squares and MLE (pp 511-512) "Bartlett's" method.

Read the R help for "factanal"

Here is some output which uses the assumption that there is one factor (which we know is true for this simulated data).

```
> a=factanal(data,1,method="regression")
> a
Call:
factanal(x = data, factors = 1, method = "regression")
```

```
Uniquenesses:
  V1  V2  V3  V4  V5  V6  V7  V8
0.505 0.468 0.472 0.584 0.565 0.569 0.467 0.555
```

Loadings:

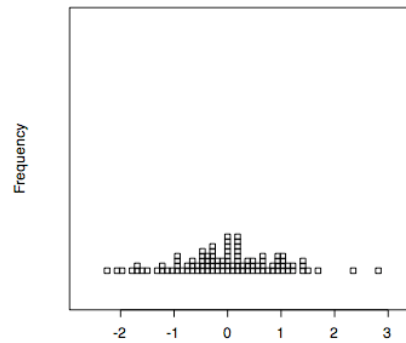
```
  Factor1
V1 0.703
V2 0.730
V3 0.727
V4 0.645
V5 0.660
V6 0.657
V7 0.730
V8 0.667
```

```
          Factor1
SS loadings    3.815
Proportion Var 0.477
```

Test of the hypothesis that 1 factor is sufficient.  
The chi square statistic is 26.69 on 20 degrees of freedom.  
The p-value is 0.144

To get the scores .....

```
> my.dotplot(a$scores)
```



```
> factanal(data,1,method="Bartlett")
```

Call:

```
>a=factanal(x = data, factors = 1, method = "Bartlett")
```

```
>a
```

Uniquenesses:

V1	V2	V3	V4	V5	V6	V7	V8
0.505	0.468	0.472	0.584	0.565	0.569	0.467	0.555

Loadings:

	Factor1
V1	0.703
V2	0.730
V3	0.727
V4	0.645
V5	0.660
V6	0.657
V7	0.730
V8	0.667

	Factor1
SS loadings	3.815
Proportion Var	0.477

Test of the hypothesis that 1 factor is sufficient.

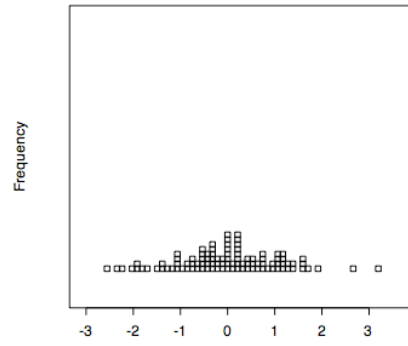
The chi square statistic is 26.69 on 20 degrees of freedom.

The p-value is 0.144

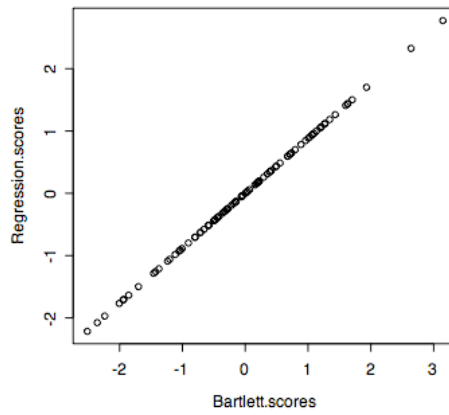
```
>
```

```
> my.dotplot(a$scores)
```

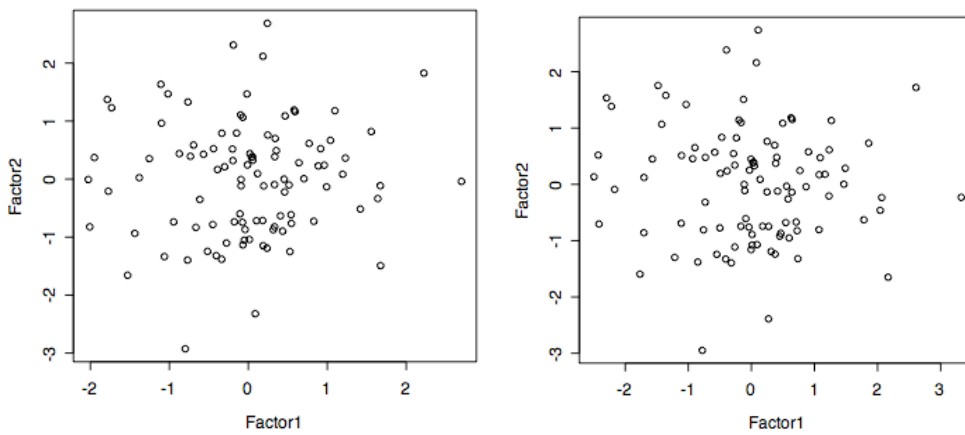




How do the two methods of producing scores compare?



Lets see what scores look like for  $m=2$  factors (we know second one is completely error)



left is regression method. right is Bartlett method.

Best Method? Johnson says on p 514 "...none is recommended as uniformly superior".

Note the intuitive approach described on pp 515-516. Remember that in math stat theory, simplicity is in danger of being under-valued!

Section 9.6 Overview of FA strategy See p 517 and example 9.14 following.

Suggestion: These are **not** to hand in.

1. Pick a data set from Ch 9 that interests you. Perform a FA on it including the estimation of factor scores. Try to interpret the result. Talk to me if you need any clarification of the outcome.
2. Return to the introductory example with  $x_i$  and  $z_i$ . How much identifiability of the model does the correlation matrix allow?

Exercise to hand in Wed. Oct 26.

Use the principal component approach and the mle approach to perform a factor analysis on data generated from the model shown in class. (I think it was ....

$$\begin{aligned}x_1 &= z_1 + z_2 \\x_2 &= z_1 + z_3 \\x_3 &= z_2 + z_4\end{aligned}$$

where  $z_i$  are  $N(0,1)$  indept, and use a moderate sample size like  $n=25$ .) Use factor rotation if it helps to clarify the solution.

Which loading estimation method works best?  
How well are the specific variances estimated?  
How well is the "variance explained by factors" estimated by the sum of eigenvalues?

Also, explore some other aspect of FA such as **one** of the following:

- 1) what happens if several more independent  $x_i$  are added to the model?
- 2) how are the results of the exercise above dependent on sample size?
- 3) how big does the variance of  $z_3$  or  $z_4$  have to be to hide the factors (using  $n=25$ )?
- 4) how reliable is the scree test for identifying the number of real factors present?
- 5) anything else you think merits exploration ....