Factor Analysis

OVERVIEW

(Note: All of the code associated with this project can be found on my GitHub repository SeanAmmirati/stats-works.)

In this project, I will be introducing the concept of factor analysis and use it to analyze a dataset of responses to a psychological survey.

Factor analysis is similar to principal components analysis, in that we are working only with a single group of variables (in this case, the response) and we wish to reduce the number of variables we use in total in order to simplify our data. This gives us a sense of what the underlying nature of the data is, and how it is organized. We wish to reduce the redundancy of the variables by limiting them to a smaller number of factors which can adequately explain the variations in the full set of variables.

Factor analysis is considered somewhat controversial by statisticians and is not encouraged by all schools of thought. This is because factor analysis is often difficult to validate in practice, as the number of factors or the interpretations are not always clear from the analysis itself.

Although at first glance factor analysis and principal component analysis may seem very similar, there are key differences which separate the two methods. First, in principal component analysis we aim to maximize the total variance of the variables in question, but in factor analysis we wish to account for the *covariance* between the variables. Secondly, while principal components analysis uses linear combinations of the variables themselves, in factor analysis we create a linear combination of factors which define the variables.

To give a concrete example, the factors that we are looking at are some underlying attributes of the variables that we believe to be correlated in some way. For instance, let's say there we have students' test scores for different classes: physics, chemistry, statistics, English, history, etc. We may expect that there would be some underlying factors that would affect an individual's ability to perform well in these classes. Perhaps this would be something like quantitative reasoning skills, critical thinking ability, or reading level and skill. We wish to reduce the variables to a smaller subset that can use these factors (which are not observed but can be derived from the data using factor analysis) to simplify our dataset. We would then use factor analysis to determine the correct number of factors and the effects these factors have on each of the variables.

It is quite easy to see how in practice this could be quite difficult to implement. This contributes to the skepticism of some statisticians to its use in the first place. Often times, the data doesn't easily lend itself to such a simplistic interpretation, and it is unclear what these factors can be and how they should be interpreted.

BASICS OF THE MODEL

Our model is constructed as follows:

$$
\begin{aligned}
y_1 - \mu_1 &= \lambda_{1\,1} f_1 + \lambda_{1\,2} f_2 + \cdots + \lambda_{1\,m} f_m + \varepsilon_1 \\
y_2 - \mu_2 &= \lambda_{2\,1} f_1 + \lambda_{2\,2} f_2 + \cdots + \lambda_{2\,m} f_m + \varepsilon_2 \\
&\qquad\qquad\qquad\qquad\qquad \cdots \\
y_p - \mu_p &= \lambda_{p\,1} f_1 + \lambda_{p\,2} f_2 + \cdots + \lambda_{p\,m} f_m + \varepsilon_p
\end{aligned}
$$

Here, $y_1 \ldots y_p$ are the original variables, $\mu_1 \ldots \mu_p$ are the means of the corresponding variables, $\lambda_{i\,j}$ are the loadings for the ith variable and mth factor ($1<i<p$, $1<j<m$), $f_1 \ldots f_m$ are the m factors, and $\varepsilon_1 \ldots \varepsilon_p$ are the error terms associated with each of the p variables.

In doing this, we are essentially defining the original *p* variables into a linear combination of *m* factors, where m<p. The *fs* in the above model are the *m* factors themselves, while the lambdas are the loadings, which serve as weights for each of the factors for each of the *p* variables.

While this may seem similar to the multiple regression model, there are key differences. The important thing is that the *fs* here are unobserved random variables, not fixed effects like in multiple regression. Also, the model here only represents one observation vector, while multiple regression represents all of the observations.

Our model can be written more simply in matrix notation as:

$$\boldsymbol{y} - \boldsymbol{\mu} = \boldsymbol{\Lambda}\mathbf{f} + \boldsymbol{\varepsilon}$$

where $\boldsymbol{y} = (y_1, \ldots, y_p)'$, $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_p)'$, $\mathbf{f} = (f_1, \ldots, f_m)'$, $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_p)'$ and

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_{1\,1} & \cdots & \lambda_{1\,m} \\ \vdots & \ddots & \vdots \\ \lambda_{p\,1} & \cdots & \lambda_{p\,m} \end{bmatrix}$$

Assumptions: In performing factor analysis, we assume that the following holds:

1) $\mathrm{E}(\mathbf{f}) = \mathbf{0}$
2) $\mathrm{cov}(\mathbf{f}) = \mathbf{I}_{mxm}$
3) $\mathrm{E}(\boldsymbol{\varepsilon}) = \mathbf{0}$
4) $\mathrm{cov}(\boldsymbol{\varepsilon}) =$ $\boldsymbol{\Psi}$ (where $\boldsymbol{\Psi}$ is a pxp matrix with $\mathrm{var}(\varepsilon_i)$ on the ith diagonal and zeros elsewhere)
5) $\mathrm{cov}(\mathbf{f}, \boldsymbol{\varepsilon}) = \mathbf{0}$

These assumptions follow naturally from the model itself. For instance, as E $(\boldsymbol{y} - \boldsymbol{\mu}) = \mathbf{0}$, it follows that both episilon and the factors themselves must also have an expected value of zero. We assume that the factors are uncorrelated, as we wish to minimize the number of variables used (so, we wish to have no correlation between the factors themselves.) The $\mathbf{\Psi}$ matrix shows that there are no covariance between the errors, only a *specific variance* for each error term, which means that the factors account for the correlations amongst the y's. This is exactly the goal of factor analysis, so these assumptions are self-checking -- if in recreating the model it is not clear what the factors should be or what number of them there should be, it is saying that these assumptions have not been met.

In achieving our goal in factor analysis of reducing the variables used, we wish to express the covariance of $\mathbf{y}$ in terms of the loadings, $\mathbf{\Lambda}$, and the variances of the errors, $\mathbf{\Psi}$ using some number of factors *m* which is less than the original number of variables *p*.

We can show that the population covariance matrix Σ can be written as follows:

$$\Sigma = \mathbf{\Lambda\Lambda}' + \mathbf{\Psi}$$

It is also of note that we can rotate the loadings by an orthogonal matrix without effecting their ability to reproduce this population covariance matrix. As such the loadings are not unique. We will use these rotations when we perform analysis using this method. The rotations prove useful as the results given in the loadings may not make it clear which variables are effected by which factors. By rotating the coordinate axis, we can more easily interpret the results from the factor analysis.

The variances of each of the individual *y*'s can be written as follows:

$$var(y_i) = {h_i}^2 + \psi_i$$

where ${h_i}^2 = \lambda_{i\,1}{}^2 + \lambda_{i\,2}{}^2 \ldots + \lambda_{i\,m}{}^2$ and ψ_i is the specific variance for the ith error term, that is, the ith diagonal of $\mathbf{\Psi}$**.** Thus we can separate the variance of each of the ys into a communal and specific part, which is ${h_i}^2$ and ψ_i , respectively. As such, the diagonal elements can be easily estimated using the loadings and the specific variance, while the off diagonal elements depend on the selection of the loadings alone. Since factor analysis is largely dealing with the estimations of the loadings, it accounts for the covariations between the variables, rather than the total variance as in principal component analysis.

There are four different methods to obtain the loadings, $\mathbf{\Lambda}$**,** from a sample. These are the principal component method, the principal factor method, the iterated principal factor method, and the maximum likelihood method. These will be explored later in the coding.

Another difficulty is determining the number of factors, *m*. There are four approaches: we can select the number of variables *m* which accounts for a prespecified amount of variance of the variables, we can choose *m* to be the number of eigenvalues of the correlation matrix **R** which is greater than the average of the eigenvalues, we can use a scree plot to determine where there is a leveling of the eigenvalues of **R,** or we can test the hypothesis using a chi-squared distribution if the number of factors is the true number of factors. Again, this will be shown further in the coding.

DATASET

The dataset I will be using to illustrate the benefits and limitations of factor analysis is collected data from an online personality questionnaire. The link can be found here: http://personality-testing.info/_rawdata/AS+SC+AD+DO.zip

In this study, 1,005 participants were prompted with 40 statements to rate on a scale of one to five. This scale is used frequently and is known as the *likert* scale. It ranges from 1 (strongly disagree) to 5 (strongly agree). The data was partitioned into four sections, with every ten statements designed to be related to some attributes of the population; in particular, it was looking for their assertiveness, social confidence, adventurousness, and dominance. The questions were designed to discover the prevalence of these attributes through the responses to the statements in the survey.

This seems like an ideal dataset to use factor analysis on, because the goal of the dataset seems to be precisely what factor analysis is used for. That is, we have some variables (the questions) that have some underlying, unobservable factor (initially hypothesized to be assertiveness, social confidence, adventurousness and dominance) that ties them all together. We are not very interested in the individual answers to the questions themselves, but of the behavior of the underlying factors that exist which are unobservable.

Some examples of questions are printed below:

AS2 I try to lead others.
SC4 I express myself easily.
AD6 I dislike changes.
DO8 I challenge others' points of view.

By looking at the coding markers of each of the questions, we can get a sense of how this survey was distributed and why. If, after doing factor analysis, there is 4 factors (that is, $m = 4$) and these factors are inclusive of each of the ten questions, we can then say that the data reflects these attributes well.

Therefore, the goal of this analysis is to see a) if the data supports the idea that these factors are well separated in the variables, and b) if not, is there a better set of factors that can describe the individual more effectively.

After doing some research, it seems that many personality scales fail to truly describe individuals completely. A very popular scale, the Myers Briggs Type Indicator (MBTI), has been purported by many to effectively separate all personalities into one of sixteen types. However, when researchers used factor analysis on data testing for the four underling dimensions of the Myers Briggs tests, they found conflicting results. One of these analyses confirmed the four-dimensionality of the data, while the other suggested there to be six, rather than the four purported by the Myers Briggs test.

The motivation for choosing this topic and subsequent dataset was to test these findings for myself. Although I was unable to find an adequate dataset with questions relating to the Myers Briggs tests, I used this dataset to see how well factor analysis can truly separate personality types from a series of related statements.

ANALYSIS

```
data <- read.csv("C:/Users/Sean/Downloads/AS+SC+AD+DO/data.csv")
data<-data[1:40]
R<-cor(data)
head(R)
```

```
##           AS1       AS2       AS3       AS4       AS5       AS6        AS7
## AS1 1.0000000 0.4137723 0.3682398 0.4448159 0.3221669 0.2931655 -0.3267101
## AS2 0.4137723 1.0000000 0.6344424 0.4177133 0.4526988 0.4913432 -0.4684408
## AS3 0.3682398 0.6344424 1.0000000 0.3838178 0.4944641 0.5456192 -0.4765066
## AS4 0.4448159 0.4177133 0.3838178 1.0000000 0.3028295 0.3159420 -0.2751444
## AS5 0.3221669 0.4526988 0.4944641 0.3028295 1.0000000 0.4661230 -0.3747829
## AS6 0.2931655 0.4913432 0.5456192 0.3159420 0.4661230 1.0000000 -0.3557865
##            AS8        AS9       AS10       SC1       SC2       SC3
## AS1 -0.2608312 -0.1858184 -0.1295499 0.4285056 0.4112874 0.4072301
## AS2 -0.2919509 -0.3102848 -0.1228974 0.2794722 0.3792339 0.3238358
## AS3 -0.2718920 -0.2233892 -0.1374873 0.2048219 0.3419072 0.3159997
## AS4 -0.2203457 -0.1744766 -0.2328334 0.2882361 0.3799405 0.3611323
## AS5 -0.1977530 -0.2327827 -0.1506424 0.2195033 0.2735981 0.2906018
## AS6 -0.2394912 -0.2095667 -0.1155672 0.2035914 0.2631961 0.2504064
##           SC4       SC5        SC6        SC7        SC8        SC9
## AS1 0.8052136 0.4244275 -0.3740276 -0.3269790 -0.3873580 -0.3303758
## AS2 0.3479028 0.3992569 -0.3088470 -0.3360871 -0.2557838 -0.3088422
## AS3 0.3174110 0.3650399 -0.2719931 -0.2595922 -0.1857361 -0.2824092
## AS4 0.4341378 0.6259835 -0.3247033 -0.4924797 -0.2697620 -0.2852133
## AS5 0.3072572 0.3285196 -0.2106510 -0.2028605 -0.1917232 -0.2209485
## AS6 0.2835608 0.2812656 -0.1879648 -0.2157404 -0.1811293 -0.1834070
##           SC10       AD1        AD2       AD3       AD4        AD5
## AS1 -0.4089318 0.1875949 0.15643465 0.1487799 0.1392974 -0.1847433
## AS2 -0.3170253 0.1590446 0.11190079 0.1109313 0.1276431 -0.1652769
## AS3 -0.2710233 0.1806743 0.09430766 0.1307351 0.1278203 -0.1418593
## AS4 -0.2724498 0.1518371 0.14846325 0.1152092 0.1269040 -0.1358413
## AS5 -0.2355771 0.2342433 0.17200751 0.1535967 0.2325989 -0.1615080
## AS6 -0.1954802 0.1327174 0.17447845 0.1674757 0.1681196 -0.1139023
##            AD6        AD7         AD8          AD9       AD10       DO1
## AS1 -0.1393561 -0.1082207 -0.11934587 -0.092401403 -0.1487103 0.1031671
## AS2 -0.1608323 -0.1895181 -0.03349077  0.002088379 -0.1154262 0.2551933
## AS3 -0.1397227 -0.1615780 -0.06755426 -0.002385583 -0.1043186 0.2574269
## AS4 -0.1123102 -0.1202723 -0.06264240 -0.096320454 -0.1239163 0.1601098
## AS5 -0.2199267 -0.1792749 -0.09872212 -0.050800773 -0.1073202 0.2189824
## AS6 -0.1192533 -0.1122125 -0.04246729 -0.043489608 -0.1163537 0.2377874
##            DO2        DO3       DO4       DO5       DO6       DO7
## AS1 0.09393072 0.05655698 0.1736394 0.1128714 0.2045945 0.2821300
## AS2 0.26932527 0.20737706 0.3264103 0.2162572 0.3004144 0.2971938
## AS3 0.27971084 0.24749086 0.3568029 0.2513066 0.3279393 0.2882716
## AS4 0.15038853 0.12135381 0.2166474 0.1918464 0.2673106 0.2415963
## AS5 0.22601369 0.15205653 0.2137453 0.1715335 0.1811936 0.2429682
## AS6 0.25424424 0.22620592 0.3082356 0.2454524 0.2617313 0.2924134
##           DO8       DO9      DO10
```

```
## AS1 0.1982907 0.1183617 0.1488767
## AS2 0.2598124 0.3015476 0.2841464
## AS3 0.2657068 0.2976032 0.3326947
## AS4 0.2472153 0.1542765 0.2298038
## AS5 0.2532139 0.2470171 0.2718944
## AS6 0.3185667 0.2968619 0.2707137
```

I begin by importing the dataset and only using the variables pertaining to the questions. The next step is to find the correlation matrix. This is the preferred matrix to work with, as it makes many of the computations easier to do (as compared with the covariance matrix itself). We can see immediately that this is a covariance matrix, as there are ones in the diagonal. However, it does not appear to have particularly strong or easily discernible groups of correlations between the variables, which implies that factor analysis may not perform as well as we'd had hoped.

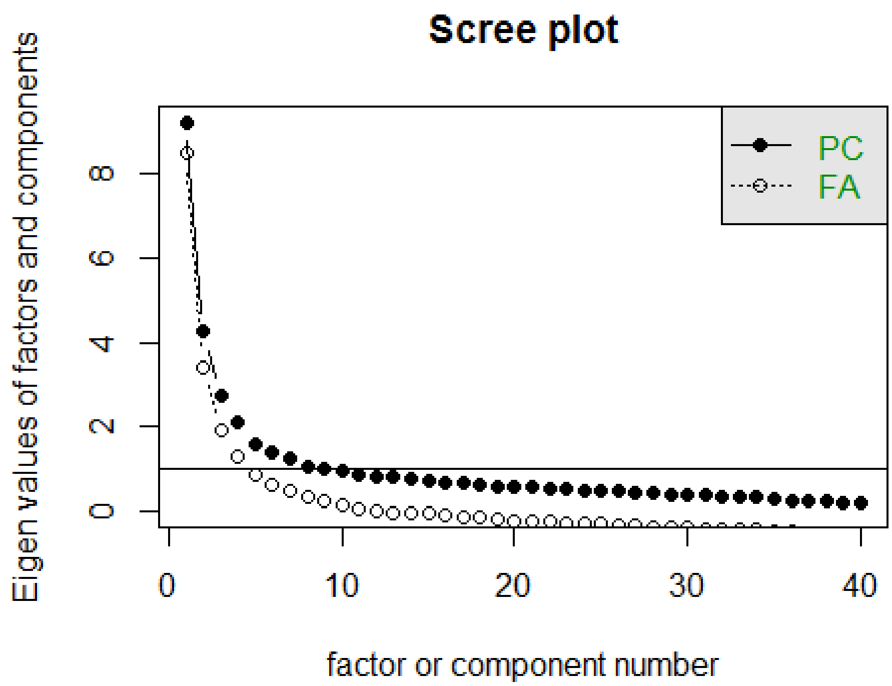
I did not produce the entire matrix, as it is a 40x40 matrix. Instead, I use the *head* function to display the first six rows of the matrix.

Selecting *m*

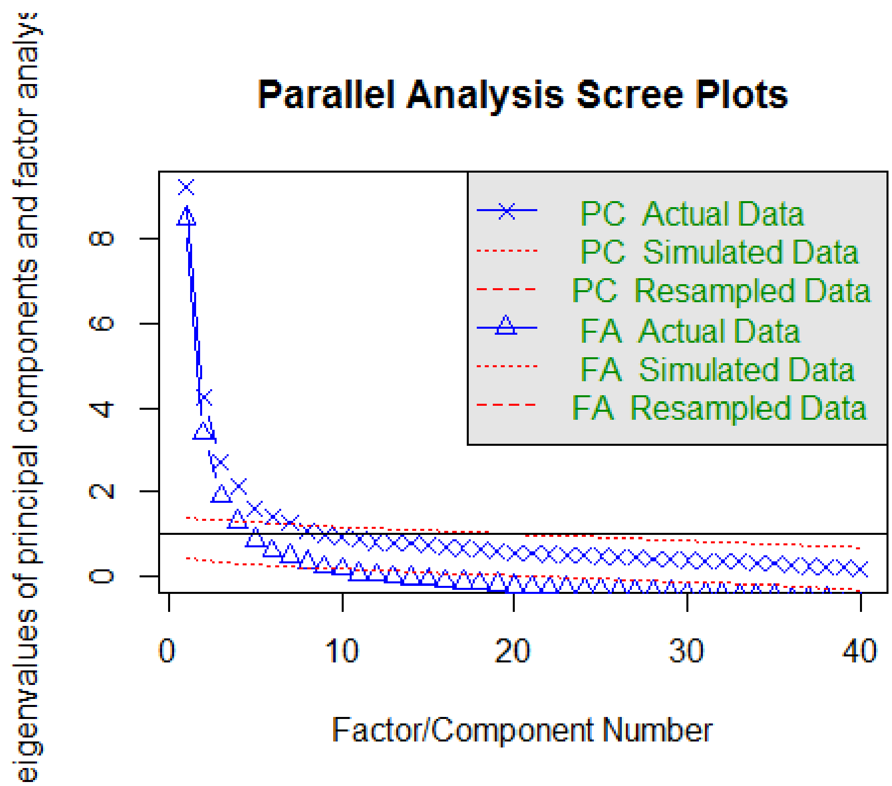
As mentioned before, it often is not clear how many factors, *m,* we should use in the factor analysis. In this example, it would be preferable to use four, as that is how the data is designed. However, as we will see, four factors are not sufficient to separate the variables.

The next step will be determining how many factors we should use.

```
eigen(R)$values

##  [1] 9.2102905 4.2476023 2.7109404 2.1236229 1.5857626 1.4029742 1.2549875
##  [8] 1.0702245 0.9805980 0.9329088 0.8759874 0.8103524 0.7940435 0.7709803
## [15] 0.7300965 0.6806931 0.6657701 0.6360141 0.5896871 0.5632701 0.5500684
## [22] 0.5214415 0.5079724 0.4900886 0.4713010 0.4559068 0.4351443 0.4127115
## [29] 0.3918664 0.3757153 0.3598197 0.3499867 0.3393500 0.3158847 0.2952949
## [36] 0.2555521 0.2354301 0.2258390 0.2038526 0.1699676

sum(eigen(R)$values[1:19])/40 ##choose 19

## [1] 0.8018384

sum(eigen(R)$values>1) ##choose 8

## [1] 8

scree(R)##choose 9
```




```
fa.parallel(data)
```



```
## Parallel analysis suggests that the number of factors =  9  and the number
of components =  7
```

Three different methods were explored to determine the amount of factors, *m,* to use. Unfortunately, it seems right from the beginning that four factors will not be enough to adequately describe the variables. The three methods provide very different numbers – 19, 8 and 9. I will use 9, as it appears to be a middle ground between the two, and the scree plot does flatten out significantly. Also, we do not wish to use nearly half of the number of variables as factors! Even 9 factors is quite high.

Now that we have determined *m,* the number of factors, we will aim to create the loadings matrix to estimate the covariance matrix, as explained above. There are four methods by which we can estimate the loadings matrices. The loadings matrix in each case will be a *p*x*m* matrix, and will use the eigenvalues of the correlation matrix. The first method we will use is the ***principal components method.***

Principal Components Method

```
eigenvectors<-eigen(R)$vectors[,1:9]
loadings<-eigenvectors%*%(diag(1,nrow=9)*(eigen(R)$values[1:9])^.5)
```

```
loadings
```

```
##              [,1]         [,2]        [,3]          [,4]        [,5]
##  [1,] -0.6277856  0.119554084  0.33171585 -0.0517100760  0.19589620
##  [2,] -0.6628787 -0.132540591  0.11886326  0.2214725941  0.22140567
##  [3,] -0.6390134 -0.197196788  0.05624741  0.2311725131  0.23664371
##  [4,] -0.6038332 -0.001172604  0.25194869  0.0287228625  0.19716616
##  [5,] -0.5623461 -0.075542614 -0.04215597  0.0889017298  0.31622038
##  [6,] -0.5588305 -0.202629294 -0.02405988  0.1321338363  0.30155513
##  [7,]  0.5610817 -0.101431288 -0.01542818 -0.4286785091 -0.17991988
##  [8,]  0.4195786 -0.089834730  0.04744726 -0.4219788203 -0.09239017
##  [9,]  0.3666707 -0.121246001  0.12235117 -0.2673284202 -0.19228688
## [10,]  0.2834756 -0.191213944  0.13406289 -0.2440643282 -0.18539624
## [11,] -0.5196860  0.270565545  0.30577781 -0.3423029175 -0.11245766
## [12,] -0.6564641  0.068827634  0.31328652 -0.2478632919 -0.34403345
## [13,] -0.5998907  0.074604175  0.26507843 -0.1892944458 -0.07212877
## [14,] -0.6074783  0.096622639  0.34093184 -0.1742184243  0.20372042
## [15,] -0.6248483  0.011604348  0.28844967 -0.1296150311  0.17592820
## [16,]  0.5974964 -0.234896621 -0.29156288  0.0140733681  0.45380545
## [17,]  0.5405337 -0.286677830 -0.22708170 -0.1085333297 -0.06070240
## [18,]  0.5065189 -0.420628297 -0.27874397  0.0715082012  0.18189460
## [19,]  0.5780675 -0.232627930 -0.26378683 -0.0556981294  0.44299196
## [20,]  0.5219373 -0.216594655 -0.13173681 -0.1032620489 -0.04359939
## [21,] -0.3992462  0.160965015 -0.36037555 -0.3690507373  0.03464572
## [22,] -0.3394289  0.150689800 -0.26903653 -0.5548752336  0.16905090
## [23,] -0.3320934 -0.027645815 -0.28907100 -0.5582492767  0.25327472
```

```
## [24,] -0.3658667  0.108273914 -0.38501438 -0.4940891181  0.19559323
## [25,]  0.3796437 -0.447822215  0.39727206 -0.1408538430  0.10110562
## [26,]  0.3772945 -0.495954442  0.47234291 -0.1033727528  0.11469191
## [27,]  0.3799486 -0.444134704  0.49194137 -0.1387764590  0.13915859
## [28,]  0.2576489 -0.483342987  0.37297134 -0.1416818547  0.21231287
## [29,]  0.2062744 -0.354617033  0.26543767 -0.0007295079  0.06470126
## [30,]  0.3334748 -0.402402498  0.39061932 -0.1902953021  0.08669907
## [31,] -0.4064146 -0.347591815 -0.19534506 -0.1439862302 -0.08445037
## [32,] -0.4213233 -0.421047564 -0.21614195 -0.1475954934 -0.14140091
## [33,] -0.3177238 -0.558528507 -0.21188225 -0.0260375231 -0.11058613
## [34,] -0.4535206 -0.568930452 -0.13969631  0.0740933171 -0.20728453
## [35,] -0.3733466 -0.534136456 -0.20889608  0.0462874732 -0.18328216
## [36,] -0.4519620 -0.486894058  0.02868335 -0.0237170894 -0.22067245
## [37,] -0.4792771 -0.359058583 -0.17026707  0.0243226733  0.01058587
## [38,] -0.4884591 -0.361002718 -0.23670362 -0.0576248626  0.04542125
## [39,] -0.3794872 -0.555721465 -0.09478251  0.1039076241 -0.19174868
## [40,] -0.4024135 -0.549377337 -0.11356106  0.1869168187 -0.12464387
##               [,6]          [,7]          [,8]         [,9]
##  [1,] -0.256426344 -0.0051466920 -0.1050080325 -0.217875830
##  [2,]  0.132387083 -0.2481776856  0.0048103856  0.073391368
##  [3,]  0.106747616 -0.3142731034  0.0725529294 -0.009258277
##  [4,] -0.096650456  0.2237402385  0.3933113732  0.175740568
##  [5,]  0.101831072 -0.3414194020  0.0325078492  0.085362217
##  [6,]  0.073081367 -0.2777839481 -0.0712005483  0.093765748
##  [7,] -0.065822588  0.1121413950  0.1320328403  0.243711028
##  [8,]  0.008179850 -0.1017128152  0.1404356750  0.359181871
##  [9,] -0.331672078 -0.1723038527  0.3330872957 -0.293884580
## [10,] -0.106304884 -0.4565341789 -0.0821946123 -0.171057932
## [11,] -0.001271796 -0.1320537729 -0.2283180107  0.265527940
## [12,]  0.087823904 -0.1132566054  0.0377935294 -0.011283238
## [13,]  0.033343619  0.0308950899  0.1426224411 -0.200220801
## [14,] -0.252897043  0.0281370136 -0.0346131237 -0.254485357
## [15,] -0.039241251  0.1844820354  0.4216051360  0.098786819
## [16,] -0.094810107  0.0093078526  0.0096644094  0.016502011
## [17,] -0.083413069 -0.3147783298 -0.2285523728 -0.186028731
## [18,]  0.002006571  0.0463524348  0.3018104199 -0.260546935
## [19,] -0.109840640  0.0355312724  0.0666607568  0.098413082
## [20,]  0.070382456 -0.2508050668  0.2867068313  0.083829363
## [21,] -0.123678525 -0.1924745951  0.1507108412 -0.019742963
## [22,]  0.011646505 -0.0474249609 -0.1572321905  0.166900898
## [23,] -0.005346193  0.1102181121 -0.1334792270 -0.157127058
## [24,]  0.064851965 -0.0003570599  0.0709592952 -0.068600025
## [25,]  0.076120098  0.0523492570 -0.1131530027  0.095050608
## [26,]  0.018459564  0.1824386999 -0.1389626180 -0.114702885
## [27,]  0.013283974  0.1229109140 -0.0973772037 -0.116962488
## [28,]  0.117125266  0.1369392140 -0.1222056233  0.146276756
## [29,]  0.099806178 -0.2945216155  0.1256130100 -0.077929380
## [30,]  0.083796843 -0.1520357393 -0.0181765447  0.107031691
## [31,]  0.611418809  0.1300478000 -0.0035088423 -0.175680364
## [32,]  0.569277998  0.1274642030  0.0120766707 -0.147817867
```

```
## [33,] -0.064299580  0.1599389824  0.0008566237  0.011078551
## [34,] -0.111993462  0.0128182266  0.0851071945  0.058119259
## [35,] -0.242376556  0.0930813931 -0.0879622294  0.120472821
## [36,] -0.121746325  0.0211133535  0.0949324784 -0.117122331
## [37,] -0.329825553  0.0814961168 -0.2338333561  0.027777255
## [38,] -0.266631231  0.1774163361 -0.1496133363 -0.028050715
## [39,] -0.083409801 -0.1393220930 -0.0449142959  0.188743726
## [40,] -0.136300169 -0.0389545949  0.0957233294  0.133687572
```

We first determine the loadings by multiplying the first m eigenvectors (in this case, 9) by the square roots of their corresponding eigenvalues. This produces the above matrix, a unsightly 40x9 matrix. Each column corresponds to a different loading.

To determine the communalities and specific variances of the variables, we can square the entries in each of the rows of the loadings matrix. This will give us the ${h_i}^2$ value, from which we can also determine the specific variances.

```
loading1<-as.matrix(loadings[,1],nrow=40);loading2<-as.matrix(loadings[,2],nr
ow=40);loading3<-as.matrix(loadings[,3],nrow=40);loading4<-as.matrix(loadings
[,4],nrow=40);loading5<-as.matrix(loadings[,5],nrow=40);loading6<-as.matrix(l
oadings[,6],nrow=40);loading7<-as.matrix(loadings[,7],nrow=40);loading8<-as.m
atrix(loadings[,8],nrow=40);loading9<-as.matrix(loadings[,9],nrow=40)

hisq<-as.matrix(loading1^2+loading2^2+loading3^2+loading4^2+loading5^2+loadin
g6^2+loading7^2+loading8^2+loading9^2,nrow=40)

psi<-as.matrix(rep(1,40))-hisq

hisq

##            [,1]
##  [1,] 0.6837702
##  [2,] 0.6537022
##  [3,] 0.6753417
##  [4,] 0.7127732
##  [5,] 0.5668960
##  [6,] 0.5586903
##  [7,] 0.6352115
##  [8,] 0.5321160
##  [9,] 0.6095668
## [10,] 0.4845742
## [11,] 0.7066714
## [12,] 0.7357218
## [13,] 0.5392320
## [14,] 0.6971639
## [15,] 0.7446073
## [16,] 0.7127661
```

```
## [17,] 0.6342772
## [18,] 0.7105137
## [19,] 0.6846617
## [20,] 0.5063354
## [21,] 0.5280230
## [22,] 0.6017274
## [23,] 0.6250849
## [24,] 0.5901456
## [25,] 0.5629343
## [26,] 0.7013619
## [27,] 0.6606930
## [28,] 0.5730633
## [29,] 0.3615022
## [30,] 0.5113686
## [31,] 0.7736378
## [32,] 0.8056109
## [33,] 0.5005422
## [34,] 0.6206625
## [35,] 0.5937235
## [36,] 0.5294148
## [37,] 0.5592003
## [38,] 0.5560673
## [39,] 0.5733943
## [40,] 0.5742527
```

```
psi
```

```
##            [,1]
##  [1,] 0.3162298
##  [2,] 0.3462978
##  [3,] 0.3246583
##  [4,] 0.2872268
##  [5,] 0.4331040
##  [6,] 0.4413097
##  [7,] 0.3647885
##  [8,] 0.4678840
##  [9,] 0.3904332
## [10,] 0.5154258
## [11,] 0.2933286
## [12,] 0.2642782
## [13,] 0.4607680
## [14,] 0.3028361
## [15,] 0.2553927
## [16,] 0.2872339
## [17,] 0.3657228
## [18,] 0.2894863
## [19,] 0.3153383
## [20,] 0.4936646
## [21,] 0.4719770
## [22,] 0.3982726
```

```
## [23,] 0.3749151
## [24,] 0.4098544
## [25,] 0.4370657
## [26,] 0.2986381
## [27,] 0.3393070
## [28,] 0.4269367
## [29,] 0.6384978
## [30,] 0.4886314
## [31,] 0.2263622
## [32,] 0.1943891
## [33,] 0.4994578
## [34,] 0.3793375
## [35,] 0.4062765
## [36,] 0.4705852
## [37,] 0.4407997
## [38,] 0.4439327
## [39,] 0.4266057
## [40,] 0.4257473
```

The above are the communalities and the specific variances of each of the variables.

To determine the amount of the total variance of which each loading contributes, we can divide the eigenvalues of **R** by the number of variables, $p = 40$.

```
eigen(R)$values[1:9]/40

## [1] 0.23025726 0.10619006 0.06777351 0.05309057 0.03964406 0.03507436
## [7] 0.03137469 0.02675561 0.02451495

sum(eigen(R)$values[1:9]/40)

## [1] 0.6146751
```

Thus, all nine factors account for about 60% of the variance of the variables. This is not very good, especially considering there were supposed to be only four underlying factors!

Rotating the Loadings

Because we have m=9, I will use the varimax rotations from the *psych* package to rotate the loadings we have above. This will make it easier to see which variables correspond to the factors.

```
print(varimax(loadings)$loadings,cutoff=.3)

## 
## Loadings:
##       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]  
##  [1,]                             -0.408                      -0.522
##  [2,] -0.681                                                        
##  [3,] -0.713                                                        
##  [4,]                                           0.744               
##  [5,] -0.681                                                        
##  [6,] -0.655                                                        
```

```
## [7,]  0.506                                                     0.486
## [8,]                                                            0.578
## [9,]  0.302                                               0.681
## [10,]                                              -0.407  0.481
## [11,]                     -0.313 -0.728
## [12,]                            -0.753
## [13,]                            -0.478         0.355
## [14,]                            -0.396         0.337         -0.501
## [15,]                                           0.745
## [16,]                             0.759
## [17,]                             0.338        -0.618
## [18,]                             0.727                 0.301
## [19,]                             0.722
## [20,]                             0.360                 0.327  0.457
## [21,]              -0.397 -0.534
## [22,]                     -0.718
## [23,]                     -0.747
## [24,]                     -0.688
## [25,]               0.716
## [26,]               0.784
## [27,]               0.769
## [28,]               0.731
## [29,]               0.391                               0.326
## [30,]               0.643
## [31,]                                    0.794
## [32,]       -0.349                       0.786
## [33,]       -0.657
## [34,]       -0.742
## [35,]       -0.765
## [36,]       -0.616
## [37,]       -0.642
## [38,]       -0.623
## [39,]       -0.685
## [40,]       -0.698
##
##                  [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
## SS loadings    2.970 4.308 3.507 2.557 3.954 1.746 2.289 1.437 1.819
## Proportion Var 0.074 0.108 0.088 0.064 0.099 0.044 0.057 0.036 0.045
## Cumulative Var 0.074 0.182 0.270 0.334 0.432 0.476 0.533 0.569 0.615
```

```
loadingsrot<- print(varimax(loadings)$loadings,cutoff=.3,sort=T)
```

```
##
## Loadings:
##       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]
## [1,] -0.681
## [2,] -0.713
## [3,] -0.681
## [4,] -0.655
## [5,]  0.506                                                     0.486
```

```
##  [6,]        -0.657
##  [7,]        -0.742
##  [8,]        -0.765
##  [9,]        -0.616
## [10,]        -0.642
## [11,]        -0.623
## [12,]        -0.685
## [13,]        -0.698
## [14,]                0.716
## [15,]                0.784
## [16,]                0.769
## [17,]                0.731
## [18,]                0.643
## [19,]               -0.397 -0.534
## [20,]                      -0.718
## [21,]                      -0.747
## [22,]                      -0.688
## [23,]                      -0.313 -0.728
## [24,]                             -0.753
## [25,]                              0.759
## [26,]                              0.727                0.301
## [27,]                              0.722
## [28,]                                     0.794
## [29,]        -0.349                       0.786
## [30,]                                            0.744
## [31,]                                            0.745
## [32,]                              0.338        -0.618
## [33,]  0.302                                            0.681
## [34,]                             -0.408                      -0.522
## [35,]                                                          0.578
## [36,]                             -0.396         0.337        -0.501
## [37,]                                           -0.407  0.481
## [38,]                             -0.478         0.355
## [39,]                              0.360                0.327  0.457
## [40,]                0.391                              0.326
##
##                 [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
## SS loadings    2.970 4.308 3.507 2.557 3.954 1.746 2.289 1.437 1.819
## Proportion Var 0.074 0.108 0.088 0.064 0.099 0.044 0.057 0.036 0.045
## Cumulative Var 0.074 0.182 0.270 0.334 0.432 0.476 0.533 0.569 0.615
```

We can see from the above that the factors do not separate so easily. The values of the loadings below .3 have been omitted to make it clearer what the underling relationships are. Although they do not separate cleanly into the four theorized factors, we can see that the first factor, for instance, largely is between the first and tenth variables (which was assertiveness), the second factor between the 30th and 40th variables (which was dominance). Similarly, the fifth factor is between the 20th and 30th variables (social confidence) and the third and factor is largely between the 30th and 40th variables (adventurousness). However, the split is not so simple as we assumed before the analysis, as the other factors show relationships between variables which are not in the same groups.

The second function simply sorts the variables by factor, so we can get a better sense of how they separate. We can see that towards the top there are larger values for the loading matrix, which indicates a stronger association with the factor. We can also see that the 6th-9th factors also have some large values, which means that their correlations cannot be ignored. This suggests that the initial separation into only four factors may have been overly simplistic.

Principal Factors Method

In the previous method, we had ignored the structure of the specific variances. Now, using the principal factors method, we will take the psi matrix into account.

We first make an initial estimate of the his, and use this to account for the specific variances. Below I create a diagonal matrix which has the specific variances in each of the diagonal elements. We will then subtract this matrix from the covariance matrix, and use the new matrix to compute our loadings.

```
hinit<-1- 1/diag(solve(R))
psiint<-1-hinit
diag(psiint,nrow=40)

New<-R-diag(psiint,nrow=40)
hinit

##       AS1       AS2       AS3       AS4       AS5       AS6       AS7
## 0.7033517 0.5479249 0.5511059 0.5052776 0.3909682 0.4178874 0.4791205
##       AS8       AS9      AS10       SC1       SC2       SC3       SC4
## 0.3272596 0.2679338 0.2193710 0.5780393 0.6556874 0.4569506 0.6958877
##       SC5       SC6       SC7       SC8       SC9      SC10       AD1
## 0.5721770 0.6392857 0.4860941 0.5795566 0.5721815 0.3779786 0.3622515
##       AD2       AD3       AD4       AD5       AD6       AD7       AD8
## 0.3823573 0.4021933 0.4125813 0.4284814 0.6107542 0.5793594 0.3926540
##       AD9      AD10       DO1       DO2       DO3       DO4       DO5
## 0.2303057 0.3609034 0.6185178 0.6575069 0.4241176 0.5388412 0.4387844
##       DO6       DO7       DO8       DO9      DO10
## 0.4146146 0.4423540 0.4621925 0.4584707 0.4849043

eigen(New)
```

```
## $values
##  [1]  8.72480786  3.73122125  2.22214484  1.55959578  1.13192575
##  [6]  0.94924084  0.69195388  0.57596597  0.51777057  0.43404257
## [11]  0.27746408  0.24907073  0.20382247  0.18319157  0.14466391
## [16]  0.10130984  0.08749738  0.06797501  0.03771954  0.01208288
## [21] -0.01256549 -0.03911447 -0.04221706 -0.06783310 -0.09186545
## [26] -0.09808676 -0.10163170 -0.11804259 -0.12715993 -0.13788099
## [31] -0.15202202 -0.15751357 -0.17112680 -0.18008415 -0.18710930
## [36] -0.19541343 -0.20017634 -0.21596482 -0.23296518 -0.24850889
##
## $vectors
##              [,1]          [,2]         [,3]         [,4]        [,5]
##  [1,] -0.21191434  0.0638645803  0.216822536 -0.022468256 -0.25615848
##  [2,] -0.21927351 -0.0651230428  0.060958567  0.176672739 -0.13311399
##  [3,] -0.21134289 -0.0975909567  0.024092374  0.184194194 -0.14976161
…
```

We will then use this new matrix to compute our eigenvalues and vectors, and subsequently create our new loadings matrix. In a similar way, we will determine the specific variances and communalities, and then rotate the loadings matrix.

```
eigenvectors2<-eigen(New)$vectors[,1:9]

loadings2<-eigenvectors2%*%(diag(1,nrow=9)*(eigen(New)$values[1:9])^.5)

loading21<-as.matrix(loadings2[,1],nrow=40);loading22<-as.matrix(loadings2[,2
],nrow=40);loading23<-as.matrix(loadings2[,3],nrow=40);loading24<-as.matrix(l
oadings2[,4],nrow=40);loading25<-as.matrix(loadings2[,5],nrow=40);loading26<-
as.matrix(loadings2[,6],nrow=40);loading27<-as.matrix(loadings2[,7],nrow=40);
loading28<-as.matrix(loadings2[,8],nrow=40);loading29<-as.matrix(loadings2[,9
],nrow=40)

hisq2<-as.matrix(loading21^2+loading22^2+loading23^2+loading24^2+loading25^2+
loading26^2+loading27^2+loading28^2+loading29^2,nrow=40)

psi<-as.matrix(rep(1,40))-hisq2

hisq2

##            [,1]
##  [1,] 0.7552672
##  [2,] 0.5893817
##  [3,] 0.6055601
##  [4,] 0.5499846
##  [5,] 0.4244615
##  [6,] 0.4485527
##  [7,] 0.4914621
##  [8,] 0.3067924
##  [9,] 0.2816954
## [10,] 0.2186141
```

```
## [11,] 0.5737638
## [12,] 0.6939457
## [13,] 0.4345569
## [14,] 0.7499541
## [15,] 0.6341489
## [16,] 0.6576837
## [17,] 0.5354152
## [18,] 0.5920439
## [19,] 0.5895881
## [20,] 0.3794710
## [21,] 0.3868157
## [22,] 0.4357815
## [23,] 0.4394091
## [24,] 0.4513715
## [25,] 0.4655380
## [26,] 0.6347434
## [27,] 0.5902703
## [28,] 0.4267045
## [29,] 0.2176438
## [30,] 0.3967158
## [31,] 0.6709917
## [32,] 0.7261606
## [33,] 0.4170825
## [34,] 0.5583676
## [35,] 0.4810193
## [36,] 0.4339867
## [37,] 0.4426319
## [38,] 0.4679031
## [39,] 0.4653067
## [40,] 0.4838398
```

```
psi2
```

```
##            [,1]
##  [1,] 0.2447328
##  [2,] 0.4106183
##  [3,] 0.3944399
##  [4,] 0.4500154
##  [5,] 0.5755385
##  [6,] 0.5514473
##  [7,] 0.5085379
##  [8,] 0.6932076
##  [9,] 0.7183046
## [10,] 0.7813859
## [11,] 0.4262362
## [12,] 0.3060543
## [13,] 0.5654431
## [14,] 0.2500459
## [15,] 0.3658511
## [16,] 0.3423163
```

```
## [17,] 0.4645848
## [18,] 0.4079561
## [19,] 0.4104119
## [20,] 0.6205290
## [21,] 0.6131843
## [22,] 0.5642185
## [23,] 0.5605909
## [24,] 0.5486285
## [25,] 0.5344620
## [26,] 0.3652566
## [27,] 0.4097297
## [28,] 0.5732955
## [29,] 0.7823562
## [30,] 0.6032842
## [31,] 0.3290083
## [32,] 0.2738394
## [33,] 0.5829175
## [34,] 0.4416324
## [35,] 0.5189807
## [36,] 0.5660133
## [37,] 0.5573681
## [38,] 0.5320969
## [39,] 0.5346933
## [40,] 0.5161602
```

```
loadings2
```

```
##             [,1]          [,2]        [,3]         [,4]        [,5]
##  [1,] -0.6259480  0.1233631821  0.32321433 -0.028059206 -0.27253213
##  [2,] -0.6476854 -0.1257940749  0.09087008  0.220635589 -0.14162264
##  [3,] -0.6242601 -0.1885102965  0.03591417  0.230028666 -0.15933437
##  [4,] -0.5877130  0.0006695812  0.21839110  0.045113069 -0.17633548
##  [5,] -0.5391421 -0.0704108777 -0.04763328  0.085072781 -0.20050663
##  [6,] -0.5372967 -0.1879751814 -0.02947288  0.123122393 -0.19626585
##  [7,]  0.5431313 -0.0931257154  0.01161833 -0.384331764  0.10768218
##  [8,]  0.3987970 -0.0780817196  0.05816718 -0.338101641  0.05357757
##  [9,]  0.3459807 -0.1030143959  0.11610427 -0.192910985  0.05665402
## [10,]  0.2653920 -0.1623512031  0.12026411 -0.170852652  0.08719401
## [11,] -0.5113188  0.2655349880  0.28030216 -0.302548928  0.08179445
## [12,] -0.6513557  0.0729607928  0.29405348 -0.225240479  0.32221455
## [13,] -0.5813585  0.0722882675  0.22719484 -0.151865179  0.05508996
## [14,] -0.6053920  0.1008112535  0.33619967 -0.148497585 -0.28197812
## [15,] -0.6130887  0.0136223537  0.26108619 -0.097399086 -0.16055386
## [16,]  0.5916555 -0.2336916422 -0.26059969  0.003323202 -0.42552828
## [17,]  0.5248875 -0.2695776491 -0.18204223 -0.107338860  0.02061318
## [18,]  0.4979476 -0.4086837017 -0.24180484  0.053979070 -0.15242309
## [19,]  0.5677672 -0.2267253033 -0.22509260 -0.060156852 -0.39922711
## [20,]  0.5001600 -0.1975776769 -0.09896936 -0.091620259  0.04431489
```

```
## [21,] -0.3810063  0.1418651559 -0.30233282 -0.305747466 -0.07230322
## [22,] -0.3247096  0.1342996395 -0.22018084 -0.463025572 -0.13323832
## [23,] -0.3181198 -0.0295386110 -0.23372806 -0.474135977 -0.19961226
## [24,] -0.3508598  0.0944251769 -0.32642084 -0.427418826 -0.14986859
## [25,]  0.3646370 -0.4079952789  0.36643304 -0.106581851 -0.04072426
## [26,]  0.3699894 -0.4753760463  0.47224997 -0.082378706 -0.07577251
## [27,]  0.3710545 -0.4209741389  0.48394767 -0.111793306 -0.09854217
## [28,]  0.2463416 -0.4378054656  0.33756747 -0.104349741 -0.11124873
## [29,]  0.1933009 -0.3055511128  0.21799039  0.011168149 -0.01790324
## [30,]  0.3174274 -0.3583648212  0.34787708 -0.137670355 -0.03094618
## [31,] -0.4001518 -0.3450609428 -0.18825758 -0.170962078  0.19698605
## [32,] -0.4164799 -0.4206024279 -0.20806101 -0.178098173  0.24305345
## [33,] -0.3051397 -0.5179683200 -0.17421092 -0.033562132  0.05982648
## [34,] -0.4419773 -0.5421995618 -0.12149627  0.063002539  0.13178726
## [35,] -0.3591802 -0.4962425675 -0.17394198  0.033167740  0.08066073
## [36,] -0.4349692 -0.4470774974  0.03083057 -0.016512509  0.12745323
## [37,] -0.4617144 -0.3336927929 -0.14309392  0.019083837 -0.07783987
## [38,] -0.4711689 -0.3390176703 -0.20274163 -0.055244133 -0.09265654
## [39,] -0.3664241 -0.5177213635 -0.07970623  0.088601753  0.12585837
## [40,] -0.3896532 -0.5164152355 -0.09984642  0.162377846  0.06391748
##               [,6]         [,7]          [,8]         [,9]
##  [1,] -0.154205050 -0.020272336  0.3206288667  0.204245329
##  [2,]  0.165125057  0.213456592 -0.0623754973 -0.018602845
##  [3,]  0.150331743  0.262243743 -0.0886518191  0.038759270
##  [4,]  0.008099623 -0.222692497 -0.2612129668  0.076575126
##  [5,]  0.132699308  0.241445296 -0.0545068739  0.015635074
##  [6,]  0.115708316  0.219980564 -0.0417705617 -0.080358994
##  [7,] -0.116758062 -0.068051049 -0.0672253340  0.074660221
##  [8,] -0.035450009  0.088867061 -0.0923174773  0.058423021
##  [9,] -0.221731953  0.057324538 -0.1109984715  0.180843773
## [10,] -0.095832748  0.236723969  0.0224387598  0.069575472
## [11,] -0.040526170  0.166110049  0.0616212332 -0.178830709
## [12,] -0.030628182  0.115152756 -0.0807735596  0.051113790
## [13,]  0.017981122 -0.030264564 -0.0422961434  0.103004423
## [14,] -0.151140424 -0.058128330  0.2667033116  0.247679807
## [15,]  0.048059070 -0.210281588 -0.3137693066  0.098368118
## [16,]  0.047276489 -0.007645798  0.0410154277 -0.006513431
## [17,] -0.110982450  0.276000498  0.2159586234  0.083778169
## [18,]  0.045649027 -0.080871901 -0.0945502740  0.273659892
## [19,]  0.025316848 -0.025270567 -0.0142625660  0.025950336
## [20,]  0.022957353  0.144059196 -0.1591118124  0.153376638
## [21,] -0.087036615  0.109816923 -0.0771418170  0.076313723
## [22,]  0.017025273  0.076777911 -0.0447100608 -0.153298320
## [23,]  0.029254658 -0.033021559 -0.0005409599 -0.126929178
## [24,]  0.056755888  0.018508890 -0.0634091144  0.008381854
## [25,]  0.070401953  0.016260855 -0.0243336932 -0.114076466
## [26,]  0.058533627 -0.108279878  0.0670068985 -0.129152732
## [27,]  0.059995614 -0.069829167  0.0624175116 -0.081144863
## [28,]  0.121853654 -0.030329435 -0.0302579315 -0.142989492
## [29,]  0.058100866  0.162485783 -0.0285066185  0.091444271
```

```
## [30,]  0.058622927  0.137786608 -0.0629854838 -0.014564812
## [31,]  0.483537153 -0.116098140  0.1772026852  0.098198807
## [32,]  0.441664855 -0.110969766  0.1545669164  0.102187623
## [33,] -0.096519688 -0.103146424 -0.0093698979 -0.024145582
## [34,] -0.167586946 -0.014984990 -0.0398857731  0.055171767
## [35,] -0.238187225 -0.061120717  0.0028612825 -0.086099024
## [36,] -0.147036717 -0.026705118 -0.0284705346  0.065567779
## [37,] -0.244899769 -0.048122161  0.1026164675 -0.135572622
## [38,] -0.189739489 -0.127976819  0.0368373132 -0.156501022
## [39,] -0.135524337  0.114424207 -0.0182543443 -0.034176195
## [40,] -0.144479492  0.009553292 -0.0383817762  0.049653626
```

```
loadings2rot1<-print(varimax(loadings2)$loadings,cutoff=.35)
```

```
##
## Loadings:
##       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]
##  [1,]                                                   0.713
##  [2,] -0.635
##  [3,] -0.662
##  [4,]                                           -0.574
##  [5,] -0.549
##  [6,] -0.554
##  [7,]  0.500
##  [8,]
##  [9,]                                                          0.368
## [10,]
## [11,]                              0.636
## [12,]                              0.729
## [13,]                              0.460
## [14,]                                                   0.717
## [15,]                                           -0.620
## [16,]                             -0.739
## [17,]                             -0.360         0.505
## [18,]                             -0.650
## [19,]                             -0.688
## [20,]                                                          0.371
## [21,]                      -0.466
## [22,]                      -0.625
## [23,]                      -0.627
## [24,]                      -0.597
## [25,]               0.649
## [26,]               0.754
## [27,]               0.732
## [28,]               0.635
## [29,]               0.362
## [30,]               0.578
## [31,]                                     0.727
```

```
## [32,]        -0.375                        0.730
## [33,]        -0.609
## [34,]        -0.711
## [35,]        -0.686
## [36,]        -0.588
## [37,]        -0.582
## [38,]        -0.581
## [39,]        -0.631
## [40,]        -0.651
##
##                 [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
## SS loadings    2.508 3.963 3.089 1.957 3.511 1.279 1.401 1.402 0.995
## Proportion Var 0.063 0.099 0.077 0.049 0.088 0.032 0.035 0.035 0.025
## Cumulative Var 0.063 0.162 0.239 0.288 0.376 0.408 0.443 0.478 0.503
```

```
loadings2rot<-print(varimax(loadings2)$loadings,cutoff=.35,sort=T)
```

```
##
## Loadings:
##       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]
##  [1,] -0.635
##  [2,] -0.662
##  [3,] -0.549
##  [4,] -0.554
##  [5,]        -0.609
##  [6,]        -0.711
##  [7,]        -0.686
##  [8,]        -0.588
##  [9,]        -0.582
## [10,]        -0.581
## [11,]        -0.631
## [12,]        -0.651
## [13,]                0.649
## [14,]                0.754
## [15,]                0.732
## [16,]                0.635
## [17,]                0.578
## [18,]                      -0.625
## [19,]                      -0.627
## [20,]                      -0.597
## [21,]                              0.636
## [22,]                              0.729
## [23,]                             -0.739
## [24,]                             -0.650
## [25,]                             -0.688
## [26,]                                     0.727
## [27,]        -0.375                       0.730
## [28,]                                           -0.574
```

```
## [29,]                                           -0.620
## [30,]                            -0.360         0.505
## [31,]                                                  0.713
## [32,]                                                  0.717
## [33,]  0.500
## [34,]
## [35,]                                                         0.368
## [36,]
## [37,]                             0.460
## [38,]                                                         0.371
## [39,]                      -0.466
## [40,]                0.362
##
##                 [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
## SS loadings    2.508 3.963 3.089 1.957 3.511 1.279 1.401 1.402 0.995
## Proportion Var 0.063 0.099 0.077 0.049 0.088 0.032 0.035 0.035 0.025
##
```

We can see a similar pattern to that of the principal component method above. The variance is not as well explained using this method, but when we perform the next step (using iterations of the above method), it will become a better representation of the data. Again, we can see similar relationships between the first five factors and the variables, but it is not all encompassing.

Iterated Principal Factors Method

We will now use a similar method to above, but iterate it each time and update it with the new initial squared loadings. That is, when we calculate the new hi2 values, we do the process again until it converges. For completeness, I perform the process 100 times below.

```
hinit2<-1- 1/diag(solve(R))
for(i in 1:100){
  New2<-R-diag((1-hinit2),nrow=40)
  eigenvectors3<-eigen(New2)$vectors[,1:9]
  loadings3<-eigenvectors3%*%(diag(1,nrow=9)*(eigen(New2)$values[1:9])^.5)
  loading31<-as.matrix(loadings3[,1],nrow=40);loading32<-
as.matrix(loadings3[,2],nrow=40);loading33<-
as.matrix(loadings3[,3],nrow=40);loading34<-
as.matrix(loadings3[,4],nrow=40);loading35<-
as.matrix(loadings3[,5],nrow=40);loading36<-
as.matrix(loadings3[,6],nrow=40);loading37<-
as.matrix(loadings3[,7],nrow=40);loading38<-
as.matrix(loadings3[,8],nrow=40);loading39<-as.matrix(loadings3[,9],nrow=40)
  hinit2<-
as.numeric(loading31^2+loading32^2+loading33^2+loading34^2+loading35^2+loadin
g36^2+loading37^2+loading38^2+loading39^2,ncol=40)
}
loadings3
```

```
##             [,1]         [,2]         [,3]         [,4]         [,5]
##  [1,] -0.6336865  0.129510159  0.342334002 -0.038023630 -0.298855098
##  [2,] -0.6502007 -0.125780976  0.094612300  0.224797379 -0.119888358
##  [3,] -0.6291168 -0.190720725  0.040105304  0.241398639 -0.146635370
##  [4,] -0.5908644  0.002653042  0.226497992  0.040334761 -0.168819189
##  [5,] -0.5403052 -0.069776558 -0.046541079  0.088730496 -0.184540364
##  [6,] -0.5383186 -0.187930368 -0.027921243  0.127559024 -0.182887463
##  [7,]  0.5427781 -0.094736535  0.006660377 -0.381431291  0.087614276
##  [8,]  0.3964542 -0.078623909  0.051902429 -0.326084495  0.044529009
##  [9,]  0.3450905 -0.102407731  0.114416779 -0.188480115  0.024731098
## [10,]  0.2642319 -0.161843476  0.115746565 -0.166040194  0.068724964
## [11,] -0.5066480  0.260520630  0.262478846 -0.290267019  0.080632270
## [12,] -0.6545789  0.074217208  0.290576461 -0.232408027  0.333829685
## [13,] -0.5774633  0.071893130  0.217354904 -0.150535446  0.064325771
## [14,] -0.6108433  0.105405443  0.349790991 -0.163838651 -0.300910254
## [15,] -0.6225742  0.015818600  0.276931601 -0.112348412 -0.156714309
## [16,]  0.5917724 -0.235014489 -0.255368899  0.006650969 -0.424432473
## [17,]  0.5279568 -0.274680621 -0.187219522 -0.101549741 -0.006358423
## [18,]  0.4940006 -0.403112322 -0.229288093  0.053826480 -0.149052853
## [19,]  0.5679135 -0.227968636 -0.220179480 -0.057163025 -0.401467891
## [20,]  0.4988083 -0.198023532 -0.098685578 -0.087517556  0.037379190
## [21,] -0.3811688  0.142742173 -0.307097139 -0.296217542 -0.089586852
## [22,] -0.3269154  0.136263759 -0.233410980 -0.474756404 -0.138868985
```

```
## [23,] -0.3203851 -0.030166568 -0.245768083 -0.487202590 -0.205275498
## [24,] -0.3520775  0.095211884 -0.336299557 -0.428447845 -0.149642422
## [25,]  0.3651857 -0.412331552  0.368944213 -0.117631028 -0.026511108
## [26,]  0.3703239 -0.479742202  0.476104604 -0.096721625 -0.057825485
## [27,]  0.3699674 -0.421161776  0.480916968 -0.124121065 -0.078094691
## [28,]  0.2463836 -0.441672543  0.339399682 -0.114323666 -0.089602239
## [29,]  0.1920995 -0.302887469  0.214310995  0.008630304 -0.010576560
## [30,]  0.3177906 -0.361924210  0.349591276 -0.146596477 -0.020838225
## [31,] -0.4021482 -0.351959582 -0.201011400 -0.176729613  0.263077182
## [32,] -0.4265001 -0.447290807 -0.237762488 -0.201792383  0.345792026
## [33,] -0.3041925 -0.513476732 -0.171039751 -0.023635033  0.034697829
## [34,] -0.4423233 -0.542324765 -0.120056095  0.074356156  0.095296276
## [35,] -0.3607007 -0.500592080 -0.174513283  0.046947376  0.037358826
## [36,] -0.4350645 -0.446475795  0.030254396 -0.010627044  0.101759433
## [37,] -0.4599254 -0.329350786 -0.137864380  0.028873337 -0.110769493
## [38,] -0.4702438 -0.336331662 -0.199380424 -0.044648421 -0.119115633
## [39,] -0.3655473 -0.514596691 -0.078419787  0.097432633  0.093421477
## [40,] -0.3888139 -0.512946164 -0.095166329  0.169291414  0.033699196
##               [,6]         [,7]          [,8]         [,9]
##  [1,] -0.106461077  0.004012698  0.4158842813 -0.135351810
##  [2,]  0.183656923 -0.219213531 -0.0853302346 -0.005595223
##  [3,]  0.173629622 -0.280919773 -0.1084499357 -0.062148373
##  [4,]  0.045883497  0.264423699 -0.2146298828 -0.139428187
##  [5,]  0.153429789 -0.241690414 -0.0715201038 -0.043392862
##  [6,]  0.135235784 -0.224915024 -0.0776985624  0.061374907
##  [7,] -0.135960537  0.081417002 -0.0316562359 -0.128293688
##  [8,] -0.049494223 -0.071142504 -0.0747379454 -0.122691640
##  [9,] -0.224110839 -0.036705193 -0.0697731205 -0.189402920
## [10,] -0.112025835 -0.220779261  0.0120029682 -0.083149450
## [11,] -0.054378180 -0.154096849  0.0042190639  0.110051557
## [12,] -0.097500210 -0.134497695 -0.0869242753 -0.052456320
## [13,]  0.004985414  0.017801279 -0.0225716598 -0.066759174
## [14,] -0.103460721  0.045917678  0.3512154313 -0.173798335
## [15,]  0.086461623  0.283676355 -0.3076787991 -0.195371434
## [16,]  0.116896158  0.026640961  0.0407768581 -0.024693231
## [17,] -0.131489718 -0.307936400  0.2109326646 -0.053543527
## [18,]  0.058937557  0.079736682 -0.0232532548 -0.217180572
## [19,]  0.088122223  0.046636312  0.0008630064 -0.070659460
## [20,]  0.007864417 -0.112906050 -0.1261029356 -0.214195713
## [21,] -0.090118823 -0.097821370 -0.0685454038 -0.104261189
## [22,]  0.016461123 -0.091602431 -0.0939518707  0.145870493
## [23,]  0.037694710  0.008704829 -0.0416043160  0.183271179
## [24,]  0.056262149 -0.029658115 -0.0706184986  0.004097048
## [25,]  0.073337166 -0.028056229 -0.0530975791  0.108645037
## [26,]  0.071261480  0.081462466  0.0360994696  0.185563020
## [27,]  0.074598867  0.047942227  0.0369657600  0.117259474
## [28,]  0.133700006  0.015520059 -0.0628408544  0.141265238
## [29,]  0.051437400 -0.149282018 -0.0246684699 -0.103773907
## [30,]  0.053217935 -0.144213470 -0.0810473995 -0.010744364
## [31,]  0.444156811  0.088841988  0.1869857115 -0.057679182
```

```
## [32,]  0.460313960  0.104878412  0.2129963667 -0.087767881
## [33,] -0.114633165  0.097987560 -0.0090573562  0.030682748
## [34,] -0.199403042  0.027929674 -0.0202432667 -0.069660173
## [35,] -0.267136034  0.074407466 -0.0093759739  0.080036468
## [36,] -0.174325196  0.029505768 -0.0120425589 -0.050011012
## [37,] -0.226812362  0.044696510  0.0707773605  0.146141059
## [38,] -0.177227803  0.122626669  0.0060156884  0.175156554
## [39,] -0.160987378 -0.093535945 -0.0298388671 -0.005968257
## [40,] -0.152326147  0.016936469 -0.0188479397 -0.086933077
```

```r
hinit2
```

```
##  [1] 0.8289141 0.6015374 0.6382330 0.5680873 0.4298459 0.4542829 0.4993689
##  [8] 0.3025154 0.2711171 0.2100527 0.5230468 0.7217756 0.4179823 0.7903544
## [15] 0.7225057 0.6674738 0.5590656 0.5417629 0.6023525 0.3814098 0.3890036
## [22] 0.4633676 0.4802789 0.4611304 0.4748262 0.6541179 0.5900188 0.4340915
## [29] 0.2110671 0.4064326 0.6699060 0.8747538 0.4109744 0.5645937 0.4981439
## [36] 0.4339110 0.4319218 0.4673458 0.4583968 0.4845446
```

```r
loadings3rot1<-print(varimax(loadings3)$loadings,cutoff=.3)
```

```
## 
## Loadings:
##       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]  
##  [1,]                              0.328                0.771       
##  [2,] -0.639                                                        
##  [3,] -0.688 -0.300                                                 
##  [4,]                                            0.593              
##  [5,] -0.551                                                        
##  [6,] -0.555                                                        
##  [7,]  0.487                                                  -0.362
##  [8,]                                                         -0.357
##  [9,]                                                         -0.376
## [10,]                                                         -0.326
## [11,]                              0.605                            
## [12,]                              0.755                            
## [13,]                              0.463                            
## [14,]                              0.324                0.749       
## [15,]                              0.300         0.703              
## [16,]                             -0.757                            
## [17,]                             -0.361        -0.509        -0.340
## [18,]                             -0.624                            
## [19,]                             -0.706                            
## [20,]                             -0.327                      -0.398
## [21,]               -0.350 -0.455                                   
## [22,]                      -0.647                                   
## [23,]                      -0.662                                   
## [24,]                      -0.606                                   
## [25,]                0.653                                          
## [26,]                0.771                                          
## [27,]                0.734                                          
```

```
## [28,]                 0.641
## [29,]                 0.347
## [30,]                 0.573
## [31,]                                        0.730
## [32,]         -0.367                         0.834
## [33,]         -0.606
## [34,]         -0.717
## [35,]         -0.698
## [36,]         -0.592
## [37,]         -0.575
## [38,]         -0.578
## [39,]         -0.628
## [40,]         -0.649
##
##                 [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
## SS loadings    2.456 3.967 3.049 1.971 3.529 1.417 1.485 1.511 1.206
## Proportion Var 0.061 0.099 0.076 0.049 0.088 0.035 0.037 0.038 0.030
## Cumulative Var 0.061 0.161 0.237 0.286 0.374 0.410 0.447 0.485 0.515
```

```
loadings3rot<-print(varimax(loadings3)$loadings,cutoff=.3,sort=T)
```

```
##
## Loadings:
##       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]
##  [1,] -0.639
##  [2,] -0.688 -0.300
##  [3,] -0.551
##  [4,] -0.555
##  [5,]        -0.606
##  [6,]        -0.717
##  [7,]        -0.698
##  [8,]        -0.592
##  [9,]        -0.575
## [10,]        -0.578
## [11,]        -0.628
## [12,]        -0.649
## [13,]                0.653
## [14,]                0.771
## [15,]                0.734
## [16,]                0.641
## [17,]                0.573
## [18,]                      -0.647
## [19,]                      -0.662
## [20,]                      -0.606
## [21,]                              0.605
## [22,]                              0.755
## [23,]                             -0.757
## [24,]                             -0.624
## [25,]                             -0.706
## [26,]                                     0.730
```

```
## [27,]        -0.367                       0.834
## [28,]                                            0.593
## [29,]                               0.300        0.703
## [30,]                              -0.361       -0.509        -0.340
## [31,]                               0.328               0.771
## [32,]                               0.324               0.749
## [33,]   0.487                                                  -0.362
## [34,]                                                          -0.357
## [35,]                                                          -0.376
## [36,]                                                          -0.326
## [37,]                               0.463
## [38,]                              -0.327                     -0.398
## [39,]               -0.350 -0.455
## [40,]                0.347
##
##                   [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
## SS loadings    2.456 3.967 3.049 1.971 3.529 1.417 1.485 1.511 1.206
## Proportion Var 0.061 0.099 0.076 0.049 0.088 0.035 0.037 0.038 0.030
## Cumulative Var 0.061 0.161 0.237 0.286 0.374 0.410 0.447 0.485 0.515
```

Again, we can see a similar pattern to the two above methods. The iterated method also created loadings which are closer to the principal component method. There are a few differences; for instance, in the iterated method we see that the third variable is part of both the 1st and 2nd factor, which wasn't present in the principal component method nor the principal factor method. Still, the analysis from before holds: much of the expected separation *is* contained in the factors (or some combination of them), but there are other correlations which those four factors cannot account for.

I will use the rotated matrix above to make some considerations about the level of effectiveness of the analysis. We know that the main goal of factor analysis is to acquire factors for which each variable can be loaded into only one factor. If we can do this, we have essentially reached our goal – we have separated the variables completely into a smaller number of factors which accounts for the covariates between them.

However, in practice, this is very difficult to achieve. The highlighted sections in the loading matrix above shows some instances where the variables correlate strongly with more than one factor at once. This indicates that factor analysis is not doing a very good job at separating the factors with the number of factors we have used.

SOME CONSIDERATIONS

In an ideal situation, we would use a very small value of m relative to the amount of variables p. Although $m = 9$ performed decently well, it is quite a large number of factors. Even with nine factors, we see that the factor analysis was only able to account for around 60% of the covariation of the original variables. A fundamental issue with factor analysis is that the correlation matrix **R** contains both structure and error, and factor analysis is not able to separate the two. Thus, the original assumptions of no relationships between the errors and the factors are too optimistic, as this rarely happens in practice.

As a result, the loadings above are quite difficult to interpret. It is difficult to say what the exact factors are in plain English. Here, we can see some of the difficulty in working with factor analysis in practice – it is often questioned if these factors truly exist.

CONCLUSIONS

Below, I have reproduced the loadings matrix of the principal factor method in order to point out some interpretations.

```
Loadings:
##       [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]
##  [1,]                                                   0.713
##  [2,] -0.635
##  [3,] -0.662
##  [4,]                                           -0.574
##  [5,] -0.549
##  [6,] -0.554
##  [7,]  0.500
##  [8,]
##  [9,]                                                          0.368
## [10,]
## [11,]                              0.636
## [12,]                              0.729
## [13,]                              0.460
## [14,]                                                   0.717
## [15,]                                           -0.620
## [16,]                             -0.739
## [17,]                             -0.360         0.505
## [18,]                             -0.650
## [19,]                             -0.688
## [20,]                                                          0.371
## [21,]                      -0.466
## [22,]                      -0.625
## [23,]                      -0.627
## [24,]                      -0.597
## [25,]                0.649
## [26,]                0.754
## [27,]                0.732
## [28,]                0.635
## [29,]                0.362
## [30,]                0.578
## [31,]                                     0.727
## [32,]        -0.375                       0.730
## [33,]        -0.609
## [34,]        -0.711
## [35,]        -0.686
## [36,]        -0.588
```

```
## [37,]         -0.582
## [38,]         -0.581
## [39,]         -0.631
## [40,]         -0.651
##
##                 [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9]
## SS loadings    2.508 3.963 3.089 1.957 3.511 1.279 1.401 1.402 0.995
## Proportion Var 0.063 0.099 0.077 0.049 0.088 0.032 0.035 0.035 0.025
## Cumulative Var 0.063 0.162 0.239 0.288 0.376 0.408 0.443 0.478 0.503
```

We know that the original variables were constructed considering four groups (assertiveness, social confidence, adventurousness, and dominance). These were the intended characteristics that the statements were testing for. Above, we can see how factors 1-5 capture a lot of the variables in each set. So, we could say that factor 1 represents assertiveness, factor 2 represents dominance, factor 3 and 4 represent adventurousness, and factor 5 represents social confidence.

However, this would not be so accurate, as there are other factors to consider as well. We can see that factors 6-9 have members from all the different groups. This suggests that the relationship between the variables was not so simple as we once thought. Additionally, variable 10 failed to have a loading greater than .3 into *any* of the factors, which suggests that even 9 factors are inadequate to capture the variation in the data.

In this example, which seemed relatively straightforward, we see that problems emerged quite quickly. What, for instance, is factor six? And why are factors 3 and 4 separated? It is not so clear that our prior ideas about the factors are accurate.

One thing is clear, though. The four factors used to create the questionnaire are insufficient to truly describe the structure of the data. To say that these statements measure these factors would be jumping to conclusions too quickly. Doing so is, in a way, oversimplifying the data.

In conclusion, we have learned that this dataset cannot be so simply separated into the four purported factors. In fact, even separating the variables into 9 factors can only account for 60% of the variation of the variables. We can see that while factor analysis is a powerful technique, it is very dependent on the dataset which it is performed on. The self-checking property of the assumptions of factor analysis ensures that only models which fit the assumptions will produce results which separate the variables adequately to the different factors.