# PATTERN MATRIX

Each column is a factor
(these are named according to the extraction method)

Each row is an item (one of our variables)

```
> fa(eg_data, nfactors=2, rotate = "oblimin", fm="ml")
Factor Analysis using method =  ml
Call: fa(r = eg_data, nfactors = 2, rotate = "oblimin", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
         ML1   ML2   h2   u2 com
item_1  0.60  0.03 0.37 0.63 1.0
item_2  0.52  0.09 0.31 0.69 1.1
item_3  0.67 -0.05 0.43 0.57 1.0
item_4  0.62 -0.08 0.36 0.64 1.0
item_5  0.42  0.35 0.40 0.60 1.9
item_6 -0.03  0.55 0.30 0.70 1.0
item_7 -0.05  0.69 0.45 0.55 1.0
item_8  0.12  0.44 0.24 0.76 1.2
item_9  0.11  0.34 0.15 0.85 1.2

                       ML1  ML2
SS loadings           1.73 1.29
Proportion Var        0.19 0.14
Cumulative Var        0.19 0.33
Proportion Explained  0.57 0.43
Cumulative Proportion 0.57 1.00

 With factor correlations of 
     ML1  ML2
ML1 1.00 0.34
ML2 0.34 1.00
```

# PATTERN MATRIX

Each column is a factor
(these are named according to the extraction method)

Each row is an item
(one of our variables)

```
> fa(eg_data, nfactors=2, rotate = "oblimin", fm="ml")
Factor Analysis using method =  ml
Call: fa(r = eg_data, nfactors = 2, rotate = "oblimin", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
        ML1   ML2   h2   u2 com
item_1  0.60  0.03 0.37 0.63 1.0
item_2  0.52  0.09 0.31 0.69 1.1
item_3  0.67 -0.05 0.43 0.57 1.0
item_4  0.62 -0.08 0.36 0.64 1.0
item_5  0.42  0.35 0.40 0.60 1.9
item_6 -0.03  0.55 0.30 0.70 1.0
item_7 -0.05  0.69 0.45 0.55 1.0
item_8  0.12  0.44 0.24 0.76 1.2
item_9  0.11  0.34 0.15 0.85 1.2

                       ML1  ML2
SS loadings           1.73 1.29
Proportion Var        0.19 0.14
Cumulative Var        0.19 0.33
Proportion Explained  0.57 0.43
Cumulative Proportion 0.57 1.00

 With factor correlations of
     ML1  ML2
ML1 1.00 0.34
ML2 0.34 1.00

...
```

Loadings show the association between each item and each factor.

With an oblique rotation, the pattern matrix shows standardised regression coefficients:

$item_i$ = $loading_{F1,i}$*Factor1 + $loading_{F2,i}$*Factor2 + … + $uniqueness_i$

With no rotation or an orthogonal rotation these are correlation coefficients, and the pattern matrix is identical to the structure matrix*

A squared loading reflects the proportion of variance in an item that is uniquely explained by a factor.

e.g., $-0.60^2 = 0.36$

36% of the variance in item 1 is explained by Factor 1

*When an oblique rotation is used, factors can be correlated. Therefore to get the *unique* association between item and factors, we need the regression weights from a model of `item ~ factor1 + factor2`.
If the factors are **not** correlated (by definition they are uncorrelated when no rotation or an orthogonal rotation is used), then these regression weights are just the same as the correlations `cor(item, factor1)` and `cor(item, factor2)`.

# COMMUNALITIES

```
> fa(eg_data, nfactors=2, rotate = "oblimin", fm="ml")
Factor Analysis using method =  ml
Call: fa(r = eg_data, nfactors = 2, rotate = "oblimin", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
         ML1   ML2   h2   u2 com
item_1  0.60  0.03 0.37 0.63 1.0
item_2  0.52  0.09 0.31 0.69 1.1
item_3  0.67 -0.05 0.43 0.57 1.0
item_4  0.62 -0.08 0.36 0.64 1.0
item_5  0.42  0.35 0.40 0.60 1.9
item_6 -0.03  0.55 0.30 0.70 1.0
item_7 -0.05  0.69 0.45 0.55 1.0
item_8  0.12  0.44 0.24 0.76 1.2
item_9  0.11  0.34 0.15 0.85 1.2

                       ML1  ML2
SS loadings           1.73 1.29
Proportion Var        0.19 0.14
Cumulative Var        0.19 0.33
Proportion Explained  0.57 0.43
Cumulative Proportion 0.57 1.00

 With factor correlations of 
     ML1  ML2
ML1 1.00 0.34
ML2 0.34 1.00

...
```

Square the **correlations** each item and add them up, and you get the proportion of variance in an item that is explained by all the factors. This is the “communality”.

The correlations are in the *structure matrix*, but this is the *pattern matrix*. Because the factor correlations here are low, we can just do this calculation on the loadings here

e.g., $-0.03^2 + -0.55^2$ = approx 0.30

30% of the variance in item 6 is explained by this 2 Factor solution

# UNIQUENESS

```
> fa(eg_data, nfactors=2, rotate = "oblimin", fm="ml")
Factor Analysis using method =  ml
Call: fa(r = eg_data, nfactors = 2, rotate = "oblimin", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
         ML1   ML2   h2   u2 com
item_1  0.60  0.03 0.37 0.63 1.0
item_2  0.52  0.09 0.31 0.69 1.1
item_3  0.67 -0.05 0.43 0.57 1.0
item_4  0.62 -0.08 0.36 0.64 1.0
item_5  0.42  0.35 0.40 0.60 1.9
item_6 -0.03  0.55 0.30 0.70 1.0
item_7 -0.05  0.69 0.45 0.55 1.0
item_8  0.12  0.44 0.24 0.76 1.2
item_9  0.11  0.34 0.15 0.85 1.2

                       ML1  ML2
SS loadings           1.73 1.29
Proportion Var        0.19 0.14
Cumulative Var        0.19 0.33
Proportion Explained  0.57 0.43
Cumulative Proportion 0.57 1.00

 With factor correlations of
     ML1  ML2
ML1 1.00 0.34
ML2 0.34 1.00

...
```

The proportion of variance in each item that is left *unexplained* by the factors is 1 minus the communality.

e.g., 1 - 0.30 = 0.70

70% of the variance in item 6 is left unexplained

# COMPLEXITY

```
> fa(eg_data, nfactors=2, rotate = "oblimin", fm="ml")
Factor Analysis using method =  ml
Call: fa(r = eg_data, nfactors = 2, rotate = "oblimin", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
         ML1   ML2   h2   u2 com
item_1  0.60  0.03 0.37 0.63 1.0
item_2  0.52  0.09 0.31 0.69 1.1
item_3  0.67 -0.05 0.43 0.57 1.0
item_4  0.62 -0.08 0.36 0.64 1.0
item_5  0.42  0.35 0.40 0.60 1.9
item_6 -0.03  0.55 0.30 0.70 1.0
item_7 -0.05  0.69 0.45 0.55 1.0
item_8  0.12  0.44 0.24 0.76 1.2
item_9  0.11  0.34 0.15 0.85 1.2

                       ML1  ML2
SS loadings           1.73 1.29
Proportion Var        0.19 0.14
Cumulative Var        0.19 0.33
Proportion Explained  0.57 0.43
Cumulative Proportion 0.57 1.00

 With factor correlations of 
     ML1  ML2
ML1 1.00 0.34
ML2 0.34 1.00

...
```

The extent to which a given item loads on to a single factor vs onto multiple factors is termed ‘complexity’.
It equals 1 if an item loads only on one factor, 2 if it loads evenly on 2 factors, and so on.

# SS Loadings & "Variance Accounted For"

```
> fa(eg_data, nfactors=2, rotate = "oblimin", fm="ml")
Factor Analysis using method =  ml
Call: fa(r = eg_data, nfactors = 2, rotate = "oblimin", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
         ML1   ML2   h2   u2 com
item_1  0.60  0.03 0.37 0.63 1.0
item_2  0.52  0.09 0.31 0.69 1.1
item_3  0.67 -0.05 0.43 0.57 1.0
item_4  0.62 -0.08 0.36 0.64 1.0
item_5  0.42  0.35 0.40 0.60 1.9
item_6 -0.03  0.55 0.30 0.70 1.0
item_7 -0.05  0.69 0.45 0.55 1.0
item_8  0.12  0.44 0.24 0.76 1.2
item_9  0.11  0.34 0.15 0.85 1.2

                       ML1  ML2
SS loadings           1.73 1.29
Proportion Var        0.19 0.14
Cumulative Var        0.19 0.33
Proportion Explained  0.57 0.43
Cumulative Proportion 0.57 1.00

 With factor correlations of
     ML1  ML2
ML1 1.00 0.34
ML2 0.34 1.00

...
```

Square all the loadings for each factor and add them up. This gives you the "SS loadings".

These are the same as the eigenvalues unless an oblique rotation is used. As the variance in each item is scaled to be 1, the total variance in the data is equal to the number of items.

# SS Loadings & "Variance Accounted For"

```
> fa(eg_data, nfactors=2, rotate = "oblimin", fm="ml")
Factor Analysis using method =  ml
Call: fa(r = eg_data, nfactors = 2, rotate = "oblimin", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
         ML1   ML2   h2   u2 com
item_1  0.60  0.03 0.37 0.63 1.0
item_2  0.52  0.09 0.31 0.69 1.1
item_3  0.67 -0.05 0.43 0.57 1.0
item_4  0.62 -0.08 0.36 0.64 1.0
item_5  0.42  0.35 0.40 0.60 1.9
item_6 -0.03  0.55 0.30 0.70 1.0
item_7 -0.05  0.69 0.45 0.55 1.0
item_8  0.12  0.44 0.24 0.76 1.2
item_9  0.11  0.34 0.15 0.85 1.2

                       ML1  ML2
SS loadings           1.73 1.29
Proportion Var        0.19 0.14
Cumulative Var        0.19 0.33
Proportion Explained  0.57 0.43
Cumulative Proportion 0.57 1.00

 With factor correlations of
     ML1  ML2
ML1 1.00 0.34
ML2 0.34 1.00

...
```

SS loadings divided by number of items gives the proportion of variance in the data explained by each factor

e.g., 1.73/9 = 0.19

19% of the variance is explained by Factor 1

# SS Loadings & "Variance Accounted For"

```
> fa(eg_data, nfactors=2, rotate = "oblimin", fm="ml")
Factor Analysis using method =  ml
Call: fa(r = eg_data, nfactors = 2, rotate = "oblimin", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
         ML1   ML2   h2   u2 com
item_1  0.60  0.03 0.37 0.63 1.0
item_2  0.52  0.09 0.31 0.69 1.1
item_3  0.67 -0.05 0.43 0.57 1.0
item_4  0.62 -0.08 0.36 0.64 1.0
item_5  0.42  0.35 0.40 0.60 1.9
item_6 -0.03  0.55 0.30 0.70 1.0
item_7 -0.05  0.69 0.45 0.55 1.0
item_8  0.12  0.44 0.24 0.76 1.2
item_9  0.11  0.34 0.15 0.85 1.2

                       ML1  ML2
SS loadings           1.73 1.29
Proportion Var        0.19 0.14
Cumulative Var        0.19 0.33
Proportion Explained  0.57 0.43
Cumulative Proportion 0.57 1.00

 With factor correlations of
     ML1  ML2
ML1 1.00 0.34
ML2 0.34 1.00

...
```

Taking each factor sequentially, we can calculate the cumulative variance explained.

e.g., 0.19+0.14 = 0.33

# SS Loadings & "Variance Accounted For"

```
> fa(eg_data, nfactors=2, rotate = "oblimin", fm="ml")
Factor Analysis using method =  ml
Call: fa(r = eg_data, nfactors = 2, rotate = "oblimin", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
         ML1   ML2   h2   u2 com
item_1  0.60  0.03 0.37 0.63 1.0
item_2  0.52  0.09 0.31 0.69 1.1
item_3  0.67 -0.05 0.43 0.57 1.0
item_4  0.62 -0.08 0.36 0.64 1.0
item_5  0.42  0.35 0.40 0.60 1.9
item_6 -0.03  0.55 0.30 0.70 1.0
item_7 -0.05  0.69 0.45 0.55 1.0
item_8  0.12  0.44 0.24 0.76 1.2
item_9  0.11  0.34 0.15 0.85 1.2

                       ML1  ML2
SS loadings           1.73 1.29
Proportion Var        0.19 0.14
Cumulative Var        0.19 0.33
Proportion Explained  0.57 0.43
Cumulative Proportion 0.57 1.00

 With factor correlations of
     ML1  ML2
ML1 1.00 0.34
ML2 0.34 1.00

...
```

Out of the total variance explained by all factors, we can calculate the proportion of this that is explained by each factor.

e.g., 0.19/0.33 = 0.57

We can see this cumulatively too

# FACTOR CORRELATIONS

```
> fa(eg_data, nfactors=2, rotate = "oblimin", fm="ml")
Factor Analysis using method =  ml
Call: fa(r = eg_data, nfactors = 2, rotate = "oblimin", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
         ML1   ML2   h2   u2 com
item_1  0.60  0.03 0.37 0.63 1.0
item_2  0.52  0.09 0.31 0.69 1.1
item_3  0.67 -0.05 0.43 0.57 1.0
item_4  0.62 -0.08 0.36 0.64 1.0
item_5  0.42  0.35 0.40 0.60 1.9
item_6 -0.03  0.55 0.30 0.70 1.0
item_7 -0.05  0.69 0.45 0.55 1.0
item_8  0.12  0.44 0.24 0.76 1.2
item_9  0.11  0.34 0.15 0.85 1.2

                       ML1  ML2
SS loadings           1.73 1.29
Proportion Var        0.19 0.14
Cumulative Var        0.19 0.33
Proportion Explained  0.57 0.43
Cumulative Proportion 0.57 1.00

 With factor correlations of 
     ML1  ML2
ML1 1.00 0.34
ML2 0.34 1.00

...
```

Correlation matrix for the factors. This will depend on whether or not a correlation is estimated (i.e. whether an oblique rotation is used). Shows how related the factors are to one another.

# OPTIONAL EXTRA: GOODNESS OF FIT TEST

Mean of the item complexities column

the “null model” is a model that assumes no correlation structure.
df = p * (p-1)/2
p = number of items

```
...
...
Mean item complexity =  1.2
Test of the hypothesis that 2 factors are sufficient.

df null model =  36  with the objective function =  1.46 with Chi Square =  578
df of  the model are 19  and the objective function was  0.04

The root mean square of the residuals (RMSR) is  0.02
The df corrected root mean square of the residuals is  0.03

The harmonic n.obs is  400 with the empirical chi square  15.6  with prob <  0.68
The total n.obs was  400  with Likelihood Chi Square =  17.6  with prob <  0.55

Tucker Lewis Index of factoring reliability =  1
RMSEA index =  0  and the 90 % confidence intervals are  0 0.04
BIC =  -96.2
Fit based upon off diagonal va
Measures of factor score adequ

Correlation of (regression) sc
Mul                          es w
Min                          ssib
```

our model
df = p * (p-1)/2 - p * nF + nF*(nF-1)/2
p = number of items
nF = number of factors

average number of non-missing observations for each pair of items

Total number of observations in the data

Chi-square ‘goodness of fit’ for our model (calculated two ways, see ?factor.stats ).

The set of parameters from our model *implies* a correlation matrix, and we have our *observed* matrix. The discrepancy between these is the residual correlation matrix, and it is on this that the Chi-Square statistic is based (think “observed minus expected”).

Lower Chi-Square values are better. A significant Chi-square value indicates possible under-extraction of factors.

# OPTIONAL EXTRA: FIT INDICES

All of these are essentially measure of how well (or how badly) the model fits to the observed data. They are more conventionally used in confirmatory factor analysis (CFA) and structural equation modelling (SEM), but are printed here too.

- We want RMSR to be low (it indicates the average size of the residual correlation)
- Tucker Lewis Index (TLI) compares the chi-square of our model to that of the null model (adjusted for the df). It ranges 0 to 1, and we want it to be high. Typical cut-offs used to indicate good fit are >0.9 or >0.95.
- RMSEA is a measure of how far our model is from a 'perfect model'. Lower is better, and typical cut-offs used to indicate good fit are <0.05, <0.08 or <0.1.
- BIC is only relevant for comparing models

```
The root mean square of the residuals (RMSR) is  0.02
The df corrected root mean square of the residuals is  0.03

The harmonic n.obs is  400 with the empirical chi square  15.6  with prob <  0.68
The total n.obs was  400  with Likelihood Chi Square =  17.6  with prob <  0.55

Tucker Lewis Index of factoring reliability =  1
RMSEA index =  0  and the 90 % confidence intervals are  0 0.04
BIC =  -96.2
Fit based upon off diagonal values = 0.99
Measures of factor score adequacy
                                                   ML1  ML2
Correlation of (regression) scores with factors   0.86 0.82
Multiple R square of scores with factors          0.74 0.67
Minimum correlation of possible factor scores     0.47 0.34
```

# OPTIONAL EXTRA: INDETERMINACY INDICES

"Factor score indeterminacy": there are an infinite number of pairs of factor loadings and factor score matrices which will fit the data equally well, and are thus indistinguishable by any numeric criteria, so there are infinite number of sets of factor scores that are consistent with a given set of loadings.

Multiple R square of scores with factors
"The multiple $R^2$ between the factors and factor score estimates, if they were to be found." (Grice, 2001). This is a little bit like the $R^2$ for a regression model of the items predicting the estimated factor score. If we could perfectly predict factor scores from items, then it would be 1.

Correlation of (regression) scores with factors
This just the square root of the multiple $R^2$.

Minimum correlation of possible factor scores
This is $2 \times R^2 - 1$, so essentially the $R^2$ but transformed to be between -1 and 1.

```
The total n.obs was  400  with Likelihood Chi Square =  17.6  with prob <  0.55

Tucker Lewis Index of factoring reliability =  1
RMSEA index =  0  and the 90 % confidence intervals are  0 0.04
BIC =  -96.2
Fit based upon off diagonal values = 0.99
Measures of factor score adequacy
                                                 ML1  ML2
Correlation of (regression) scores with factors   0.86 0.82
Multiple R square of scores with factors          0.74 0.67
Minimum correlation of possible factor scores     0.47 0.34
```

```
set.seed(533)
makeitems <- function(){
  S = runif(5,.4,2)
  f = runif(5,.4,.99)
  R = f %*% t(f)
  diag(R) = 1
  items = round(MASS::mvrnorm(400, mu = rnorm(5,3,.6), Sigma=diag(S)%*%R%*%diag(S)))
  apply(items, 2, function(x) pmin(7,pmax(1,x)))
}
eg_data = do.call(cbind,lapply(1:2, function(x) makeitems()))
eg_data[,5] <- round(rowMeans(eg_data[,c(5,10)]))
eg_data <- eg_data[,-10]
eg_data[,1] <- max(eg_data[,1]) - eg_data[,1] + 1
eg_data[,6] <- max(eg_data[,6]) - eg_data[,6] + 1
eg_data <- as.data.frame(eg_data)
names(eg_data) <- paste0("item_",1:9)


mm = fa(eg_data, nfactors=2, rotate = "oblimin", fm="ml")
mm

# tli
((1138.13/36) - (20.71/19)) /
  ((1138.13/36) - 1)

# rmsea
sqrt(20.71 - 19) /
  sqrt(19*(400-1))
```