# Factor Analysis

In this assignment, you'll be learning about the technique of *factor analysis*. Put concisely, in PCA, one tries to explain *all* of the variance in $p$-dimensional data with the $p$ principal components, whereas in factor analysis, one picks a *number of factors* $k < p$ and calculates those factors such that they explain as much of the *shared variance* in the data as they possibly can. Don't worry if this definition seems a little opaque to you right now – you'll first get some computational intuition by empirically comparing the results of PCA to factor analysis, and afterward we'll have a lecture covering some of the theoretical details.

Keep the following in mind while you work:

- The outputs of factor analysis, accessed with `fa()`, are similar to the outputs of `prcomp()`. In particular, if we have `p = prcomp(...)` and `f = fa(...)`, then `p$rotation` is analogous to `f$loadings` and `p$x` is analogous to `f$scores`.
- You should be consistently using `corrplot()` to visualize the loadings of principal components / factors.
- The two dataset analysis questions are *intentionally* unstructured and open-ended. Feel free to spend time exploring the data from multiple angles, reading about related material, and in general just trying any of the tools you've learned about.

Write up your analysis in an R Markdown file, including your interpretation of any results.

## PCA vs. factor analysis with simulated data

For reproducibility, place `set.seed(1)` at the top of your code.

### Part 1: PCA

- Make a `factors` data frame with 100 observations of 3 normally distributed variables `X`, `Y`, and `X`. Each variable should be drawn from the

standard normal distribution. The "factors" here have nothing to do with factors in R, instead representing latent variables.

- Write a function `noisyIndicators(feature, k, correlation)` that takes a vector `feature` and returns a data frame with `k` noisy proxies to the feature which are (1) correlated with the feature at the level of `correlation` and (2) differ from the feature by an error term drawn from the standard normal distribution.
- Make a dataframe `noisies` with 4 noisy proxies to `X` and 3 noisy proxies to `Y`.
- Plot the correlation matrix.
- Run PCA on the features and plot the correlations between the principal components and the `noisies` data frame. Also, plot the correlations between the principal components and the `factors` data frame.

You should see that the first two principal components pick up on the factors `X` and `Y`, but only imperfectly.

## Part 2: Orthogonal factor analysis

Factor analysis is an alternative to principal component analysis for dimensionality reduction. Here, one picks a natural number $k$ and assumes that each of the variables $V_i$ can be written as a sum of

$$V_i = a_1 F_1 + a_2 F_2 + \cdots + a_k F_k + \text{error}_i$$

where each $a_j$ is a constant that depends on $V_i$. Goodness of fit is measured by taking the correlation matrix of the errors $\text{error}_i$ and measuring how far it is from being the identity matrix with 1s down the diagonal (as usual) and 0s off of the diagonal.

The key difference from PCA here here is that the factors are supposed to explain as much of the correlations *between* the variables as possible, rather than as much *total* variance as possible: we don't try to pick up on the variables to the extent that they're not correlated with one another.

- Factor analysis is implemented in the R library `psych` as `fa()`. Run factor analysis on the data from the previous section with `nfactors = 2` and `rotate = "varimax"`, and compare the correlations between the modeled factors and the true factors `X` and `Y`. The modeled factors should be closer to the true factors than the principal components were.
- Generate 50 variables given by

```
X*runif(1) + Y*runif(2) + Z*runif(3) + 0.5*error
```

where `error` is normally distributed with standard deviation 1.

- Run principal component analysis on the 50 variables. Compute the correlations between the first 3 principal components and the variables `X`, `Y`, and `Z`.
- Now do factor analysis with `k = 3` and `rotate = "varimax"` and compute the correlations between the modeled factors and the true factors.

### Part 3: Oblique factor analysis:

In the last factor analysis exercise, we were using a type of factor analysis which is analogous to PCA in assuming the factors to be orthogonal (perpendicular). Here we explore oblique factor analysis, which allows for correlated factors. We will talk about what this is later in class.

- Set `W = 0.5*X + Y`.
- Examine the correlation between `W` and `Y`.
- Use `noisyIndicators()` to generate 10 noisy indicators associated with `X` and 4 noisy indicators associated with W, with correlations 0.8 in both cases.
- Plot the associated correlation matrix.
- Compare the results of using `fa()` with `nfactors = 2` and `rotate = "varimax"` and `fa()` with `nfactors = 2` and `rotate = "oblimin"` by looking at the correlations of the results with the noisy indicators, and with the true factors. The former fails to correctly identify the factors because it tries to force them to be orthogonal, whereas the latter correctly identifies them.

## Speed dating data

- Do factor analysis on the speed dating data activities. Try `nfactor = 1`, `2`, `3`, and `4`, both with `rotate = "varimax"` and with `rotate = "oblimin"`. Use `corrplot()` to visualize the factor loadings (which you can get with `$loadings`).

## Big Five personality data

- Download the BIG5 dataset here (posted 5/18/2014).

- Compare the results of PCA on the 50 questions with factor analysis using `nfactor = 5`. Visualize the loadings `corrplot()` and `$loadings`.
- Create a logistic regression predictive model of gender in terms of the 50 questions, then in terms of the 5 factors derived from them. Compare the coefficients of the two models. You can use regular `glm()` because there are so many examples that it's not necessary to worry about regularization.
- By focusing on the 10 questions corresponding to a trait, for each trait, can you replicate the subfactors of the Big 5 personality inventory traits given in The Items in the Big Five Aspects Scales? See also the original paper Between Facets and Domains: 10 Aspects of the Big Five?