Principal Components Analysis

Use the `prcomp(..., scale=TRUE)` function to explore the principal components of various datasets.

A couple things to keep in mind: Let `p = prcomp(df, scale=TRUE)` be the result of PCA run on a data frame `df`. Then:

- `p$rotation` gives a matrix with rows corresponding to columns of `df`, columns corresponding to principal components, and entries corresponding to the *loadings* of a particular column of `df` on a particular principal component. Put another way, each principal component is formed out of a linear combination of the variables of `df`, and the column corresponding to each principal column corresponds to the *coefficients* of that linear combination.
- `p$x` gives the *principal component scores* for each row of `df`, that is, the *actual value* of each principal component for every row in `df`.
- `p$sdev` gives the *eigenvalues of the covariance matrix* of the data. One can interpret the `n`th value in `p$sdev` as corresponding to the relative proportion of the variance in the data explained by the `n`th principal component. Put another way, `p$sdev[n] / sum(p$sdev)` is the proportion of the variance in the data explained by the `n`th principal component.

PCA on the `msq` dataset

Prepare the data in this fashion:

- Extract the columns `active:scornful` from the `msq` dataset.
- Look at the number of `NA`s in each column (hint: use `colSums()` in conjunction with `is.na()`). For simplicity's sake, throw out the columns with a huge number of missing values and subsequently remove all the rows with any `NA`s.

Afterward, run a PCA on the remaining variables.

- Write a function `top(n)` that prints out the top 10 *loadings* of the `n`th principal component, ordered by absolute value.
- Look at the PCA loadings for the first 5-10 principal components. Use `corrplot()` on the loadings (with the `is.corr=FALSE` option) to visually explore these relationships. After doing so, interpret and assign concise names to the principal components which seem to represent something coherent.
- Plot the eigenvalues obtained via `prcomp(...)$sdev`. How do their relative magnitudes relate to the interpretability of each principal component?
- Suppose that we use the first nth principal components to predict Extraversion and Neuroticism using a simple, unregularized linear model. Calculate a cross-validated RMSE for $n = 1, 2, \ldots$, plot them against n, and compare to the cross-validated RMSE which you got in the self-assessment when using regularized linear regression with all of the original variables. Interpret the results.
- Spend a couple minutes reading about the history of trait theories. Can you assign any hierarchical interpretation to the principal components obtained for this dataset? How do they relate to Extraversion and Neuroticism?

PCA on the speed dating dataset