Practical Data Science: Reducing High Dimensional Data in R

I can't remember the last time I worked on a data set with less than **500** features. This isn't a big deal with today's computing power, but it can become unwieldy when you need to use certain forest-based models, heavy cross-validation, grid tuning, or any ensemble work. *Note: the term variables, features, predictors are used throughout and mean the same thing.*

Here are some ways of dealing with **high-dimensionality data** (i.e. having too many variables) are:

1. Get more computing muscle (like RStudio on an Amazon Web Services EC2 instance),
2. Prune your data set using feature selection (measure variables effectiveness and keeps only the best - built-in feature selection - see fscaret),
3. Work on a sample subset or break up the data into chunks (How To Work With Files Too Large For A Computer's RAM? Using R To Process Large Data In Chunks),
4. And finally, the subject of this course, use **feature reduction** (also refereed as feature extraction) to create new variables made of bits and pieces of the original variables.

According to wikipedia:

> "Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components."

Principal component analysis (PCA)

You'll find reams of explanations on the web, but, in a nutshell, **PCA** looks for the set of related variables in your data that explain most of the variance and creates a new feature out of it. This becomes your first component. It will then keep doing so on the next set of variables unrelated to the first, and that becomes your next component, and so on and so forth. This is done in an unsupervised manner so it doesn't care what your response variable/outcome is (but you should exclude it from your data before feeding it into **PCA**). *As a side note, this is the basis of a lot of compression software – it is that good.*

Package {stats}

There are plenty of PCA libraries in R, in this course, we'll focus on two common ones available in the base {stats} package (i.e. you don't have to download any package, its already there):

1. princomp performs a principal components analysis on the given numeric data matrix and returns the results as an object of class “princomp”.
2. prcomp Performs a principal components analysis on the given data matrix and returns the results as an object of class “prcomp”.

They both perform PCA but get there in slightly different ways - princomp uses the eigen function while prcomp uses singular value decomposition of the data. I won't pretend to understand the nuances between both approaches, and as I want to stay away from formulas and keep this very practical, I will only say that it is important to know the existence of both functions. If one doesn't work on a particular data set, try the other…