A few months ago, I wrote about the [**Predictive Power Score (PPS)**](https://paulvanderlaken.com/2020/05/04/predictive-power-score-finding-patterns-dataset/): a handy metric to quickly explore and quantify the relationships in a dataset.

As a social scientist, I was taught to use a [correlation matrix](https://paulvanderlaken.com/2020/07/28/publication-ready-correlation-matrix-significance-r/) to describe the relationships in a dataset. Yet, in my opinion, the PPS provides three handy advantages:

1. **PPS works for any type of data, also nominal/categorical variables**
2. **PPS quantifies non-linear relationships between variables**
3. **PPS acknowledges the asymmetry of those relationships**

[Florian Wetschoreck](https://twitter.com/Wetschoreck) came up with the PPS idea, wrote the [original blog](https://towardsdatascience.com/rip-correlation-introducing-the-predictive-power-score-3d90808b9598), and programmed a [Python implementation](https://github.com/8080labs/ppscore) of it (called ppscore).

Yet, I work mostly in R and I was very keen on incorporating this powertool into my general data science workflow.

So, over the holiday period, I did something I have never done before: **I wrote an R package!**

**Installation**

# You can get the development version from GitHub:

# install.packages('devtools')

**Usage**

The ppsr package has three main functions that compute PPS:

* score() – which computes an x-y PPS
* score\_predictors() – which computes X-y PPS
* score\_matrix() – which computes X-Y PPS

**Visualizing PPS**

Subsequently, there are two main functions that wrap around these computational functions to help you visualize your PPS using ggplot2:

* visualize\_predictors() – producing a barplot of all X-y PPS
* visualize\_matrix() – producing a heatmap of all X-Y PPS

PPS matrix for iris

Note that Species is a nominal/categorical variable, with three character/text options.

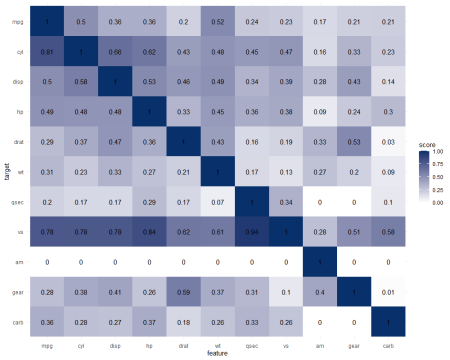
A correlation matrix would not be able to show us that the type of iris Species can be predicted extremely well by the petal length and width, and somewhat by the sepal length and width. Yet, particularly sepal width is not easily predicted by the type of species.

Correlation matrix for iris

**Exploring mtcars**

It takes about 10 seconds to run 121 decision trees with visualize\_matrix(mtcars). Yet, the output is much more informative than the correlation matrix:

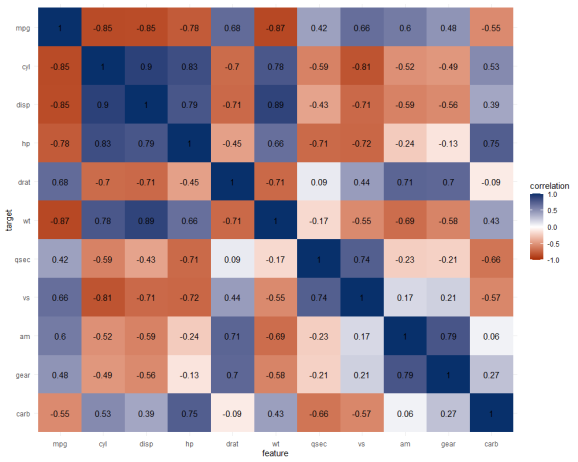
* cyl can be much better predicted by mpg than the other way around
* the classification of vs can be done well using nearly all variables as predictors, except for am
* yet, it’s hard to predict anything based on the vs classification
* a cars’ am can’t be predicted at all using these variables

PPS matrix for mtcars

The correlation matrix does provides insights that are not provided by the PPS matrix. Most importantly, the sign and strength of any linear relationship that may exist. For instance, we can deduce that mpg relates strongly negatively with cyl.

Yet, even though half of the matrix does not provide any additional information (due to the symmetry), I still find it hard to derive the most important relations and insights at a first glance.

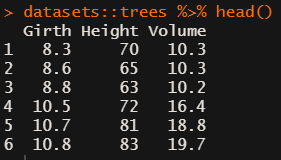
Moreover, the rows and columns for vs and am are not very informative in this correlation matrix as it contains pearson correlations coefficients by default, whereas vs and am are binary variables. The same can be said for cyl, gear and carb, which contain ordinal categories / integer data, so you can discuss the value of these coefficients depicted here.

Correlation matrix for mtcars

**Exploring trees**

In R, there are many datasets built in via the datasets package. Let’s explore some using the ppsr::visualize\_matrix() function.

datasets::trees has data on 31 trees’ girth, height and volume.

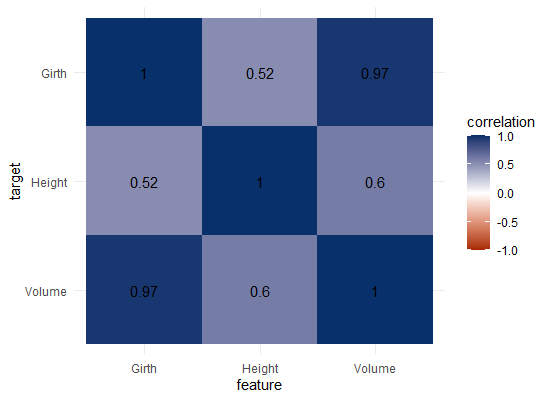


visualize\_matrix(datasets::trees) shows that both girth and volume can be used to predict the other quite well, but not perfectly.

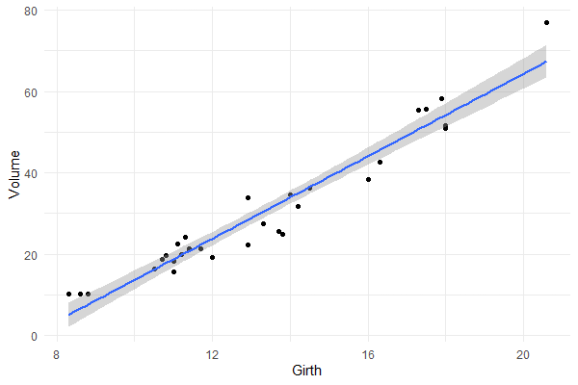


Let’s have a look at the correlation matrix.

The scores here seem quite higher in general. A near perfect correlation between volume and girth.



Is it near perfect though? Let’s have a look at the underlying data and fit a linear model to it.



You will still be pretty far off the real values when you use a linear model based on Girth to predict Volume. This is what the original PPS of 0.65 tried to convey.

Actually, I’ve run the math for this linaer model and the RMSE is still 4.11. Using just the mean Volume as a prediction of Volume will result in 16.17 RMSE. If we map these RMSE values on a linear scale from 0 to 1, we would get the PPS of our linear model, which is about 0.75.

So, actually, the linear model is a better predictor than the decision tree that is used as a default in the ppsr package. That was used to generate the PPS matrix above.

Yet, the linear model definitely does not provide a perfect prediction, even though the correlation may be near perfect.

**Conclusion**

In sum, I feel using the general idea behind PPS can be very useful for data exploration.