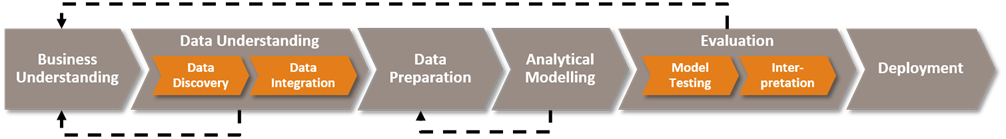
R is one of the most popular software packages for any kind of statistical analysis and data science. However, with friends of the (in parts) competing Python ecosystem R users share the challenge of finding suitable packages for the task at hand from the overwhelming package repository.

This article provides a (non-comprehensive) list of my favourite R packages throughout the Data Science Process.

The Data Science Process

The workflow from data acquisition to deployment of an analytical model begins at a business-level understanding of a problem. The five steps that follow the business understanding stage are often linked with each other or to business understanding.

These links highlight the need for constant checks of the business scope and improvement of the data foundation. Because of this, many of the R packages listed below are useful in more than one step of the way.



In this post, I will not take the business understanding step into account, since it is a task of human-to-human interaction rather than R packages (possibly IBM Watson and alike have an edge there). I also neglect the deployment step, since the roll-out of a data science solution will most likely not be done in R.

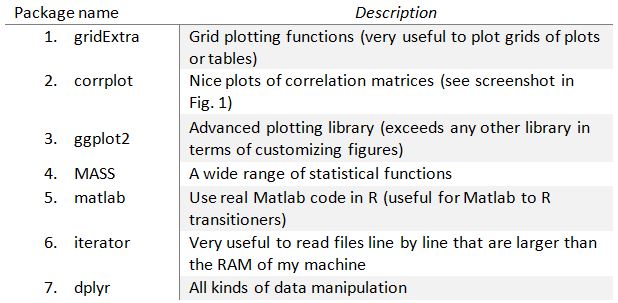
Data Understanding

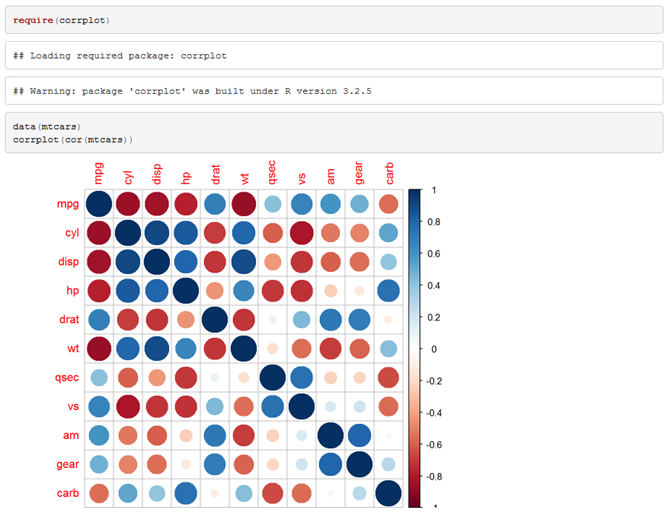
In particular the first two steps following business understanding, data understanding and data preparation, are key to successful data science. A tidy dataset and cleverly chosen features for further analysis are prime requirements for the following modelling step.

Here, by data understanding I mean exploratory data analysis, e.g.

* Histograms and distribution analysis
* Density plots
* Correlation analysis
* Principal component analysis.

The packages I name below are particularly useful to automate and optimize this process step.



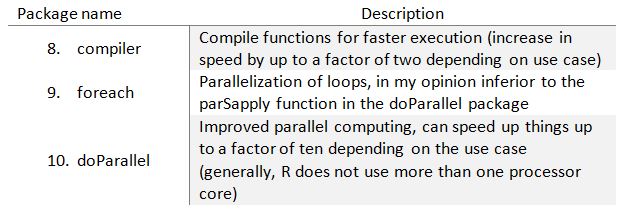


*Fig. 1: Correlation plot of the variables in the mtcars data set generated using the corrplot library.*

Data Preparation

Generally, R is very well suited to handle data preparation tasks. In addition to some of the packages listed in the previous section (in particular dplyr) there are three further packages that fall in between the section of data understanding and preparation.

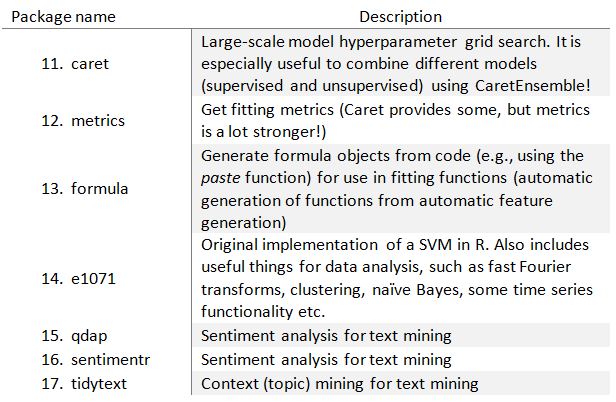
There are a number of parallel processing packages. I personally prefer doParallel, since it does not require the user to manually install MPI or OpenMP (both of which is relatively simple on a Linux system).



Analytical Modelling

Modelling in R is particularly easy to implement, since all libraries share a reasonably common syntax for predicting from models (even though in some cases the *predict* function requires varying data structures). The model training process on the other hand is more complex, since every model package has its own method.

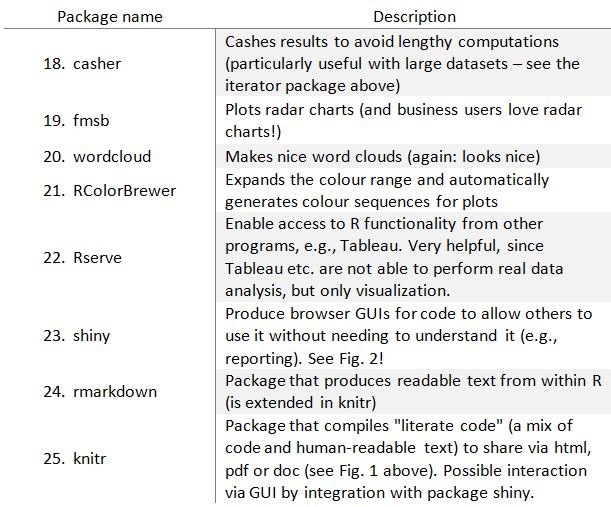
The caret package overcomes this issue. It provides a frontend to many modelling packages and supplies the user with a common *train* function. Secondly, caret adds an outstanding method for model hyperparameter grid searches and data pre-processing.

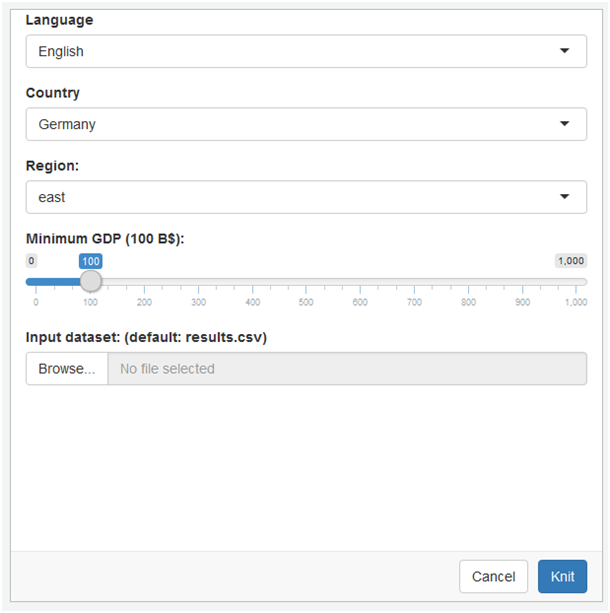


Evaluation

One of the key strengths of R is its capability to present data and the workflow leading to a result in a concise manner. Of these packages shiny, rmarkdown and knitr stand out. They make it incredibly easy to reproduce results and understand work in a shared environment.

Shiny is a browser-based GUI for an R script and is a development framework for sophisticated apps around R code. Rmarkdown extends the ecosystem in the other direction: Combined with knitr it prints code, plots and regular (formatted!) text in a single document that can be compiled to html, pdf or Word documents.





*Fig. 2: Exemplary Shiny user interface for parametrized reporting.*

Since this list of packages is not intended to be comprehensive, I am curious about your opinions and favourite packages in the comment section!