"10 R packages that I wish I knew about earlier" - revisited

10 R packages I wish I knew about earlier

by yhat

February 10, 2013

I started using R about 3 years ago. It was slow going at first. R had tricky and less intuitive syntax than languages I was used to, and it took a while to get accustomed to the nuances. It wasn't immediately clear to me that the power of the language was bound up with the community and the diverse packages available.

R can be more prickly and obscure than other languages like Python or Java. The good news is that there are tons of packages which provide simple and familiar interfaces on top of Base R. This post is about ten packages I love and use everyday and ones I wish I knew about earlier.

10 R packages every data scientist should know about

The yhat blog lists 10 R packages they wish they'd known about earlier. Drew Conway calls them "10 reasons to always start your analysis in R". They're all very useful R packages that every data scientist should be aware of. They are:

1. **sqldf** (for selecting from data frames using SQL)
2. **forecast** (for easy forecasting of time series)
3. **plyr** (data aggregation)
4. **stringr** (string manipulation)
5. Database connection packages **RPostgreSQL**, **RMYSQL**, **RMongo**, **RODBC**, **RSQLite**
6. **lubridate** (time and date manipulation)
7. **ggplot2** (data visulization)
8. **qcc** (statistical quality control and QC charts)
9. **reshape2** (data restructuring)
10. **randomForest** (random forest predictive models)

You can find links to all of these packages and tips on how to use them at link below.

yhat blog: 10 R packages I wish I knew about earlier

10 R packages to win Kaggle competitions

Xavier Conort, DataRobot

*Contact author: xavier@datarobot.com

Keywords: Kaggle, Machine Learning

R is rapidly growing in popularity among statisticians, data scientists and actuaries. An actuary by training, I became an *R* enthusiast myself 3 years ago when I discovered that *R* offered me a powerful platform for statistical and actuarial analysis. The main draw for me was the large palette of Machine Learning algorithms to tackle predictive modeling problems outside my comfort zone -- ranging from churn prediction to essay scoring, sales forecasts, flight arrival prediction, recommendation engine and credit scoring. Machine Learning helped me significantly reduce the modeling effort compared to traditional statistical parametric techniques, work on dirty data with limited domain insight and extract value from unstructured or large dimensionality datasets. Thanks to *R*, I placed in the top 10 of more than 15 Kaggle competitions and won several of them. 10 *R* packages were the key ingredients of my Kaggle solutions. In this talk, I will cover why I found those 10 packages particularly powerful and how I used them to build winning solutions.

10 R Packages:

Allow the Machine to Capture Complexity

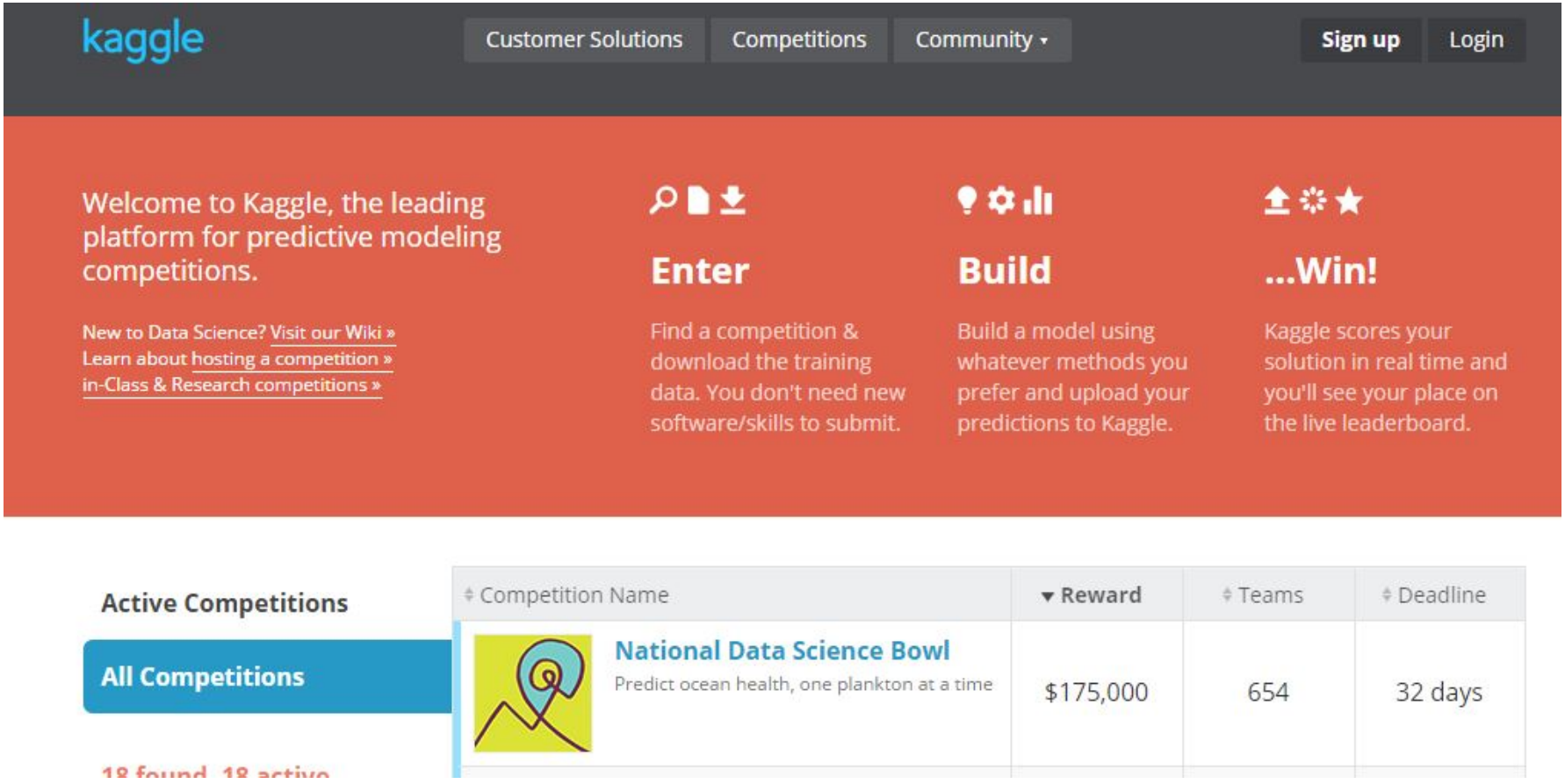
1. gbm
2. randomForest
3. e1071

Take Advantage of High-Cardinality Categorical or Text Data

4. glmnet
5. tau

Make Your Code More Efficient

6. Matrix
7. SOAR
8. forEach
9. doMC
10. data.table


kaggle
Customer Solutions
Competitions
Community
Sign up
Login
Welcome to Kaggle, the leading platform for predictive modeling competitions.
New to Data Science? Visit our Wiki »
Learn about hosting a competition »
in-Class & Research competitions »
Enter
Find a competition & download the training data. You don't need new software/skills to submit.
Build
Build a model using whatever methods you prefer and upload your predictions to Kaggle.
...Win!
Kaggle scores your solution in real time and you'll see your place on the live leaderboard.
Active Competitions
All Competitions
Competition Name
Reward
Teams
Deadline
National Data Science Bowl
Predict ocean health, one plankton at a time
$175,000
654
32 days

Package 'dplyr'

January 27, 2015

Type Package

Version 0.4.1

Title A Grammar of Data Manipulation

Description A fast, consistent tool for working with data frame like objects, both in memory and out of memory.

URL https://github.com/hadley/dplyr

BugReports https://github.com/hadley/dplyr/issues

Depends R (>= 3.0.2)

Imports assertthat, utils, R6, Rcpp, magrittr, lazyeval (>= 0.1.10), DBI (>= 0.3)

Suggests RSQLite (>= 1.0.0), RMySQL, RPostgreSQL, data.table, testthat, knitr, microbenchmark, ggplot2, mgcv, Lahman (>= 3.0-1), nycflights13, methods

VignetteBuilder knitr

Introduction to dplyr

2015-01-13

When working with data you must:

- Figure out what you want to do.
- Precisely describe what you want in the form of a computer program.
- Execute the code.

The dplyr package makes each of these steps as fast and easy as possible by:

- Elucidating the most common data manipulation operations, so that your options are helpfully constrained when thinking about how to tackle a problem.
- Providing simple functions that correspond to the most common data manipulation verbs, so that you can easily translate your thoughts into code.
- Using efficient data storage backends, so that you spend as little time waiting for the computer as possible.

The goal of this document is to introduce you to the basic tools that dplyr provides, and show how you to app them to data frames. Other vignettes provide more details on specific topics:

- databases: as well as in memory data frames, dplyr also connects to databases. It allows you to work with remote, out-of-memory data, using exactly the same tools, because dplyr will translate your R coc into the appropriate SQL.
- benchmark-baseball: see how dplyr compares to other tools for data manipulation on a realistic use case.
- window-functions: a window function is a variation on an aggregation function. Where an aggregate function uses `n` inputs to produce 1 output, a window function uses `n` inputs to produce `n` outputs.

magrittr

magrittr: A Forward-Pipe Operator for R

- **Authors** : Stefan M Bache and Hadley Wickham
- Provides a mechanism for chaining commands with a new forward-pipe operator, %>%.
- This operator will forward a value, or the result of an expression, into the next function call/expression.
- There is flexible support for the type of right-hand side expressions.
- To quote Rene Magritte, "Ceci n'est pas un pipe."

```
value %>%
  foo %>% {
    x <- bar(.)
    y <- baz(.)
    x * y
  } %>%
  and_whatever
```

data.table: Extension of data.frame

Fast aggregation of large data (e.g. 100GB in RAM), fast ordered joins, fast add/modify/delete columns by group using no copies at all, list columns and a fast file reader (fread). Offers a nat and flexible syntax, for faster development.

Version:	1.9.4
Depends:	R (≥ 2.14.0)
Imports:	methods, chron, reshape2
Suggests:	ggplot2 (≥ 0.9.0), plyr, reshape, testthat (≥ 0.4), hexbin, fastmatch, nlme, x bit64
Published:	2014-10-02
Author:	M Dowle, T Short, S Lianoglou, A Srinivasan with contributions from R Saporta, E Antonyan
Maintainer:	Matt Dowle <mdowle at mdowle.plus.com>

Rdatatable / **data.table**

Home

Matt Dowle edited this page 5 days ago · 38 revisions

The R `data.table` package provides an enhanced version of `data.frame` including:

- fast **aggregation** of large data; e.g. 100GB in RAM (see benchmarks on two billion rows)
- fast **ordered joins**; e.g. rolling forwards, backwards, nearest and limited staleness
- fast **overlapping range joins**; e.g. GenomicRanges
- fast add/modify/delete of columns **by reference** by group using no copies at all
- cells may themselves contain vectors/objects/functions; i.e. **columns of type list**
- **fast and friendly file reader**: fread

RStudio Blog

Introducing tidyr

July 22, 2014 in **Packages**, **Uncategorized**

tidyr is new package that makes it easy to "tidy" your data. Tidy data is data that's easy to work with: it's easy to munge (with dplyr), visualise (with ggplot2 or ggvis) and mode (with R's hundreds of modelling packages). The two most important properties of tidy data are:

- Each column is a variable.
- Each row is an observation.

MASS is an amazing package, I always overlook it as well. Maarten-Jan (Kallen, co-developerof Renjin) made a very interesting point about it too. One of the issues their particular interpreter has is that they haven't gotten MASS working for some reason (technical in nature I think) and there are a phenomenal number of packages that have it as a dependency.

Michael Cooney - Applied AI Data Scientist