

# The Yhat Blo

machine learning, data science, engineering

Get Updates

## Moving from R to Python: The Libraries You Need to Know

by Yhat

August 29, 2016

[Learn More](#)

[Share on Facebook](#)

[Share on Twitter](#)

[Share on LinkedIn](#)

[Share on Reddit](#)

### Why the switch?

One of my favorite parts of machine learning in Python is that it got the benefit of observing the R community and then emulating the best parts of it. I'm a big believer that a language is only as helpful as its libraries. So in this post I'm going to go over some critical packages that I use almost every time I work in R, and their counterpart(s) in Python.

glm, knn, randomForest, e1071 (yes, this is actually a meaningful package's name) -> scikit-learn

One thing that is a blessing and a curse in R is that the machine learning algorithms are *generally* segmented by package. Meaning instead of having a single (or set) of ML libraries that each implement some common algorithms, each algorithm gets its own package. It's sort of nice because you can find very esoteric,

cutting edge implementations of algorithms, but it can be a pain for day-to-day use where you might be switching between algorithms. This pain is something that Python's `scikit-learn` solves really well. `scikit-learn` provides a common set of ML algorithms all under the same API. It makes switching between `LogisticRegression` and `GradientBoostingMachines` a one-liner.

## reshape/reshape2, plyr/dplyr -> pandas

This was actually the subject of one of [our first posts](#). `pandas` took the best parts of data munging in R and turned it into a Python package. This includes its own implementation of a data frame along with ways to modify and restructure it. Basically it took the best parts of `reshape / reshape2` and `plyr / dplyr` and Pythonified it!

## ggplot2 -> ggplot + seaborn + bokeh

One thing that R still does better than Python is plotting. Hands down, R is better in just about every *facet*. Even so, Python plotting has matured though it's a fractured community. If you like the ggplot-style syntax, then look no further than [Yhat's own ggplot](#). If you're after super statistical and technical plots then reach for `seaborn`. And if you're in the market for some super slick, great looking interactive plots then try out `bokeh`.

## stringr -> nothing

String manipulation in "base R" is nearly as unintuitive as it is silly. Any time I'm working with strings in R I do 2 things (in order):

- briefly nod in appreciation to New Zealand for producing Hadley Wickham
- import `stringr`



Much obliged, New Zealand

`stringr` is an absolute lifesaver. It's well written, performant (at least I think so), and easy to install (don't overlook this last item. if people can't install your software, there's no sense in making it).

Ok so `stringr` appreciation monologue complete. So the good news for you is that Python is so great for string manipulation, you don't really need a string library! It has a fantastic built-in regular expressions library, `re`, and a built-in string meta-library appropriately called `string`. So lucky for you, Python comes with all string-related batteries included!

## RStudio -> Rodeo

To many users, `RStudio` is synonymous with R. And why not? It's a great IDE for data analysis in R. Historically speaking, there haven't been a lot of comparable options for Python. Of course this is no longer the case. We released the very first version of Rodeo just over a year ago and released the 2.0 for Windows, OSX, and Linux about a month ago.

"Ever since we've used RStudio, we've been looking for an IDE like it for Python. We went through IDEs such as Sublime Text and Spyder, none of which suited our likings. We searched and found Rodeo and couldn't have been more pleased with the IDE." -Stephen Hsu, University of California, Berkeley

[Download Rodeo!](#)

## Knitr -> Jupyter

`Knitr` is a great way to create reproducible and highly visual analysis using R. It's been a staple in RStudio for a while now. In the Python world, the most analogous package is `Jupyter`. Jupyter notebooks provide an interactive environment for programming in Python (and other languages) that focuses on reproducibility and visualization--it even has a plugin for R!

## sqldf -> pandasql

`sqldf` is a great way for SQL users to comfortably manipulate data in R. I myself used it when I first started learning R. Way back when, Yhat actually built a similar package for Python called `pandasql`. Same concept: write SQL queries against your data frames, get data frames back! Fast-forward 3 years and `pandasql` has over 256 stars on GitHub :). Not bad for a library with only 358 lines of code!

[LEARN MORE](#)

[f SHARE ON FACEBOOK](#)

[t SHARE ON TWITTER](#)

[in SHARE ON LINKEDIN](#)

[r SHARE ON REDDIT](#)

---

### Our Products



A Python IDE built for doing data science directly on your desktop.

[DOWNLOAD IT NOW!](#)



A platform for productionizing, scaling, and monitoring predictive models in production applications.

LEARN MORE



Yhat (pronounced Y-hat) provides data science and decision management solutions that let data scientists create, deploy and integrate insights into any business application without IT or custom coding.

With Yhat, data scientists can use their preferred scientific tools (e.g. R and Python) to develop analytical projects in the cloud collaboratively and then deploy them as highly scalable real-time decision making APIs for use in customer- or employee-facing apps.

atro

info@yhathq.com  
+1 718 855 2107  
+49 15735983455

P r0

ScienceOps  
Rodeo

rad



Company  
Blog  
Jobs  
RSS

el sN

Email Address

Get Updates

CEPTOC

Made in New York City • © 2016 yhat