O'REILLY®

Python vs R for Data Science

Michael Grogan

COMPARISONS

Python vs. R for Data Science

Michael Grogan

O'REILLY®

Beijing • Boston • Farnham • Sebastopol • Tokyo

ython vs. R for Data Science

by Michael Grogan

Printed in the United States of America.

Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O’Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*ttp://oreilly.com/safari*). For more information, contact our corporate/institutional sales department: 800-998-9938 or *corporate@oreilly.com*.

Editor: Nicole Tache

Production Editor: Melanie Yarbrough

Copyeditor: Octal Publishing Services

Interior Designer: David Futato

Cover Designer: Karen Montgomery

Illustrator: Rebecca Demarest

June 2018: First Edition

Revision History for the First Edition

- 2018-06-18: First Release

While the publisher and the author have used good faith efforts to ensure that the information

P*h*

nd instructions contained in this work are accurate, the publisher and the author disclaim all responsibility for errors or omissions, including without limitation responsibility for damages resulting from the use of or reliance on this work. Use of the information and instructions contained in this work is at your own risk. If any code samples or other technology this work contains or describes is subject to open source licenses or the intellectual property rights of others, it is your responsibility to ensure that your use thereof complies with such licenses and/or rights.

978-1-492-03393-6

a

ython vs. R for Data Science

Introduction

Python and R are two of the mainstream languages in data science. Fundamentally, Python is a language for **programmers**, whereas R is a language for **statisticians**. In a data science context, there is a significant degree of overlap when it comes to the capabilities of each language in the fields of **regression analysis** and **machine learning**. Your choice of language will depend highly on the environment in which you are operating. In a **production** environment, Python integrates with other languages much more seamlessly and is therefore the modus operandi in this context. However, R is much more common in **research** environments due to its more extensive selection of libraries for statistical analysis.

Basics

P

Python	R
CURRENT VERSION	
3.6	3.4.3
SELF-DEFINED AS	
Python is a **programming language** that lets you work quickly and integrate your systems effectively. According to the official website, the Python quote emphasizes productivity as well as its use as a glue language.	R is an open source language that is specifically designed for conducting **statistical analysis**. As such, it is highly popular within fields such as data science, engineering, and other cognitive disciplines. The R Project for Statistical Computing describes the R language as an environment specifically designed for "statistical computing and graphics."
STRENGTHS	
Python has significantly more flexibility in interacting with other programming languages and is quite fast compared to R. Moreover, the **pandas** and **scikit-learn** libraries are advantageous when it comes to data manipulation and machine learning, respectively. Python is a general-purpose programming language and therefore is more universal when it comes to combining with other languages such as Java and PHP.	R is far more efficient at conducting statistical analysis; provided packages automate much of this process. Strictly speaking, R is a statistical environment rather than a programming language, thus it is more adept at doing intensive statistical analysis. However, many data scientists prefer Python's scikit-learn library when it comes to implementing machine learning algorithms.
HURDLES	
Coding in Python is more cumbersome when it comes to statistical analysis; packages in R make this process much more intuitive.	R is significantly slower than Python and is less adept at integrating with other programming languages. Whereas R has the upper hand when it comes to statistical analysis, it is significantly slower than Python, and the latter language is often more flexible when it comes to algorithmic development or for final program release.

<table>
<tr><th colspan="2">LICENSE</th></tr>
<tr><td>PSFL (BSD-like)</td><td>GNU General Public License (Both Python and R have “permissive” licenses, allowing redistribution of modified language implementations and tools without source code)</td></tr>
<tr><th colspan="2">POPULARITY AND USAGE</th></tr>
<tr><td>#4 on TIOBE index
#1 on IEEE Spectrum ranking
#2 on GitHub (by opened pull request)
#1 Packt’s 2017 Developer Skills and Salary Report Ranking
#2 Python is the <b>second</b> most popular language among data scientists according to O’Reilly’s 2017 US Data Science Survey, with slightly more than 60% of respondents using this language.</td><td>#8 on TIOBE index
#9 on IEEE Spectrum ranking
#9 on GitHub (by opened pull request)
#8 Packt’s 2017 Developer Skills and Salary Report Ranking
#3 R is the <b>third</b> most popular language among data scientists according to O’Reilly’s 2017 US Data Science Survey, with slightly more than 50% of respondents using this language.</td></tr>
<tr><th colspan="2">BACKWARD COMPATIBILITY</th></tr>
<tr><td>Promised within the 3.x releases. Python 2.7 lifetime was extended to allow migration. Python had a good track record with backward compatilbility until the 3.0 release in 2008, which required nontrivial changes in almost all 2.x programs. Nine years later, all important projects either provide 2.x and 3.x compatibility or support 3.x only. No major breakage is expected when Python 4 is released.</td><td>Package dependent. R expert Hadley Wickham describes backward compability as somewhat of an “academic” issue among the R community, given that packages are updated to the latest version using the update.packages() command, which updates to the latest package version even if there has been a change in the major version.</td></tr>
<tr><th colspan="2">COMMUNITY</th></tr>
<tr><td>python.org. Additionally, the Python Weekly email newsletter provides significant information on the use of Python as well as related news such as jobs, new releases, and others.</td><td>R-bloggers. The community comprises more than 750 contributors that regularly contribute useful information on the R language</td></tr>
<tr><th colspan="2">MARQUEE USERS</th></tr>
</table>

Engineers, **programmers**, web developers. In general terms, Python is better for programming-oriented tasks.	Engineers, researchers, **statisticians**. R has the upper hand when it comes to statistics-oriented tasks such as regression analysis or time series modeling.
MAIN USE CASES	
Data wrangling, machine learning, preprocessing, text mining, web scraping. Generally speaking, Python has the upper hand over R when it comes to creating a general-purpose program that interacts more seamlessly with other programming languages. The Pandas library for data manipulation and scikit-learn for machine learning have long been favored by Python users.	Statistics, regression analysis, time series analysis, visualisation. R has been known to have a slower run time than Python, and less flexibility in interacting with other languages. With that being said, R is significantly more intuitive when it comes to implementing statistics or machine learning functions and therefore is ideal for designing a program/testing before implementing in another language.
EASE OF LEARNING	
Better suited to users more experienced with low-level programming languages such as **C++** and **Java**.	Better suited to users who have been exposed to other statistical environments or languages such as **MATLAB**, **SAS**, and **SPSS**. R is technically a statistical environment rather than a programming language, per se. Users who are already adept at using other statistical environments will likely find the transition to R to be more seamless than that of Python.
SALARY EXPECTATIONS	
Python-only salary: **$55,000–135,000** according to the 2017 O’Reilly US Data Science Survey. In particular, those who had expertise in Python’s scikit-learn were reported to have a median salary of **$100,000**.	R-only salary: **$50,000–125,000** according to the 2017 O’Reilly US Data Science Survey. In general, those users who had a command of both Python and R are seen as more marketable from a career standpoint.
INTEGRATION	

Python can integrate with Tableau through **TabPy**, and with Apache Spark via PySpark. This language also allows for integration with machine intelligence libraries such as TensorFlow.

The **microstrategyR** package is of particular interest to business analysts, allowing for connection of R to business intelligence platforms, whereas **sparklyr** allows for analysis of large datasets through connection to Apache Spark.

Core Characteristics

Python	Go
EXECUTION MODEL	
Interpreted. Python is actually compiled to bytecode, which is then interpreted by the Python runtime. The **Pypy** implementation includes a just-in-time (JIT) compiler to generate machine code on the fly. Go is compiled directly to machine code.	Interpreted. R is also an interpreted language, in the sense that it provides an interface to compile the code; that is, expressions in R are also JIT-compiled to bytecode, which can then be interpreted.
INTERACTIVE CONSOLE	
Bundled. Also, multiple enhanced consoles are available, including iPython, Jupyter Notebook, and Python Anywhere (online). An interactive console or REPL is an excellent learning, debugging, and exploration tool. iPython and its offspring, Jupyter Notebook, were keys to the success of Python in data science.	RStudio, R Commander, Jupyter Notebook. RStudio is the cornerstone user interface for the R language. However, the language is also compatible with the R Commander and Jupyter Notebook GUIs.
STANDARD LIBRARY	
Comprehensive	Comprehensive
INHERITANCE	
Multiple inheritance	Multiple inheritance
TYPING	
Strong, dynamic. Both languages are strong typed (i.e., they avoid implicit type conversion, a source of many bugs in JavaScript). Python introduced type hints in v3.5 (2015) to support static checking and enhanced IDE support for autocompletion; the type annotations have no effect on runtime behavior, so the language remains dynamically typed.	Strong, dynamic. R is also dynamically typed in the sense that it is not necessary to predefine variables before execution (as would be the case in a low-level language such as C++).
INTERFACE ENFORCEMENT	

At runtime, via **duck typing** (allows for the running of operations on objects without specifically having to predefine those objects beforehand).	At runtime, via **duck typing** (allows for the running of operations on objects without specifically having to predefine those objects beforehand).
FIRST CLASS FUNCTIONS	
Yes. This means that a function can be assigned to a variable, passed to another function as an argument, and returned as a result from another function.	Yes. A function can be assigned to a variable in the same way as Python.
CLOSURES	
Yes	Yes. The scope of a function encompasses variables that appear in its body but that are not local variables or arguments. This is necessary for correct support of first class functions in a language with lexical scoping.
ANONYMOUS FUNCTIONS	
Limited syntax. Python’s lambda keyword allows anonymous function declarations that hold only expressions but not statements such as loops.	Limited syntax. R uses the function keyword in place of a lambda to do so.
DATA ABSTRACTION	
Classes, named tuples	Classes. Both Python and R support object-oriented programming, where classes in R are of an S3 or S4 type. Although you can add methods to a class in R using the setMethod() function, this process is more simplistic in Python.

Native Types

Python	Go
NATIVE COLLECTIONS	
list, tuple, dict, set	**array, data frame, factor, list, matrices, vectors.** We consider a collection “native” if it is built in (no import needed) and is supported by special literal syntax to create instances. Even though R and Python share some similarities in terms of the different data types they use, R is distinctly known for operating using data frames, or distinct data tables in R that allow for advanced statistical manipulation.
NATIVE STRING TYPES	
str, bytes. The Python 3 str type represents human text (elements are Unicode characters), whereas bytes are sequences of integers with values from 0 to 255.	**character (string).** In R, there is no distinct difference between the definition of a string or character—both represent a character variable that contains non-numeric data. In R, the str()function is used to identify the data type in question; it does not represent any particular data type in its own right.
INTEGERS	
int()	**as.integer()**
FLOATS	
float(). You can convert a numeric value into float format in Python by using the float() command.	**as.double(), as.numeric().** In R, float data types are represented by double or numeric format and you convert values into float format by using either as.double() or as.numeric().

Data Science Libraries

Python	Go
DATA WRANGLING	
Pandas. Python is particularly strong in this area, with the Pandas library being very extensive in this regard.	**data.table, dplyr, plyr**
DATABASE CONNECTIONS	
mysql-connector-python, psycopg2, SQLAlchemy. Both Python and R have several libraries available to connect to a SQL database, import data, and commit queries, among other common tasks.	**rmysql, rpostgresql**
MACHINE LEARNING	
PyBrain, PyLearn2, scikit-learn, statsmodels. scikit-learn in Python is quite popular for running machine learning algorithms, and the faster processing speed of Python makes it more suitable for this purpose.	**caret, randomForest, rpart, neuralnet**
REGRESSION ANALYSIS	
Numpy, scikit-learn, SciPy, statsmodels	**lmtest, car.** Both languages are capable of conducting advanced statistical analysis, including regression analysis. However, the associated packages in R are more extensive and offer more flexiblility in this area.
TIME SERIES	
Prophet, PyFlux, statsmodels	**MASS, tseries, forecast.** As in the case of regression analysis, both languages have the capability to conduct analysis on time series data. However, the packages in R are more extensive in conducting such analysis.
VISUALIZATION	

matplotlib is the dominant plotting library in Python. Others include **Plotly**, **Pygal**, **Bokeh**, **and Seaborn**.	**ggplot2** is the dominant plotting library in R. You can aslo use **Plotly** in R as well as **caret**, **igraph**, and **highcharter**.

How to Choose

What is your background?

Your choice of language will be highly dependent on your background. If you are a **programmer** who has used other **low-level languages** such as **C++**, using Python will prove a much more seamless transition. However, if you come from an **academic** background or have previously used **statistical programs** such as **SAS**, **SPSS**, and others, R will likely be easier to come to grips with compared to Python.

What types of tasks do you wish to accomplish?

Are you looking to conduct a high degree of **statistical modeling**, or is **data manipulation** and **machine learning** your goal? If it's the former, the packages in R are specifically geared toward **statistics** and **regression analysis**, which would make R a better choice in this regard. However, **Pandas** and **scikit-learn** are highly renowned for their use in data manipulation and machine learning, respectively, and Python is therefore a better choice in these areas. Moreover, even though the **TensorFlow** machine learning framework—which was developed by researchers working on the Google Brain Team—is available in both Python and R, the environment works much more seamlessly with Python's Anaconda environment.

What environment are you operating in?

As mentioned in the introductory section, Python is much more flexible for integrating with other programming languages. In this regard, if you are working with developers or in an environment where there is an emphasis on **production**, Python is the better choice. However, if you are conducting statistical analysis for **research purposes**, R is a more efficient choice as implementation of statistical algorithms are, in many cases, easier than Python thanks to R's many libraries designed for this purpose.