Learning Bytes

by Byte Academy

PYTHON V. R WEBINAR

For Interactive Brokers

Part I Premiere of Learning Bytes Series

7.11.2017

Intro to Byte Academy

Producer of Learning Bytes Series

Industry focused coding school headquartered in NYC with full, part-time, remote + corporate training programs

Known for its FinTech program, the first of its type

1st Python Fullstack Full-Time Program in NYC

Programs in Data Science, Quant Algos, Blockchain, more

Career Guarantee + Tuition Deferral

Emphasis on events + building the FinTech, Quant, Developer Communities

www.byteacademy.co

Meet Our Expert Leading the webinar

Lesley Cordero
Data Science instructor at Byte Academy

Will use iPython notebook

Course + Workshop topics include:

- Data Science Bootcamp
- Intro to Python
- Intro to Data Science + Statistics Using R
- Data Wrangling with Python
- Data Visualization With Python
- Fundamentions of Machine Learning
- Natural Language Processing
- Deep Learning

Five Star Top Reviews

Check them out ....

Our Byte Academy Presenter: Word on the Street* (+ now your computer)

★★★★★ Attended: Data Sci 201: Data Visualization with Python

Amazing instructor! The lectures were very detailed and comprehensive and the instructor was very well informed with the material and explained it in an easy manner.

★★★★★ Attended: Data Sci 201: Data Visualization with Python

The geospatial data analysis workshop was phenomenal! I didn't even know that tools like plotly and geojsonio existed at all. The curriculum was really well developed and the instructor (Ms. Cordero) did a great job at teaching the concepts and walking through the code. Would recommend to anyone looking to go into data science!

★★★★★ Attended: Python 101: Data Science Prep

Leslie is a diligent teacher, who has enormous knowledge on Python. I definitely would recommend her.

BYTE ACADEMY

*real Course HorseReviews

Questions during the webinar?

- Use the Questions section of the Control Panel
- We'll try the best to answer your questions at the given time
- Also check out our Python v. R blog: http://byteacademy.co/blog/python-vs-r/

- **More questions or follow-up?**
- Visit Byte Academy slack (to join go to www.byteacademy.co + click righthand icon)

- Enroll in classes www.byteacademy.co
- Email info@byteacademy.co

Now Starting...

python-vs-r

July 11, 2017

1 Introduction

Data science is an interdisciplinary field where scientific techniques from statistics, mathematics, and computer science are used to analyze data and solve problems more accurately and effectively. It is no wonder then, that languages such as R and Python, with their extensive packages and libraries that support statistical methods and machine learning algorithms are cornerstones of the data science revolution.

1.1 R

R is an open source, statistical computing language created with the intention of making data analysis, statistical models and graphical models easier. R has a large repository of packages called CRAN that users routinely contribute to.

One of R's main strengths is that it has a very active community that provides ample support to users via mailing lists, StackOverFlow forums, and very extensive documentation of all its packages.

R has a slightly quirky syntax which can be hard to pick up for beginners but is especially suited for people from a statistical and research background looking to get started with creating their models quickly.

1.2 Python

Python is a high level, interpreted, general purpose language that was built to improve program- mer productivity and code readability. It is usually the preferred language for programmers and people with a computer science background looking to get into data analysis.

It's a very flexible language, making it great for production level work and, like R, has libraries of packages around statistics and machine learning in pip, the repository of Python packages. It has great community support, although being a general purpose language it is not all concentrated around data science.

The biggest advantage to using Python is the availability of packages such as Theano, Keras, scikit-learn that are important machine learning and deep learning libraries used by both academic research purposes as well as for commercial intent.

1.3 Choosing

As professional problem solvers, data science practitioners need to have a versatile set of tools as part of their repertoire. While learning both R and Python is ideal, given that R makes data cleaning and manipulation a very easy task while Python is better for building models on larger data sets and scale, we all have to begin somewhere. And the right choice for you can be de- termined by the following factors - previous programming experience, educational background, career aspirations, and interest in working with deep learningtechnology.

1.3.1 Previous Programming Experience

If you have any programming experience prior to learning data science, our recommendation would be for you to learn Python. Its clear syntax would be easy for you to take up; and with it being a general purpose language, you'd have the added flexibility for building novel stuff. Even a complete novice is advised to learn Python, as it is one of the most beginner friendly languages in Computer Science, being the most popular introductory teaching language in the top U.S. universities (Communications of the ACM article, 2014). R code gets to the point more quickly and is less verbose as well, but it has a quirky syntax that would be difficult to learn for both hardcore programmers and beginners alike. We recommend this course for those interested in learning Python programming.

1.3.2 Educational Background

Having a background in statistics or mathematics makes R a better choice for you. This is because R is a domain specific language created specifically for statistics, making its usage intuitive for people with a degree in statistics. R was created by statisticians and made with other statisticians in mind, so having a grasp of statistical analysis makes the transition into this language all the more easy.

1.3.3 Career Aspirations

As a data analyst/business analyst/financial analyst, your focus would be on extracting the most information out of your data, without needing to create a product out of your content. For this reason, learning R and a database language like SQL would serve you better as R is great for working with tabular data on a single system/server and has great libraries like ggplot2 for easy visualizations.

But a data scientist has different requirements, as they're expected to carry out analysis as well as create products such as machine learning engines that work on the database of a website or a software. This would require both software development as well as predictive modelling work which can be better accomplished by a general purpose language like Python. These principles would apply across all industries.

1.3.4 Interest in Deep Learning

Deep Learning is the trending topic du jour and anyone with an interest in contributing to the growth of artificial intelligence technology should be learning Python. Its overwhelming popularity for both machine learning as well as deep learning comes from the fact that

Python acts as an interface between the programmer and lower level languages like C/C++, this making it very easy for experimenting, creating models and debugging without compromising on computational speed (as the machine uses C/C++and CUDA technology to build the models). This makes Python a very accessible language for mathematicians and statisticians looking to create neural network models without having to start creating them from scratch due to the pre-existing frameworks provided by Python.

2 Training & Test Data

Since we'll be doing supervised machine learning later in this workshop, we'll need to split the data into training and testing sets so we don't overfit and are able to test foraccuracy.

```
In [ ]: trainRowCount <- floor(0.8 * nrow(nba)) set.seed(1)
        trainIndex <- sample(1:nrow(nba), trainRowCount) train  <- nba[trainIndex,]
        test <- nba[-trainIndex,]
```

```
In [ ]: train = nba.sample(frac=0.8, random_state=1) test =
        nba.loc[~nba.index.isin(train.index)]
```

Notice that R has more data analysis focused built-in functions, like floor, sample, and set.seed, whereas these are called through packages in Python (math.floor, random.sample, ran- dom.seed).

3 Challenge

Using Python, read the file titanic.csv as a DataFrame. Split this data frame into two individual DataFrames train and test, containing 70% and 30% of the original data, respectively. Set the random state value to 9.

4 Data Preparation & Basic Functionality

Regardless of the language you use, its ability to handle, read, and clean is crucial to the field of data science. In this section we'll review the different ways each language handles basic data preparation.

4.1 Reading the Data

The data is located within a csv file, so we'll start off by reading the data so that we can we can perform analysis later in this tutorial.

4.1.1 CSV Files

The following snippet of code uses the pandas module to easily open and read the file.

```
In [ ]: import pandas
        nba = pandas.read_csv("nba_2013.csv")
```

Meanwhile, in R, we can do this in one line:

```
In [ ]: nba <- read.csv("nba_2013.csv")
```

The only real difference is that in Python, we need to import the pandas library to get access to Dataframes. Dataframes are available in both R and Python, and are two-dimensional arrays (matrices) where each column can be of a different datatype. At the end of this step, the csv file has been loaded by both languages into a dataframe.

4.1.2 Viewing the Data

Now, let's take a look at the actual data through Python and R functionality. First, let's take a look at the header column and its first 5 rows.

In Python, we do this with:

```
In [ ]: nba.head(5)
```

Similarly, in R:

```
In [ ]: head(nba, 5)
```

Pretty straightforward!

4.1.3 Simple Stats

One very simple thing we can do in just one line is mean of each attribute:

```
In [ ]: nba.mean()
```

```
In [ ]: sapply(nba, mean, na.rm=TRUE)
```

In both, we're applying a function across the dataframe columns. But in python, the mean method on dataframes will find the mean of each column by default.

In R, taking the mean of string values will just result in NA – not available. However, we do need to ignore NA values when we take the mean (requiring us to pass na.rm=TRUE into the mean function). If we don't, we end up with NA for the mean of columns like x3p. This column is three point percentage. Some players didn't take three point shots, so their percentage is missing. If we try the mean function in R, we get NA as a response, unless we specify na.rm=TRUE, which ignores NA values when taking the mean. The .mean() method in Python already ignores these values by default.

4.1.4 Selecting SpecificColumns

After loading data into a DataFrame, its dimensions can be found using the dim(sample) com- mand. Column names can be found using

```
In [ ]: names(sample)
```

In R, the $ operator is used to select a specific column of the DataFrame. The following example shows a mean operation on salary column of the DataFrame:

```
In [ ]:  mean(sample$salary)
```

5 Challenge

Using R, read the file scores.csv available in the working directory as a DataFrame. Find the mean math score from the dataset.
Hint: Math score is represented by the column math.score.

6 Final Words

Both Python and R have their own strengths and weaknesses. As we'll go into soon, Python is a much more dynamic language whereas R has great statistical support. Combining both language's strengths and weaknesses is the ideal scenario - we'll go into what that looks like soon.

6.0.1 R Statistical Support

R was built as a statistical language, and it shows. statsmodels in Python and other packages provide decent coverage for statistical methods, but the R ecosystem is far larger. R has a rich ecosystem of cutting-edge packages and active community. Packages are available at CRAN, Bio- Conductor and Github. You can search through all R packages at Rdocumentation.

6.0.2 Python Non-Statistical Support

Python is a general purpose language that is easy and intuitive. This gives it a relatively flat learning curve, and it increases the speed at which you can write a program.

Furthermore, the Python testing framework is a built-in, low-barrier-to-entry testing frame- work that encourages good test coverage. This guarantees your code is reusable and dependable.

6.0.3 Visualizations

Visualizations are an important criteria when choosing data analysis software. Although Python has some nice visualization libraries, such as Seaborn, Bokeh and Pygal, there are maybe too many options to choose from. Moreover, compared to R, visualizations are usually more convoluted, and the results are not always so pleasing to the eye.

In R, visualized data can often be understood more efficiently and effectively than the raw numbers alone. R and visualization are a perfect match. Some must-see visualization packages are ggplot2, ggvis, googleVis and rCharts.

6.0.4 Speed

R was developed for statisticians, so its code is not necessarily the most efficient. Although R can be experienced as slow, there are multiple packages to improve R's performance: pqR, renjin and FastR, Riposte and more.

6.0.5 When do we use each then?

Python Use Python when your data analysis needs to be integrated with web apps or if your statistical code needs to be incorporated into a production database. Being a dynamic programming language, it's a great tool to implement algorithms for production use. If your data analysis needs integration with a web application or database, Python is probably your best bet. Compared to R, the support for these sorts of application is much better since it's more of a general-purpose language.

R is mainly for when the data analysis task requires standalone computing. It's great for exploratory work and for almost any type of data analysis because of the huge number of packages and readily usable tests that often provide you with the necessary tools to get up and running quickly. R can even be part of a big data solution.

When getting started with R, a good first step is to install RStudio. Once this is done, you should continue to have a look at the following packages:

- dplyr, plyr and data.table to easily manipulate packages,
- stringr to manipulate strings,
- zoo to work with regular and irregular time series,
- ggvis, lattice, and ggplot2 to visualize data, and
- caret for machine learning

In [ ]:

Want to learn more Python or Data Science?

Full, Part-Time, Remote programs for all skill levels at www.byteacademy.co

Other Offerings

Data Science

Learn data acquisition, data analysis, Pandas, prediction and machine learning, statistical modeling, Hadoop, SQL, NoSQL and more.

PythonFullstack Development

Full-stack software development with Python, JavaScript, SQL, HTML, CSS, Flask & Django.

Fintech / BlockChain

Learn finance & fundamentals of full-stack software development and apply it to real-world Fintech or BlockChain project.

Quant Algos

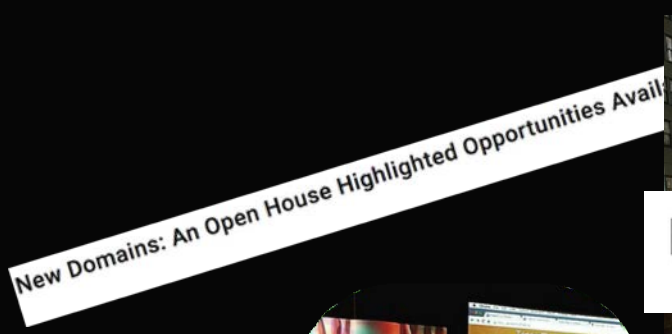
Become a Rocket Scientist of Wall Street
Learn how to model time series for all industries

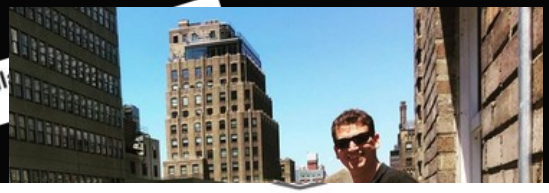
Engage with the Community

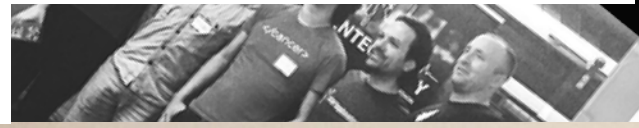
https://www.meetup.com/Byte-Academy-Finance-and-Technology-community/

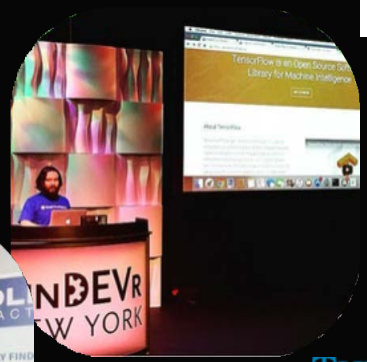
To Join Slack Channel: go to www.byteacademy.co, click on upper righthand icon

Hackathons, events, startup speaker series + more worldwide









THE BUSINESS TIMES

Contact

295 Madison Ave | NYC | 10017

www.byteacademy.co
info@byteacademy.co

@byteacademyco

@byteacademy

Byte-Academy-Finance-and-Technology-community

Byte-Academy-Finance-and-Technology-community
byteacademy2.slack.com

We welcome questions about our curriculum,
hiring developers + partner opportunities


THANK YOU