Why we should learn Python

Kevin Lee, Clindata Insight, PA

ABSTRACT

Python is one of the most popular language nowadays. Python can be used to build just about anything, and it is a great language for data analysis, scientific computing, application development, backend web development, especially machine learning and many more. Python is currently featured in 70% of introductory programming courses at US universities and the latest report from Forbes states that Python grew by more than 450 percent in 2017.

We, statistical programmers, have been using SAS ® and because of the popularity of Python, we wonder if we should learn Python. The paper will start with the current Python implementation and the future of its implementation. The paper will also show basic concepts of Python programming, similarities with SAS programming and difference from SAS programming. The paper will also introduce the benefits of learning Python including the opportunity in Data Science and Machine Learning, career opportunities and a high salary. And also, the paper will discuss the weakness of Python such as regulatory restriction and a lack of metadata. Finally, the paper will discuss the future of statistical programming in the use of Python programming.

INTRODUCTION OF PYTHON

Python language was founded in 1991 by Guido Van Rossum and has been developed by Python Software Foundation. Python have two major versions – Python 2 and Python 3. The most recent version is 3.7.3. Python is considered as general-purpose language and mainly used for web development (server-side), software development, mathematics and system scripting.

BENEFITS OF LEARNING PYTHON

Data Science

"Data Science" is the single, biggest reasons that programmers want to learn Python. Many programmers who know other languages learned Python because they want to move into more exciting work and higher pay in Data Science field. Obviously, Python language is one of the most important tools for data scientists.

Machine Learning Programming

Machine Learning and AI are changing the landscape of all the industry now. The potential for machine learning is vast, and not even close to be fulfilled. Python is one of the most important Machine Learning language, if not the best. All the major Machine Learning packages including sklearn, tensorflow, keras and pytorch work primarily with Python and the data analysis and pre-processing required for Machine Learning go well with Python. A good grasp of Python programming will put anyone a step ahead of others learning Machine Learning from scratch.

Multi-purpose

Python is like a Swiss knife. Python is a general purpose program and can be used for many things – web development, software development, data analysis and Machine Learning. With Python, ones can do many different things.

Big Name Companies Support Python

Python is already used by some of the biggest technology companies as one of main languages – Uber, PayPal, Google, Facebook, Instagram, Netflix and many more. Many technology companies are using Python for their development and testing areas.

Great Community Support

Due to its popularity, Python has good online support from programming community. Anyone who needs programming support or has questions can find the necessary information or get support. There are also a lot of tutorials and courses available for anyone who wants to learn Python programming.

Free
Python is not only free to download, but it is open-sourced. The python project is entirely open-sourced and GPL (General Public License) compatible, which guarantees end users the freedom to run, study, share and modify the software. Python belongs to the public, truly an open-source language. Python's no strings-attached status makes it a perfect tool for all to learn and use.

Easy to learn
Python is considered as easy to learn, especially for those who already knew about programming. Python is readable and simple, and it is easy to set up. Since Jupyter notebook is developed, its usage in data science and Machine Learning becomes whole lot easier. This is the reason that 70% of introductory courses in universities are using Python.

Careers and Growth
Python is growing really fast and big time. Based on Dice jobs database (below graph), python developer jobs are spiking lately, and it does not seem to level off any time soon thanks to its popularity in Machine Learning.

PYTHON PROGRAMMING USE CASES

The programmer can use Python language for pretty much all the data analysis jobs.

Import SAS datasets
The programmers can import SAS datasets using Python Pandas read_sas function in Jupyter notebook IDE (Integrated Development Environment).

The data scientists can import ADSL datasets in Jupyter notebook. To import ADSL datasets, the data scientist should import pandas package. Pandas is a Python package providing fast, flexible, and expressive data structure designed to make working relational data both easy and intuitive. Since ADSL is tabular data structured, it can be easily imported using Python Pandas.

```
### Import Pandas and Numpy package
import pandas as pd
import numpy as np
```

```
## Read ADSL datasets
adsl = pd.read_sas('./data/SAS/ADaM/adsl.xpt')
adsl.head()
```

	USUBJID	STUDYID	DOMAIN	SITEID	SITEGRP	SUBJID	VISIT1DT	RANDDT	TRTSTDT	RFSTDTC	...
0	b'01-701-1015'	b'CDISCPILOT01'	b'ADSL'	b'701'	b'701'	b'1015'	19718.0	19725.0	19725.0	b'2014-01-02'	...
1	b'01-701-1023'	b'CDISCPILOT01'	b'ADSL'	b'701'	b'701'	b'1023'	19196.0	19210.0	19210.0	b'2012-08-05'	...
2	b'01-701-1028'	b'CDISCPILOT01'	b'ADSL'	b'701'	b'701'	b'1028'	19550.0	19558.0	19558.0	b'2013-07-19'	...
3	b'01-701-1033'	b'CDISCPILOT01'	b'ADSL'	b'701'	b'701'	b'1033'	19792.0	19800.0	19800.0	b'2014-03-18'	...
4	b'01-701-1034'	b'CDISCPILOT01'	b'ADSL'	b'701'	b'701'	b'1034'	19898.0	19905.0	19905.0	b'2014-07-01'	...

5 rows × 51 columns

Data Preparation and manipulation

Programmers and data scientists can use Python language pretty much all the data analysis.

The programmers can create new variables.

```
In [144]: ### Prepare variables
df_dm2['SEX'] = df_dm2.SEXC.replace(['Male','Female'], ['M','F']) # Create SEX variable
df_dm2['DOMAIN'], df_dm2['COUNTRY'] = 'DM', 'USA' # Create varialbes DOMAIN & COUNTRY
df_dm2['USUBJID'] = df_dm2.STUDYID + '-' + df_dm2.SITEID + '-' + df_dm2.SUBJID  # create USUBJID
```

The programmers can also merge two datasets using key variable using left-join.

Merge dispostion data to demog data

```
In [152]: df_dm5 = pd.merge(df_dm4, df_ds4, on='SUBJID', how='left')
```

Statistical analysis

The data scientist can conduct statistical analysis. In below codes, programmers can find the mean values of AGE by RACE group using groupby function.

```
In [34]: ### Calculate the means of age by Race Group
mean_age = adsl.groupby(['RACE'])['AGE'].mean()
print(mean_age)
```

```
RACE
b'AFRICAN DESCENT (NEGRO, BLACK)'                                  72.869565
b'CAUCASIAN'                                                       75.793578
b'HISPANIC (MEXICAN - AMERICAN, MEXICO, CENTRAL AND SOUTH AMERICA)'  67.666667
b'OTHER (MIXED - RACIAL HERITAGE, AMERICAN INDIAN, ESKIMO)'        61.000000
Name: AGE, dtype: float64
```

Export SAS datasets

After all data analysis, programmers can save the finished dataset in local drive.

Write CDISC DM dataset

```
In [154]: with open('./data/cdisc/dm.xpt', 'wb') as f:
              xport.from_dataframe(df_dm6, f)## write DM
```

Data Visualization

The data scientist imports matplotlib for data visualization. Matplotlib is a Python plotting library which provides line plot, histogram, scatter plot and many more. It is one of the most popular graphic packages.

```
### Import matplotlib for data visualization
import matplotlib
import matplotlib.pyplot as plt

### Horizontal bar the means of age by Race Group
plt.title('Histogram of RACE')
plt.barh( mean_age.index.values, mean_age)
```

```
<Container object of 4 artists>
```

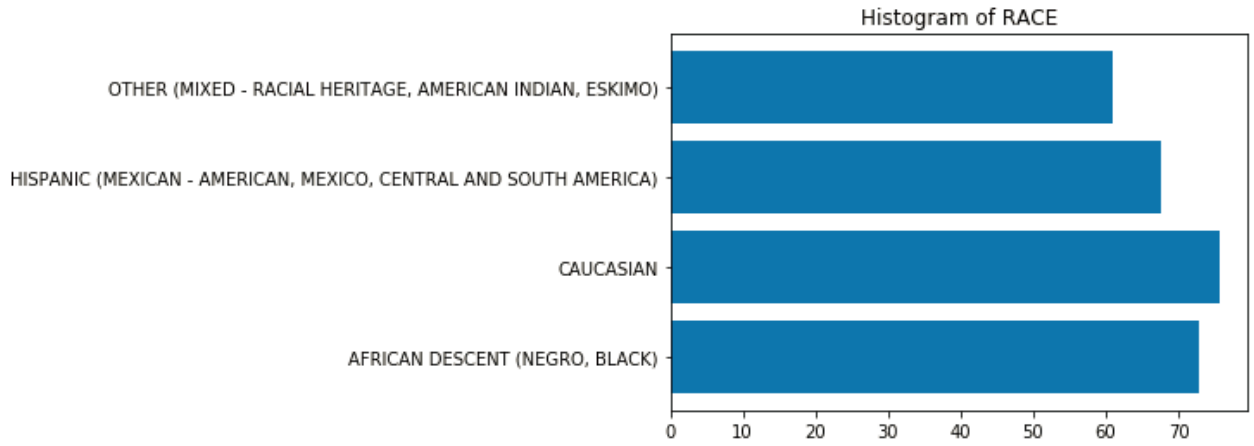


Figure 1. Histogram of RACE using matplotlib

Programmers can also create Kaplan Meier Curves. First, programmers import Time to Event ADTTEOS datasets in Jupyter notebook environment.

```
In [10]: #import Time to Event ADaM datsets
adtteos = pd.read_sas('./data/SAS/ADaM/adtteos.sas7bdat')
adtteos
```

Out[10]:

	SUBJID	SITEID	FASFL	SAFFL	TRTP	TRTPN	PARAM	PARAMCD	PARAMTYP	AVISIT	AVISITN	AVAL	STARTDT	ADT	ADTF	CNSR	EVNT
0	b'310-001'	b'310'	b'Y'	b'Y'	b'Control'	1.0	b'Days to Death'	b'DEATH'	b'DERIVED'	b'Double-Blind Period'	91.0	1.0	2007-11-12	2007-11-12	NaN	0.0	b'D
1	b'310-002'	b'310'	b'Y'	b'Y'	b'Control'	1.0	b'Days to Death'	b'DEATH'	b'DERIVED'	b'Double-Blind Period'	91.0	83.0	2008-01-11	2008-04-02	NaN	1.0	b'COMP PE WITI E
2	b'310-003'	b'310'	b'Y'	b'Y'	b'Study Drug'	2.0	b'Days to Death'	b'DEATH'	b'DERIVED'	b'Double-Blind Period'	91.0	84.0	2008-02-01	2008-04-24	NaN	0.0	b'D
3	b'310-004'	b'310'	b'Y'	b'Y'	b'Study Drug'	2.0	b'Days to Death'	b'DEATH'	b'DERIVED'	b'Double-Blind Period'	91.0	9.0	2008-02-07	2008-02-15	NaN	0.0	b'D
4	b'310-005'	b'310'	b'Y'	b'Y'	b'Study Drug'	2.0	b'Days to Death'	b'DEATH'	b'DERIVED'	b'Double-Blind Period'	91.0	51.0	2008-02-21	2008-04-11	NaN	0.0	b'D

Next, programmers prepare datasets for data visualization.

```
In [13]: ### ADTTEOS with Placebo
adtteos_c = adtteos[adtteos.TRTP == b'Control']

### ADTTEOS with Study Drug
adtteos_sd = adtteos[adtteos.TRTP == b'Study Drug']

print(adtteos_c.head())
print(adtteos_sd.head())
print(adtteos_c.describe())
print(adtteos_sd.describe())
```

```
        SUBJID  SITEID FASFL SAFFL        TRTP  TRTPN            PARAM  \
0   b'310-001'  b'310'  b'Y'  b'Y'  b'Control'    1.0  b'Days to Death'
1   b'310-002'  b'310'  b'Y'  b'Y'  b'Control'    1.0  b'Days to Death'
5   b'310-006'  b'310'  b'Y'  b'Y'  b'Control'    1.0  b'Days to Death'
7   b'310-008'  b'310'  b'Y'  b'Y'  b'Control'    1.0  b'Days to Death'
10  b'310-011'  b'310'  b'Y'  b'Y'  b'Control'    1.0  b'Days to Death'

     PARAMCD    PARAMTYP                 AVISIT  AVISITN  AVAL    STARTDT  \
0   b'DEATH'  b'DERIVED'  b'Double-Blind Period'     91.0   1.0 2007-11-12
1   b'DEATH'  b'DERIVED'  b'Double-Blind Period'     91.0  83.0 2008-01-11
5   b'DEATH'  b'DERIVED'  b'Double-Blind Period'     91.0  62.0 2008-03-10
7   b'DEATH'  b'DERIVED'  b'Double-Blind Period'     91.0   3.0 2008-07-11
10  b'DEATH'  b'DERIVED'  b'Double-Blind Period'     91.0  85.0 2009-02-24
```

Programmers use pre-processed data to create Kaplan Meier curves using KaplanMeierFitter library

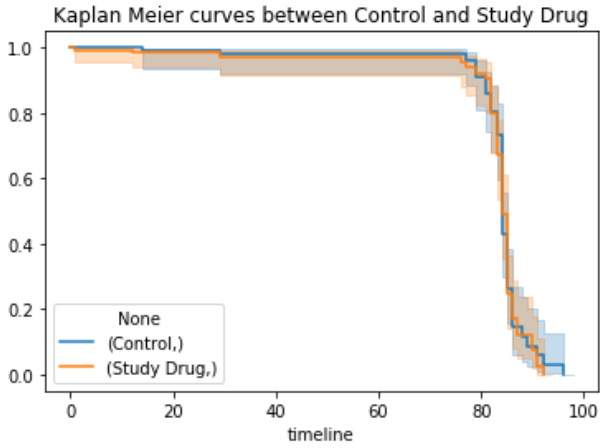
In [31]:
```
### Import lifelines and Kaplan Meier Curves
from lifelines.estimation import KaplanMeierFitter
kmf = KaplanMeierFitter()

### Prepare Kaplan Meier Curves
ax = plt.subplot(111)

kmf.fit(adtteos_c.AVAL, event_observed=adtteos_c.CNSR, label=['Control'])
kmf.plot(ax=ax)
kmf.fit(adtteos_sd.AVAL, event_observed=adtteos_sd.CNSR, label=['Study Drug'])
kmf.plot(ax=ax)

plt.title('Kaplan Meier curves between Control and Study Drug')
```

Out[31]:
```
Text(0.5,1,'Kaplan Meier curves between Control and Study Drug')
```



Convert image to numeric record

One of the reasons that Python is used for all purpose language is that Python can read image data. This is a big advantage especially in Computer Image Recognition.

Data Scientist can convert image to numeric number. Below image 0 can be converted into the number.

```
### import MNST datasets using sklearn API
from sklearn.datasets import fetch_mldata
mnist = fetch_mldata('MNIST original')
mnist
```

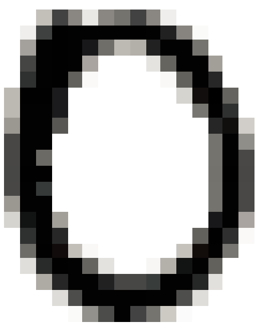
```
{'COL_NAMES': ['label', 'data'],
 'DESCR': 'mldata.org dataset: mnist-original',
 'data': array([[0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0],
        ...,
        [0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0],
        [0, 0, 0, ..., 0, 0, 0]], dtype=uint8),
 'target': array([0., 0., 0., ..., 9., 9., 9.])}
```

```
### prepare X and Y dataset
X, y = mnist["data"], mnist["target"]
print('Shape of X: ', X.shape)
print('Shape of y: ', y.shape, 'value of y: ', y[1200])

### Pick one image
digit_image1 = X[1200].reshape(28, 28)

### Show that image
plt.imshow(digit_image1, cmap = matplotlib.cm.binary,
           interpolation="nearest")
plt.axis("off") ## remove the outside frame
plt.show() # show plot
```

```
Shape of X:  (70000, 784)
Shape of y:  (70000,) value of y:  0.0
```



The data scientist can display the image of 0 using raw data.

```
### print that image
print('Shape of image: ', digit_image1.shape)
print(digit_image1)
```

```
Shape of image:  (28, 28)
```

Below is 28 by 28 pixel of digit image1. As you see it, it is actually 28 * 28 numbers. We can see it as an image using matplotlib package.

```
[[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0  64 191 118  24 118 138 191 118  17   0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0  17 193 253 252 252 252 252 253 252 227 120   5   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0  93 252 253 252 233 141  69  79 227 252 252 137   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0 187 252 253 252  89   0   0   0  29 154 252 252  95   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0 207 252 253 157   6   0   0   0   0   7 158 252 220   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0  70 253 253 231   0   0   0   0   0   0   0  43 241 255 154   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0  70 252 252 230   0   0   0   0   0   0   0   0 135 253 206   0   0   0   0   0   0]
```

```
[  0   0   0   0   0   0   0 101 252 252 167   0   0   0   0   0   0   0   0   0 222 240  50   0   0   0   0   0]
[  0   0   0   0   0   0   0 184 252 252   0   0   0   0   0   0   0   0   0   0 138 252 111   0   0   0   0   0]
[  0   0   0   0   0   0   0 184 252 147   0   0   0   0   0   0   0   0   0   0 138 252 183   0   0   0   0   0]
[  0   0   0   0   0   0   0 185 253 253   0   0   0   0   0   0   0   0   0   0 139 253 184   0   0   0   0   0]
[  0   0   0   0   0   0   0 184 252 200   0   0   0   0   0   0   0   0   0   0 138 252 183   0   0   0   0   0]
[  0   0   0   0   0   0   0 101 252 252   0   0   0   0   0   0   0   0   0   0 138 252 183   0   0   0   0   0]
[  0   0   0   0   0   0   0  44 236 252  95   0   0   0   0   0   0   0   0   0 233 252  89   0   0   0   0   0]
[  0   0   0   0   0   0   0   0 207 252 220   0   0   0   0   0   0   0   0 158 253 210   6   0   0   0   0   0]
[  0   0   0   0   0   0   0   0 145 253 244  65   7   0   0   0   0   0  62 243 252 135   0   0   0   0   0   0]
[  0   0   0   0   0   0   0   0  34 217 253 252 154  30   0   0   9  78 236 252 157   0   0   0   0   0   0   0]
[  0   0   0   0   0   0   0   0   0  84 222 252 252 227 184 185 197 252 252 221  32   0   0   0   0   0   0   0]
[  0   0   0   0   0   0   0   0   0   0  36 177 252 252 252 253 252 252 176  35   0   0   0   0   0   0   0   0]
[  0   0   0   0   0   0   0   0   0   0   0   5 107 179 252 190 137  54   4   0   0   0   0   0   0   0   0   0]
[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
[  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0]
]
```

Machine Learning Implementation

The very reason that many people now learn Python is due to its usage and popularity in Machine Learning. In Machine Learning Programming environment, Python is go-to-language. There are many Machine Learning packages available in Python and its ecosystem is very well suited for Machine Learning Programming.

The data scientist can use sci-kit learn Machine Learning package to train the model.

```
### Prepare test and train data
from sklearn.model_selection import train_test_split
X_train2, X_test2, y_train2, y_test2 = train_test_split(X, y, test_size=0.2, random_state=42)

### import the algorithm
from sklearn.linear_model import SGDClassifier

### Train the model
sgd_clf2 = SGDClassifier(max_iter=5, random_state=42)
sgd_clf2.fit(X_train2, y_train2)
```

```
SGDClassifier(alpha=0.0001, average=False, class_weight=None, epsilon=0.1,
       eta0=0.0, fit_intercept=True, l1_ratio=0.15,
       learning_rate='optimal', loss='hinge', max_iter=5, n_iter=None,
       n_jobs=1, penalty='l2', power_t=0.5, random_state=42, shuffle=True,
       tol=None, verbose=0, warm_start=False)
```

```
### Validate the accuracy
from sklearn.model_selection import cross_val_score
cross_val_score(sgd_clf2, X_train2, y_train2, cv=3, scoring="accuracy")  ### three folding validataion
```

```
array([0.87686127, 0.87126326, 0.84735319])
```

With the trained model, the data scientist can predict the next value.

```
print('predicted y value: ', sgd_clf.predict([X_train2[3333]]))
print('actual y value: ', y_train2[3333])
```

```
predicted y value:  [1.]
actual y value:  1.0
```

COMPARISON WITH SAS

SAS has been the main stream on data analysis programming for a long time and it is still the most effective data analysis program language. However, Python has been growing fast and big time in data analysis programming for last couple of years. SAS is the most effective data analytic platform working on structure

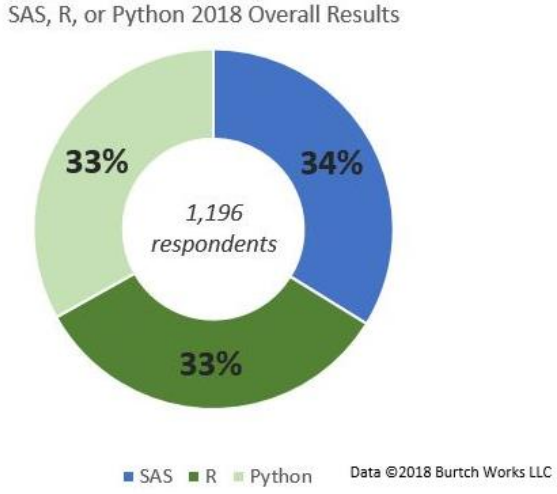
data and Python is the most popular choice for data science and machine learning. Below is the simple comparison between SAS and Python Language.

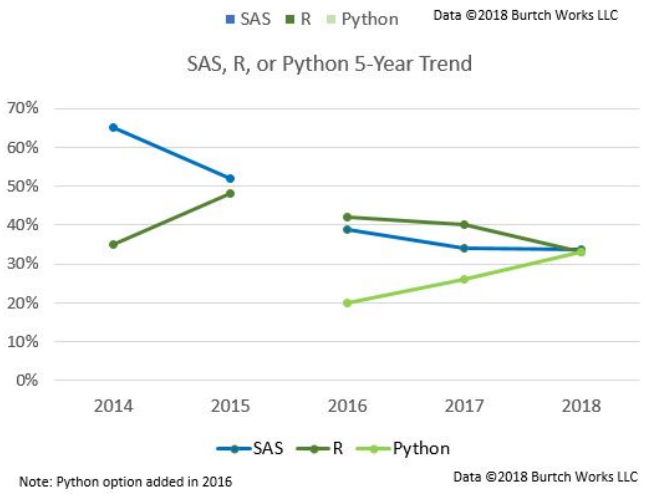
	SAS	Python
Main Usage	Enterprise Analytics	Free
Programming Platform	A good GUI for users	Open source – Jupyter notebook
Support	Technical Customer Support	A huge Python community
Functions/Packages	A huge array of validated statistical functions	Open library
Price	Expensive commercial software	Free to download
Main users	Large corporation with huge budgets	All the users

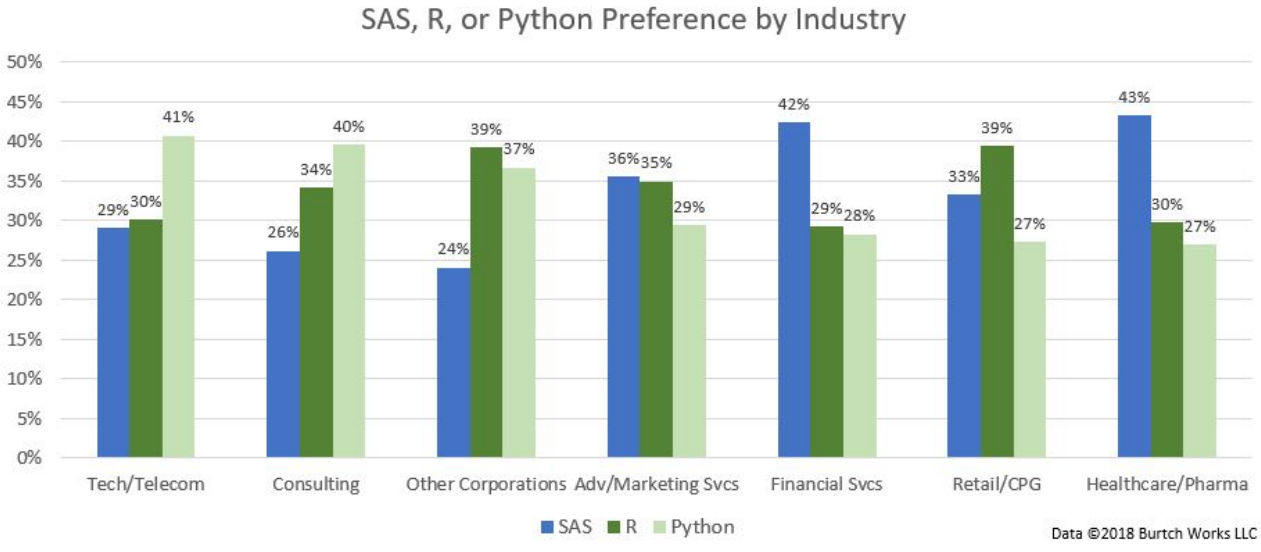
As you see in above, each has its own strength and its own place. That is why many data scientists or programmers want to know and use both programming. Because every problem that they need to solve has the different challenges and approaches, data scientists are better off knowing more than one language.

PROGRAMMING COMPARISON- SAS, R, PYTHON

"Burtch Works LLC" has asked 1000 data scientists on their preference on programming language for last five years. Below are the survey results. Below survey results show where popularity of each programming resides and show how Python has been catching up R and SAS programming.







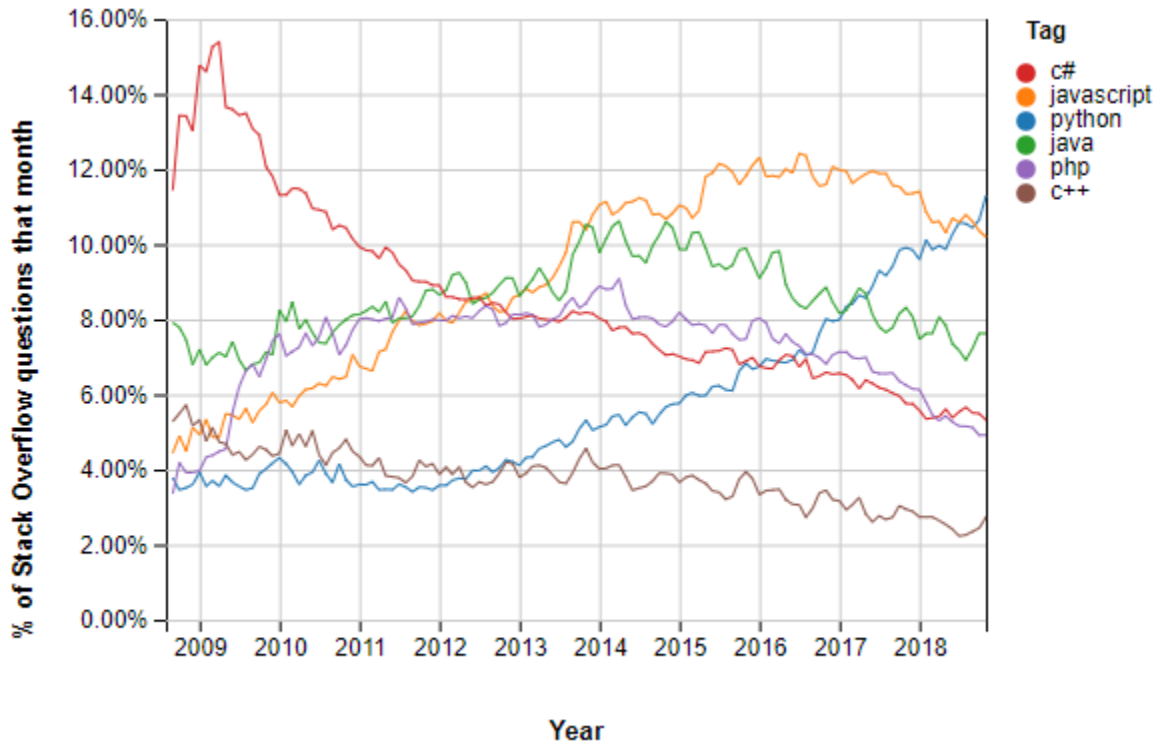
CHALLENGE OF PYTHON

Even though Python programming are getting popularity in programming community, there are a couple of challenges in using Python programming in enterprise level.

- A lack of customer service. Python programming is open source, so there is no enterprise level customer service like SAS. Users need to find out the solution on their own.
- Regulatory challenges. Python is open-source, so no one has fully validated the packages or functions. To get regulatory approval for some algorithm, the burden of algorithm validating will be on the sponsor.
- A lack of talents. Even though Python programming gains a huge popularity in programming world, it is still considered as new and up-and-coming language and there are not many programmers who know now. To fully implement Python programming in enterprise level, the organizations need more Python programmers.

FUTURE OF PYTHON

Nobody can predict the future of Python language, but in general perception, the future of Python is very bright. Based on Stack Overflow Trends, Python is up and coming and its usage is exploding right now.



CONCLUSION

Python programming language is in a great position. It's already one of the most popular languages or skills, and its popularity and usages keep growing up. Especially for those who are interested in data science and Machine Learning fields, Python is a must-know language. As the pharmaceutical industry and biometric department move into data science and Machine Learning area, programmers with Python programming experience and knowledge can grow with the trend.

REFERENCES

https://www.python.org/ Python home page

http://jupyter.org/ Jupyter notebook home page
https://matplotlib.org/ Python matplotlib home page
Python for Data Analysis
https://stackoverflow.com/ Stack Overflow Trends
https://www.dice.com/ Dice Jobs Database
https://www.burtchworks.com/ SAS, R and Python Survey

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Kevin Lee
Director of Data Science at Clindata Insight
Klee@clindatainsight.com
Tweet: @HelloKevinLee
LinkedIn: www.linkedin.com/in/HelloKevinLee/

TRADEMARKS