

# Python

## An Introduction



# What is Python?

- Interpreted high-level programming language for general purpose programming
- Created by [Guido van Rossum](#)
- First released in 1991
- Has a design philosophy that emphasizes code readability
- It supports multiple programming paradigms, including object-oriented, functional and procedural, and has a large and comprehensive standard library



# Why learn Python for Data Science?

- General Purpose interpreted language
- Easy to learn
- Popular among the top 10 programming languages
- Has Libraries for Maths and Machine Learning

# Data Science Libraries in Python

- SciPy
  - NumPy
  - Matplotlib
  - Pandas
- scikit-learn
- Keras
- TensorFlow

# SciPy

- An ecosystem of Python libraries for mathematics, science and engineering
- Comprises of
  - **Numpy**: Supports working with arrays
  - **Matplotlib**: Visualization
  - **Pandas**: Supports organizing and analysing data

# Scikit-learn

- Built on SciPy
- Implements Supervised Learning and Unsupervised Learning Algorithms
- Scikit-learn is usable commercially under the BSD license

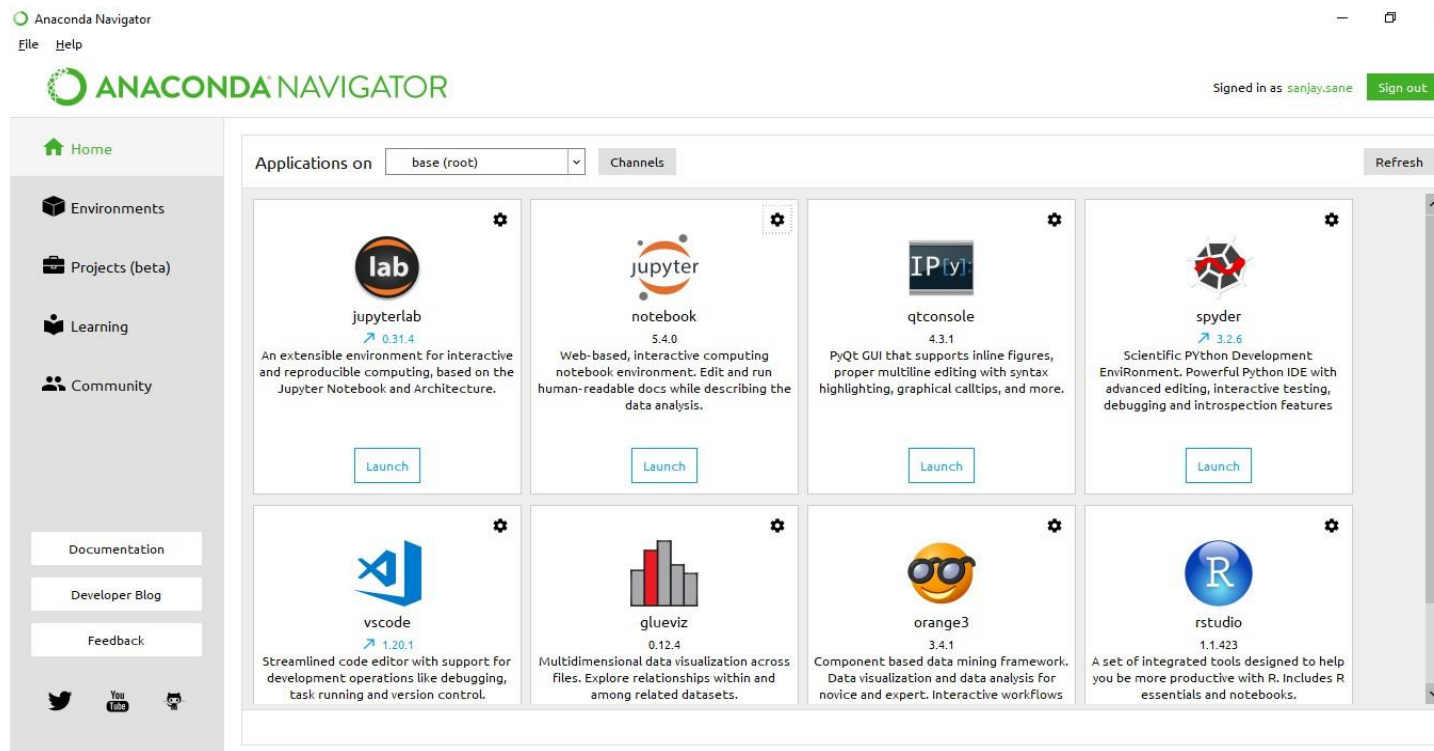
# Anaconda Installation

- Anaconda platform by Anaconda Inc., provides default installation of all the basic libraries and also machine learning library scikit-learn
- Some of its elements which we are going to use are:
  - Spyder IDE
  - VS Code
  - Jupyter Notebook
  - IPython



# Anaconda Navigator

- Anaconda Navigator is a desktop graphical user interface included in Anaconda that allows you to launch applications and easily manage conda packages, environments and channels without the need to use command line commands.
- It provides you the links to the installed Spyder, Jupyter and also VS Code





# Spyder IDE

Spyder (Python 3.6)

File Edit Search Source Run Debug Consoles Projects Tools View Help

Editor - F:\ML with Python\2. Python Fundamentals and Statistical Basics\DescriptiveStats.py

pandas.py Basics.py DescriptiveStats.py

Code Editor

```
17 set_option('precision', 3)
18
19 # Mean
20 stars.mean()
21 stars.mean()[0]
22 np.mean(stars.iloc[:,3])
23
24
25 # Median
26 stars.median()
27 stars.median()[1]
28 from statistics import mode
29 # Mode
30 mode(stars["Gender"])
31
32 # 1st Quartile
33 stars.quantile(0.25)
34
35 # For specific variable
36 stars.quantile(0.25)[1]
37
38 # Multiplie Qunatiles
39 stars.quantile([0.2,0.4,0.6,0.7])
40
41 ## STD
42 stars.std()
43 stars.std()[2]
44
45 (stars.std()[0]/stars.mean()[0])*100
46 (stars.std()[1]/stars.mean()[1])*100
47
48
49 import matplotlib.pyplot as plt
50 plt.hist(stars["Domestic Gross"])
51 stars.skew()[0]
52
53 stars.kurt()[0]
54
```

IPython console

Console 1/A

```
...:                                     dtype=object)

In [49]: from sklearn.metrics import confusion_matrix,
classification_report, accuracy_score
...: print(confusion_matrix(existing, predicted))
[[5 2]
 [3 4]]

In [50]: print(classification_report(existing, predicted))
              precision    recall  f1-score   support

      N         0.62        0.71         0.67         7
      Y         0.67        0.57         0.62         7

avg / total         0.65        0.64         0.64        14

In [51]: print(accuracy_score(existing,predicted))
0.6428571428571429

In [52]: from sklearn.metrics import confusion_matrix,
classification_report, accuracy_score
...: print(confusion_matrix(existing, predicted))
...: print(classification_report(existing, predicted))
...: print(accuracy_score(existing,predicted))
[[5 2]
 [3 4]]
              precision    recall  f1-score   support

      N         0.62        0.71         0.67         7
      Y         0.67        0.57         0.62         7


avg / total         0.65        0.64         0.64        14



0.6428571428571429













In [53]:
```

IPython console Variable explorer File explorer Help Profiler Static code analysis

# Jupyter Notebook

jupyter Untitled9 Last Checkpoint: Yesterday at 10:05 AM (unsaved changes)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted  Python 3 

        Run    Code 

```
In [1]: w = 6 + 9
```

```
In [2]: w
```

```
Out[2]: 15
```

```
In [ ]: |
```

# Thank You

