Data Application Lab

Analytics Foundation

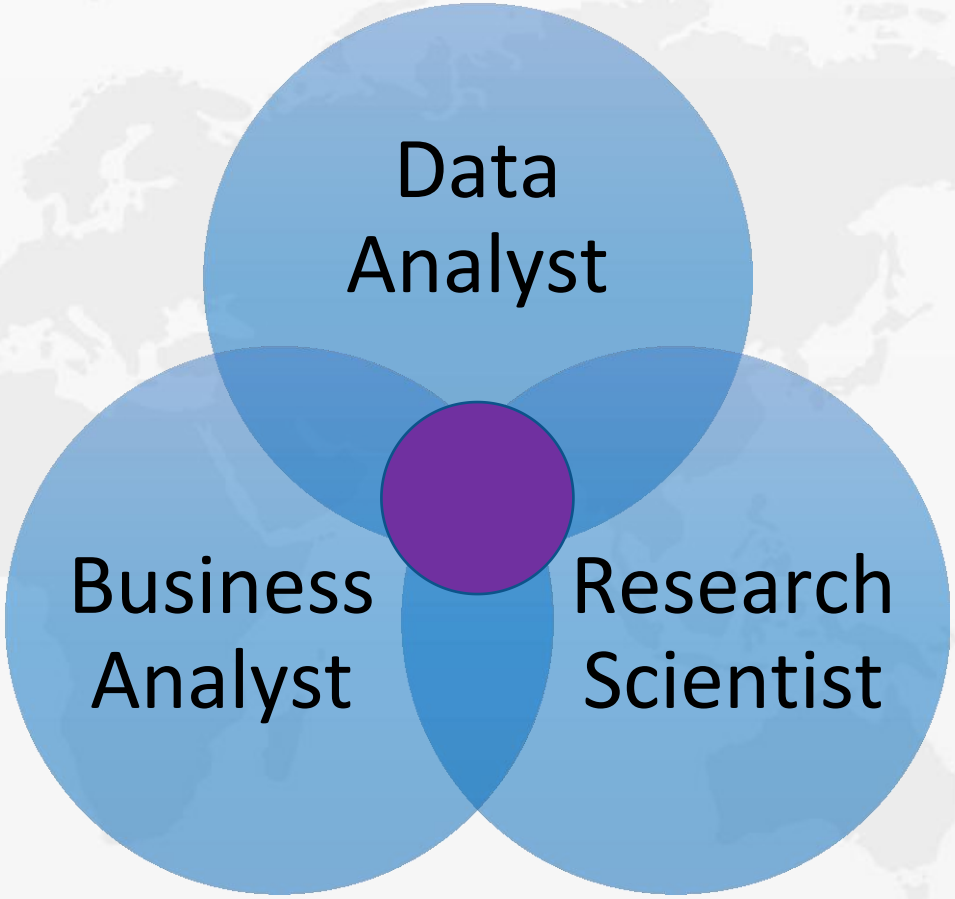
Outline

1. Demystify “Data Scientist”
2. Python key packages
3. Pandas – data analytics
4. Case studies in Python

What is Data Scientist?

1

What is Data Scientist?

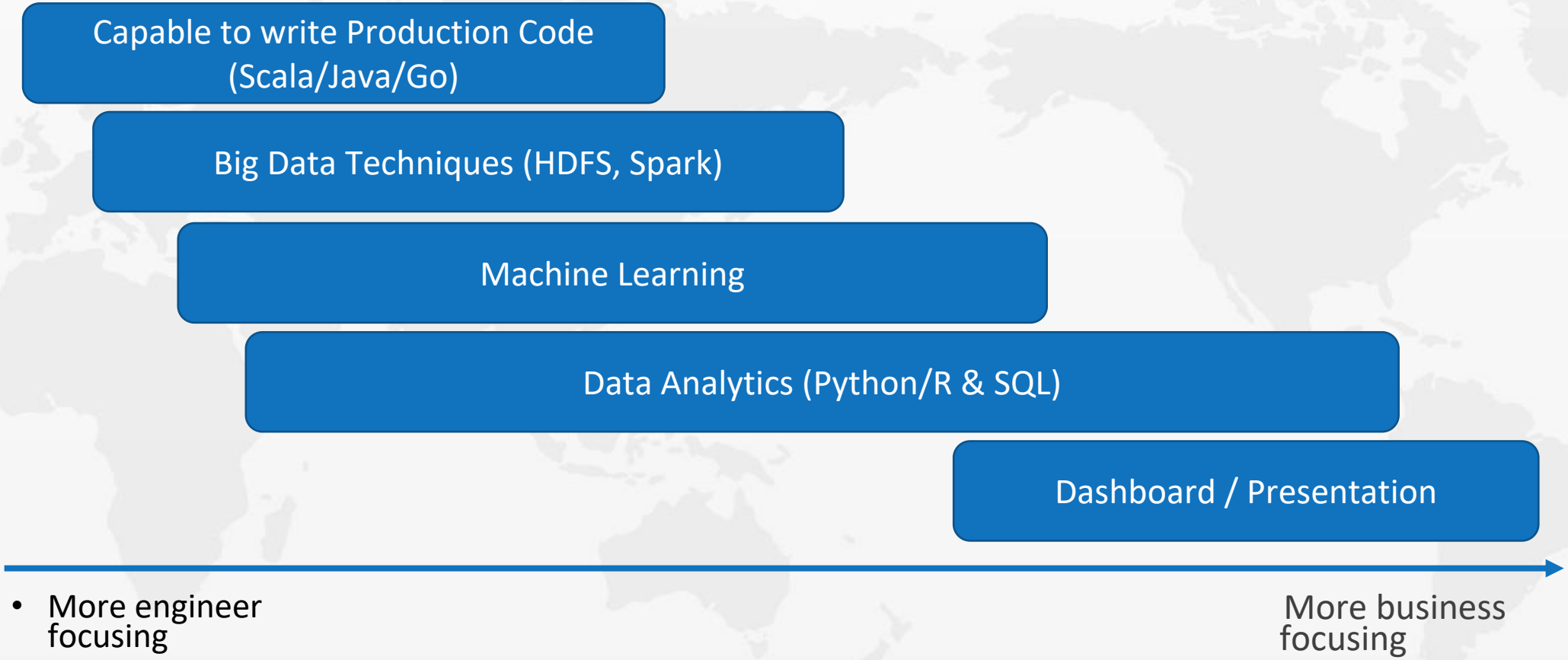


Analytical skills

Design algorithm

Business insight

Data Science Skills Spectrum



Recommended Stack for future DS

Data Science foundation

1. SQL – the universal language for analytics
2. Python – the most popular language for data science

Analytics in the business world

1. Microsoft-Suite/G-Suite
2. Tableau (BI visualization)

Python in Data Science

2

Why Python?

General purpose programming language

Purpose

R focuses on better, user friendly data analysis, statistics and graphical models.

Python emphasizes productivity and code readability.

Used By?

R has been used primarily in academics and research. However, R is rapidly expanding into the enterprise market.

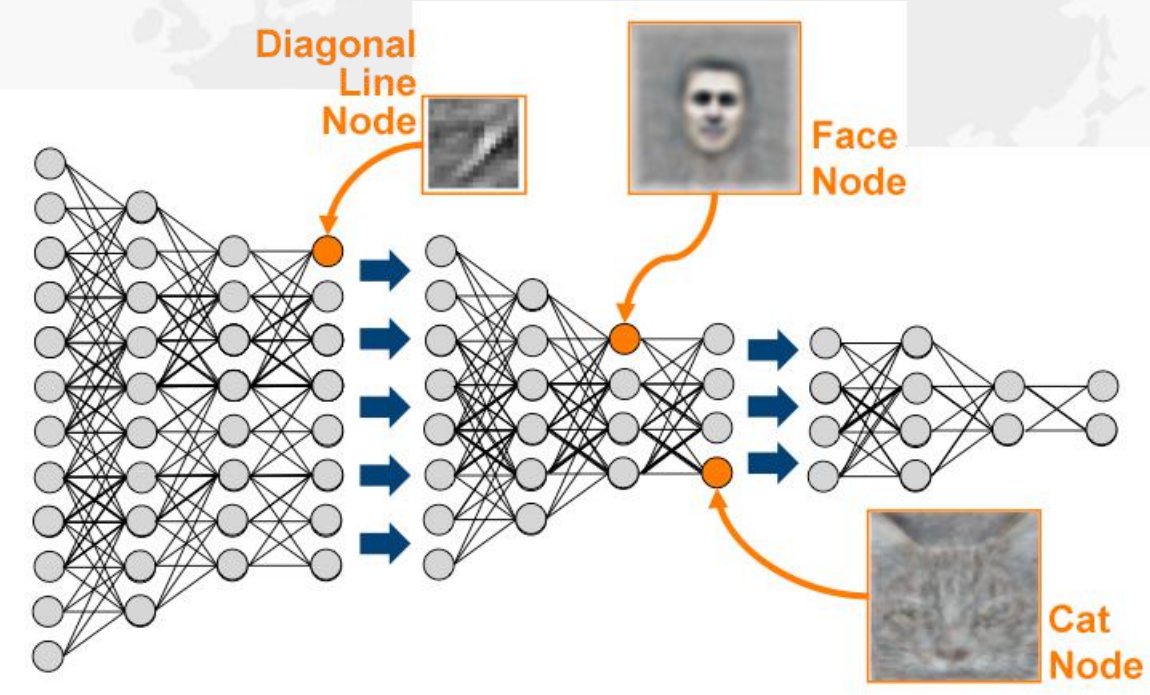
"The closer you are to statistics, research and data science, the more you might prefer R."

Python is used by programmers that want to delve into data analysis or apply statistical techniques, and by developers that turn to data science.

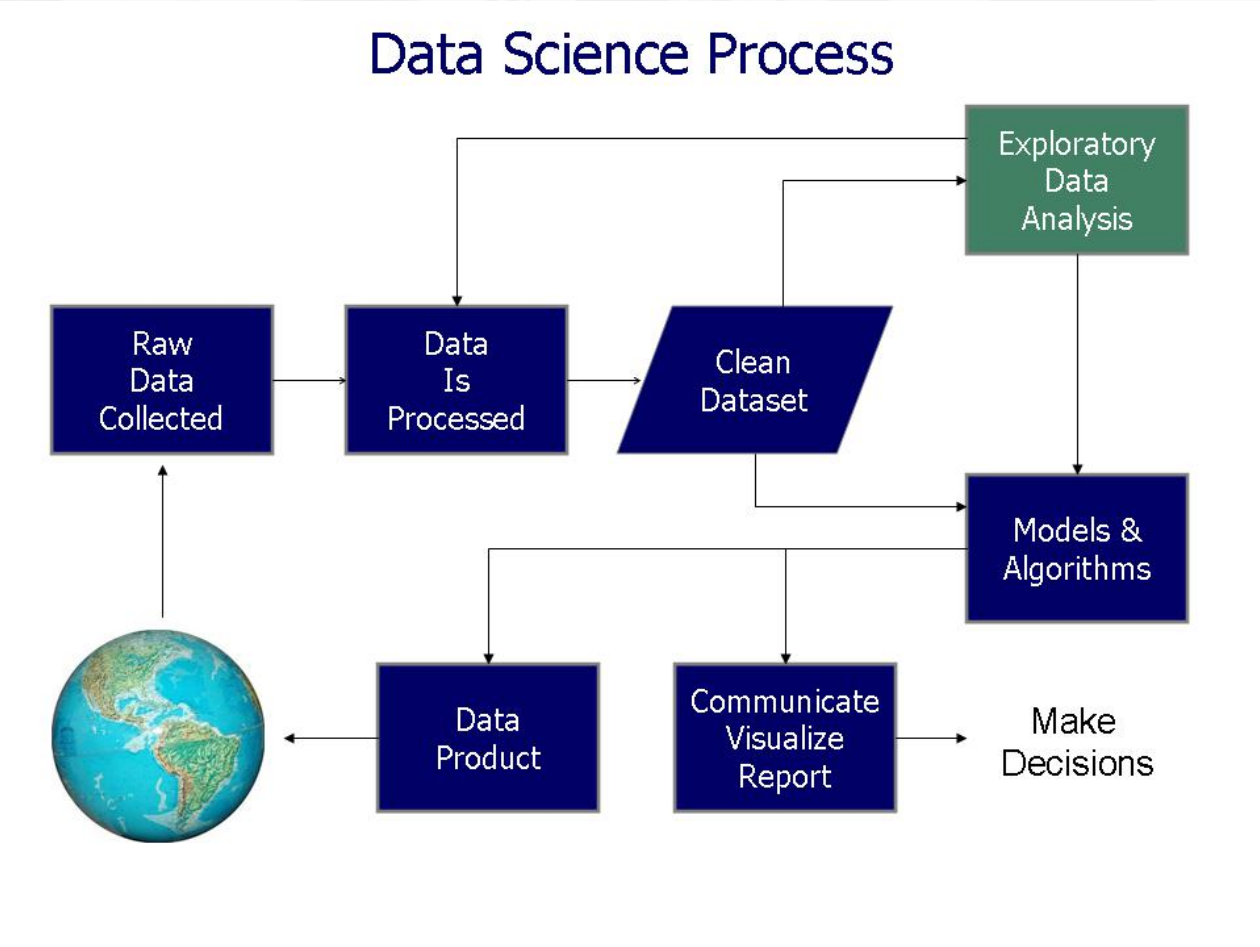
"The closer you are to working in an engineering environment, the more you might prefer Python."

Life as a Data Scientist

"Ideal"

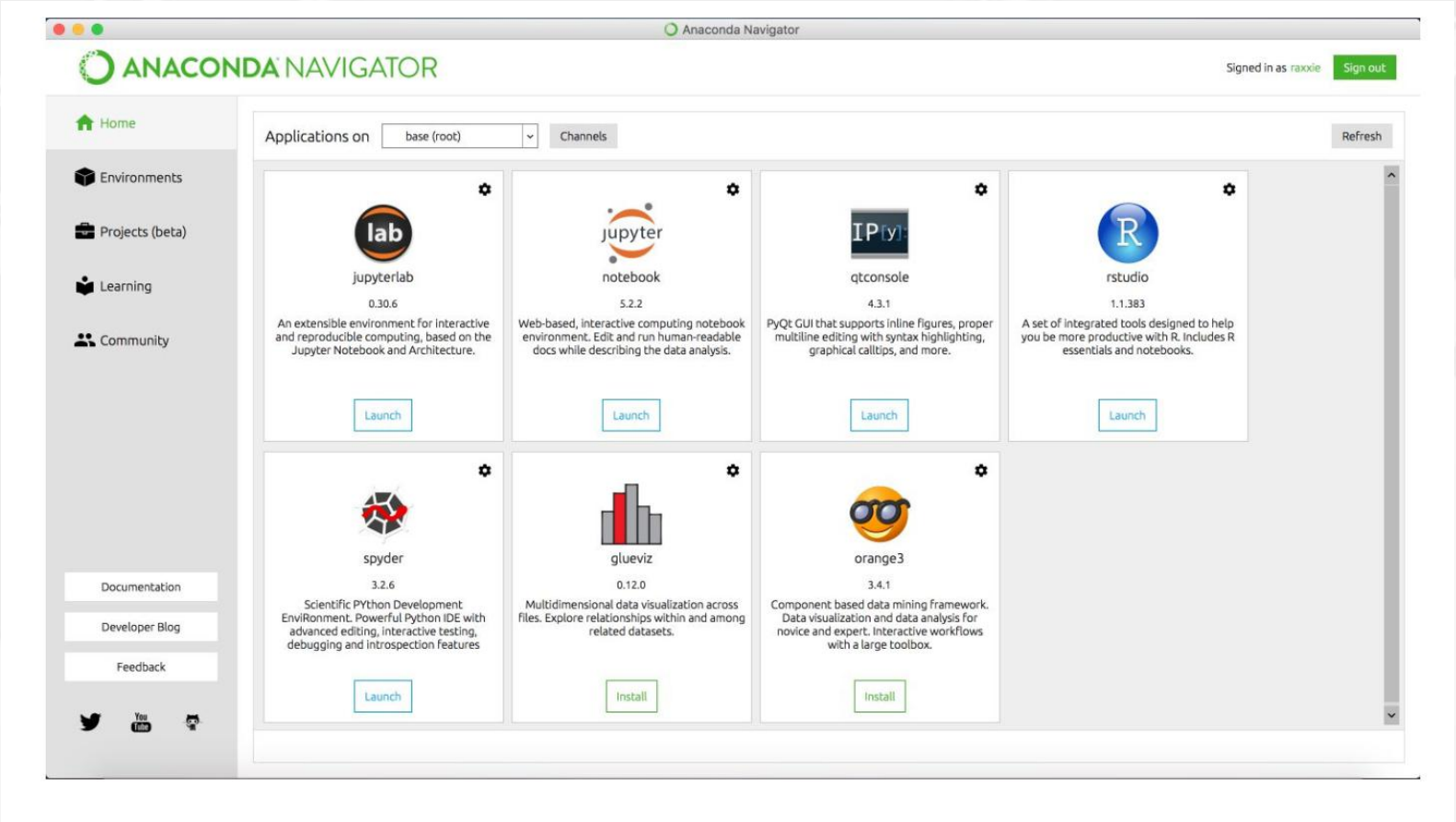


"Reality"

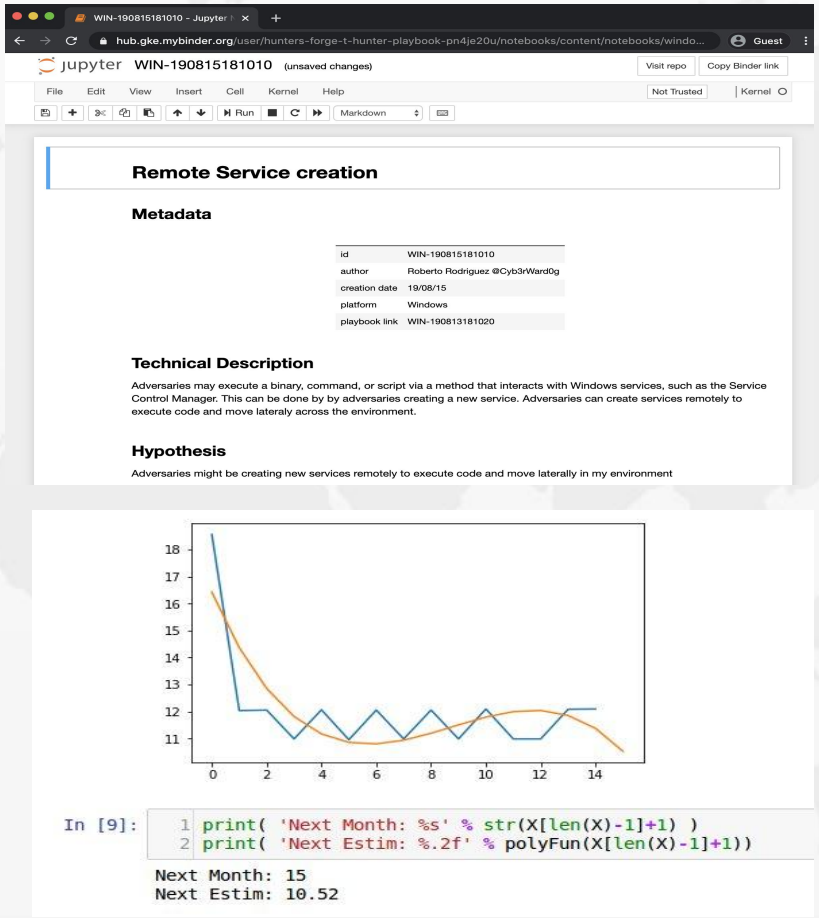


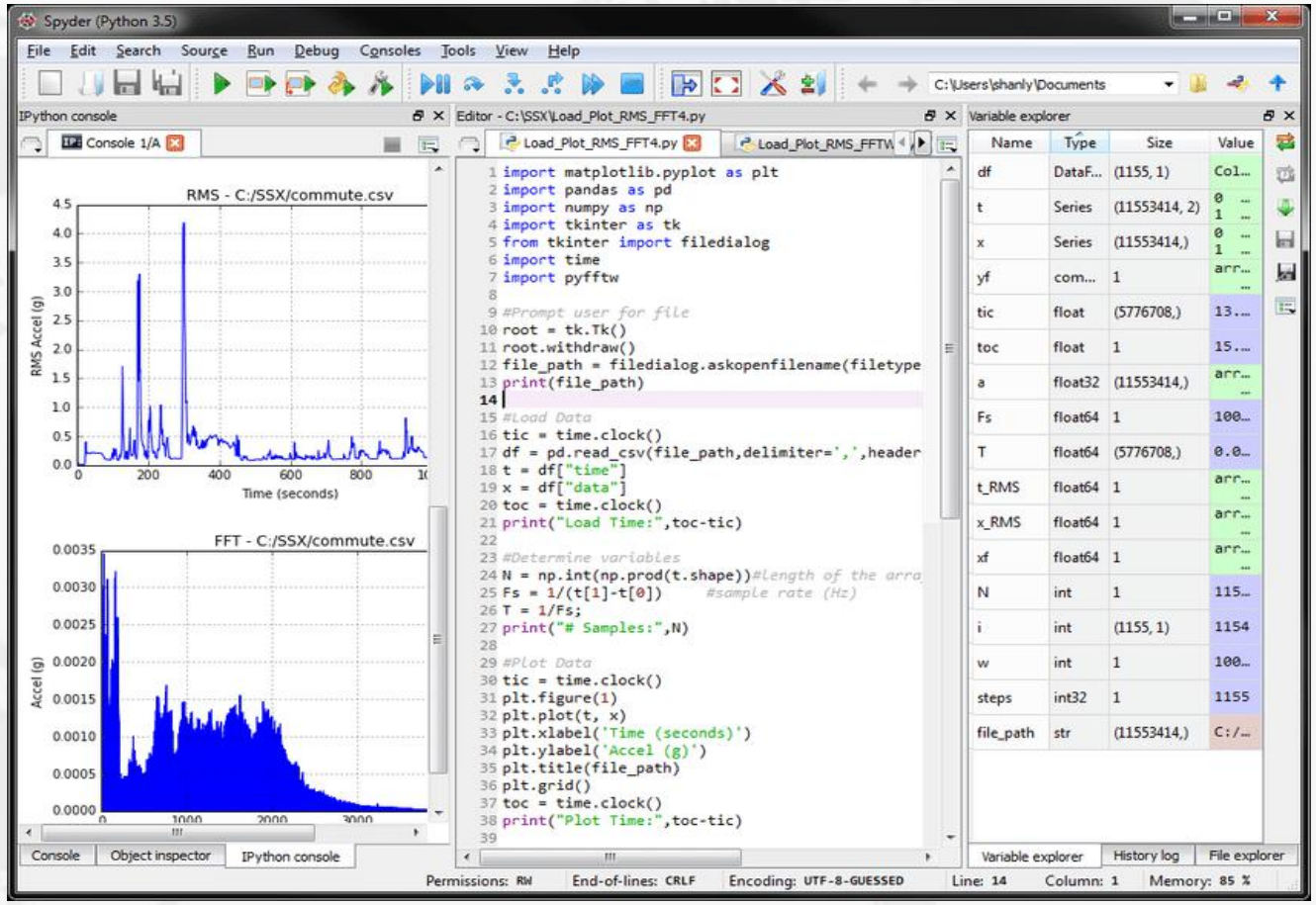
安装python

https://www.anaconda.com/products/individual



Jupyter VS Spyder





Data type

Data Structure	Ordered	Mutable	Constructor	Example
List	Yes	Yes	`[ ]` or `list()`	`[5.7, 4, 'yes', 5.7]`
Tuple	Yes	No	`( )` or `tuple()`	`(5.7, 4, 'yes', 5.7)`
Set	No	Yes	`{}`* or `set()`	`{5.7, 4, 'yes'}`
Dictionary	No	Yes**	`{ }` or `dict()`	`{'Jun': 75, 'Jul': 89}`

	列表	元组	集合	字典
英文	list	tuple	set	dict
可否读写	读写	只读	读写	读写
可否重复	是	是	否	是
存储方式	值	值	键(不能重复)	键值对(键不能重复)
是否有序	有序	有序	无序	无序，自动正序
初始化	`[1,'a']`	`('a', 1)`	`set([1,2])` 或 `{1,2}`	`{'a':1,'b':2}`
添加	`append`	只读	`add`	`d['key'] = 'value'`
读元素	`l[2:]`	`t[0]`	无	`d['a']`

Pandas: **Pan**el **Da**ta **S**ystem

Rich data structures and functions to work with structured data fast, easy and expressive

Build on top of NumPy

Ideal tool for:
munging/cleaning/analyzing/modeling data

Quiz#1

Python >>>

```python
>>> colors = ["red", "green", "burnt sienna", "blue"]
>>> colors[2]
```

What is the output of the `colors[2]` expression?

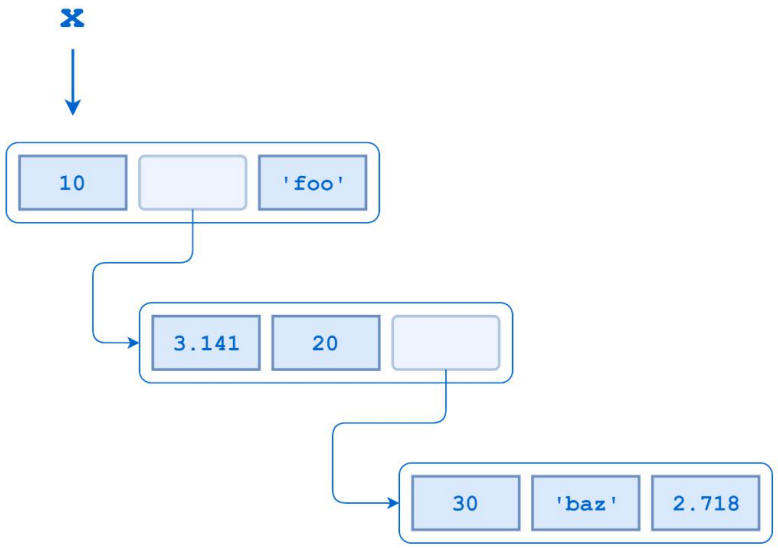
- It causes a run-time error.
- `'green'`
- `'burnt sienna'`
- `'red'`
- `'blue'`

Quiz #2

Consider the following nested list definition:

```python
x = [10, [3.141, 20, [30, 'baz', 2.718]], 'foo']
```

A schematic for this list is shown below:



What is the expression that returns the `'z'` in `'baz'`?

Quiz #2 - Answer

```python
x = [10, [3.141, 20, [30, 'baz', 2.718]], 'foo']
```

Expression	Selects
`x[1]` `x[-2]`	The second element of `x`: `[3.141, 20, [30, 'baz', 2.718]]`
`x[1][2]` `x[1][-1]`	The third element of that sublist: `[30, 'baz', 2.718]`
`x[1][2][1]` `x[1][2][-2]`	The second element of that sublist: `'baz'`
`x[1][2][1][2]` `x[1][2][1][-1]`	The third character of `'baz'`: `'z'`

Regular Expression

Identifiers	Modifiers	White space characters	Escape required
\d= any number (a digit)	\d represents a digit.Ex: \d{1,5} it will declare digit between 1,5 like 424,444,545 etc.	\n = new line	. + * ? [] $ ^ () {} | \
\D= anything but a number (a non-digit)	+ = matches 1 or more	\s= space	
\s = space (tab,space,newline etc.)	? = matches 0 or 1	\t =tab	
\S= anything but a space	* = 0 or more	\e = escape	
\w = letters (Match alphanumeric character, including "_")	$ match end of a string	\r = carriage return	
\W =anything but letters (Matches a non-alphanumeric character excluding "_")	^ match start of a string	\f= form feed	
. = anything but letters (periods)	| matches either or x/y	----------	
\b = any character except for new line	[] = range or "variance"	----------	
\.	{x} = this amount of preceding code	----------	

https://www.guru99.com/python-regular-expressions-complete-tutorial.html

Data Reshaping for Single Table

Group by
Aggregation
Pivot
Pivot Table
Stack
Unstack
Melt

Pivot

df

	foo	bar	baz	zoo
0	one	A	1	x
1	one	B	2	y
2	one	C	3	z
3	two	A	4	q
4	two	B	5	w
5	two	C	6	t

```
df.pivot(index='foo',
         columns='bar',
         values='baz')
```

bar	A	B	C
foo			
one	1	2	3
two	4	5	6

Melt

df3

	first	last	height	weight
0	John	Doe	5.5	130
1	Mary	Bo	6.0	150

```
df3.melt(id_vars=['first', 'last'])
```

	first	last	variable	value
0	John	Doe	height	5.5
1	Mary	Bo	height	6.0
2	John	Doe	weight	130
3	Mary	Bo	weight	150

Stack

df2

		A	B
first	second		
bar	one	1	2
	two	3	4
baz	one	5	6
	two	7	8

MultiIndex

```
stacked = df2.stack()
```

first	second		
bar	one	A	1
		B	2
	two	A	3
		B	4
baz	one	A	5
		B	6
	two	A	7
		B	8

MultiIndex

Unstack

stacked

first	second		
bar	one	A	1
		B	2
	two	A	3
		B	4
baz	one	A	5
		B	6
	two	A	7
		B	8

MultiIndex

```
stacked.unstack()
```

		A	B
first	second		
bar	one	1	2
	two	3	4
baz	one	5	6
	two	7	8

MultiIndex

Quiz # 3

```
exp = lambda x: x ** 3
print(exp(2))
```

A. 6

B. 222

C. 8

D. None of the above

Quiz#4

```
def func(message, num = 1):
    print(message * num)

func('Welcome')
func('Viewers', 3)
```

A. Welcome
Viewers

B. Welcome
ViewersViewersViewers

C. Welcome
Viewers,Viewers,Viewers

D. Welcome

Quiz #5

Q.10 Which of the following can be used to make a Dataframe?

- [ ] Series
- [ ] DataFrame
- [ ] Structured ndarray
- [ ] All of the above

Data Operation for Multiple Tables

Append
Concat
Join/merge

df1

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

df2

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

df3

	A	B	C	D
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

Result

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

df1

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

df4

	B	D	F
2	B2	D2	F2
3	B3	D3	F3
6	B6	D6	F6
7	B7	D7	F7

Result

	A	B	C	D	B	D	F
2	A2	B2	C2	D2	B2	D2	F2
3	A3	B3	C3	D3	B3	D3	F3

df1

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

df4

	B	D	F
2	B2	D2	F2
3	B3	D3	F3
6	B6	D6	F6
7	B7	D7	F7

Result

	A	B	C	D	B	D	F
0	A0	B0	C0	D0	NaN	NaN	NaN
1	A1	B1	C1	D1	NaN	NaN	NaN
2	A2	B2	C2	D2	B2	D2	F2
3	A3	B3	C3	D3	B3	D3	F3

df1

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

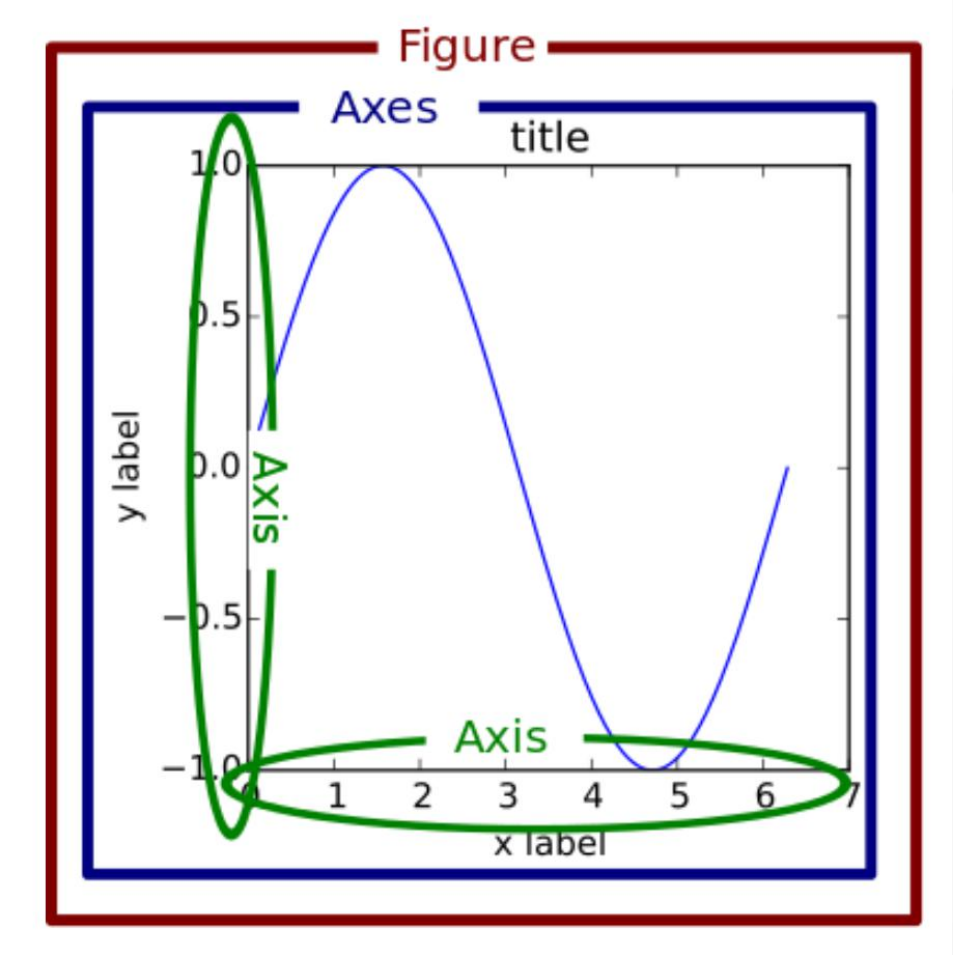
s1

	X
0	X0
1	X1
2	X2
3	X3

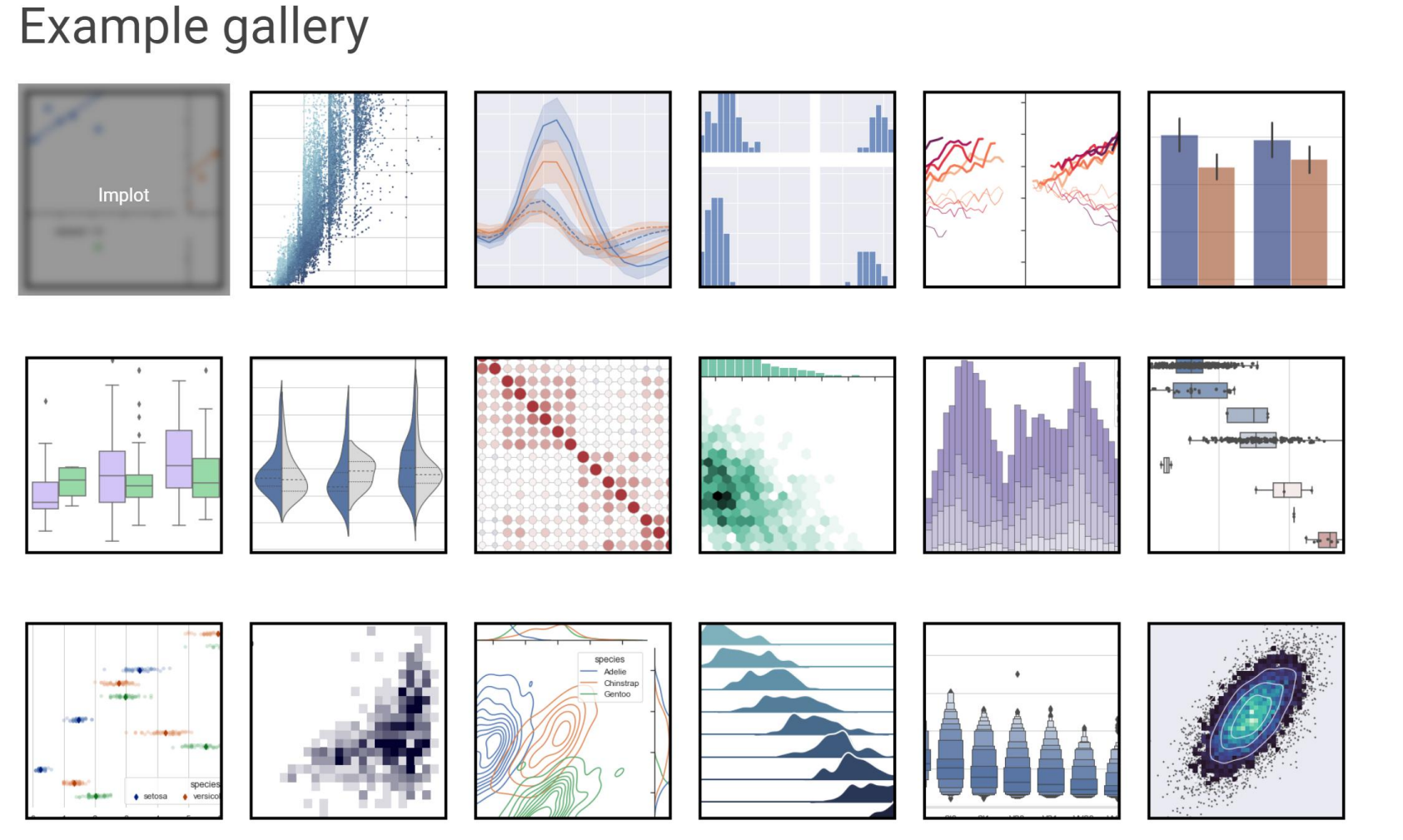
Result

	A	B	C	D	X
0	A0	B0	C0	D0	X0
1	A1	B1	C1	D1	X1
2	A2	B2	C2	D2	X2
3	A3	B3	C3	D3	X3

Matplotlib Plots



Seaborn Plot



link：https://seaborn.pydata.org/examples/index.html