DATA 690
Statistical Analysis and Data Visualization With Python

Dr. Chaojie (Jay) Wang

About the Instructor

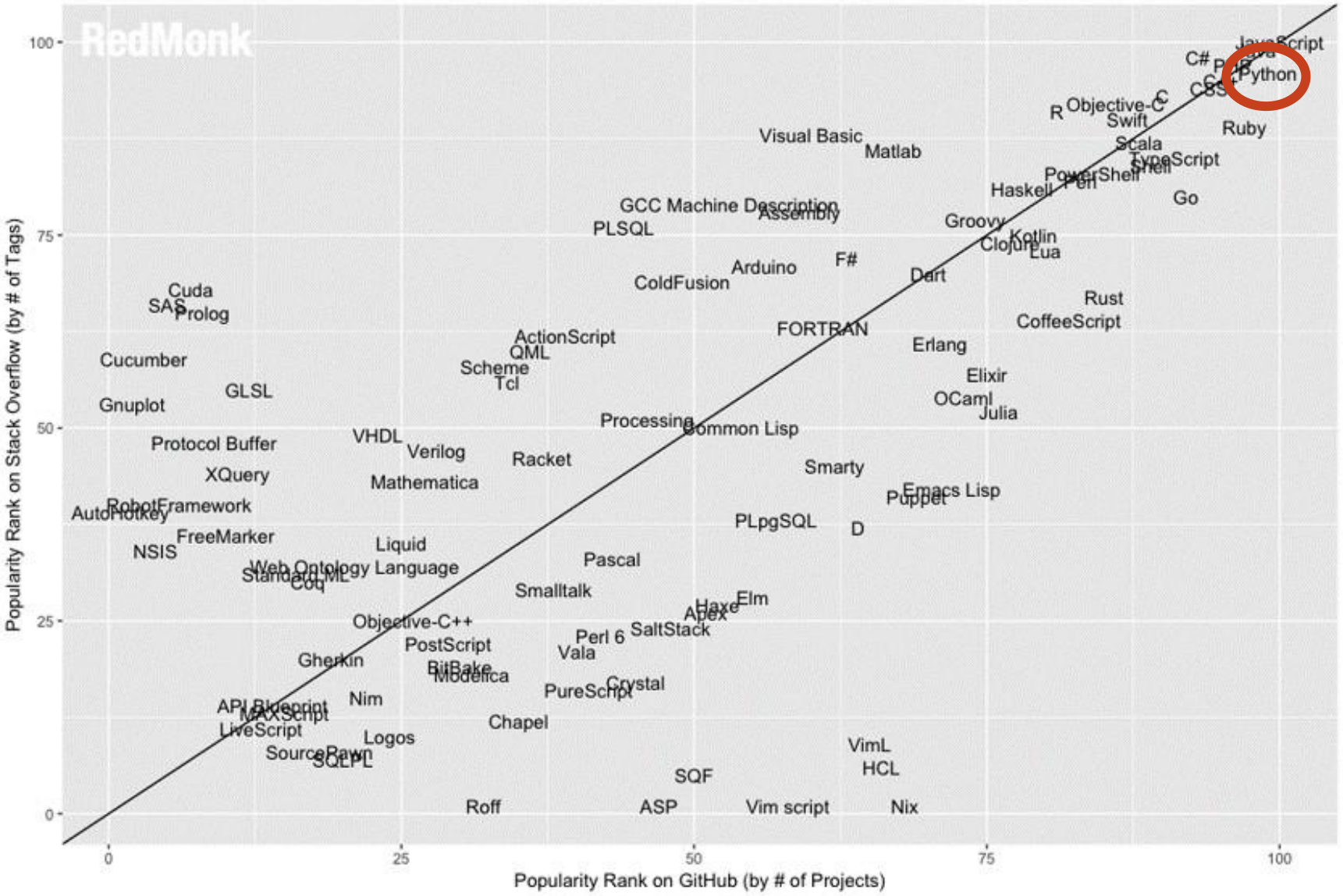
- **D.Sc. in Info Sys & Communications**
- **MBA in Finance, MS in Stats, MA in Econ**
- **Principle Systems Engineer, MITRE**
- **Adjunct Faculty, Data Science, UMBC**
- **Corporate Faculty, Health Informatics, Harrisburg Univ.**
- **Editor-in-Chief, International Journal of Patient-Centered Healthcare (IJPCH)**

wcj365@gmail.com
http://wcj365.gitee.io

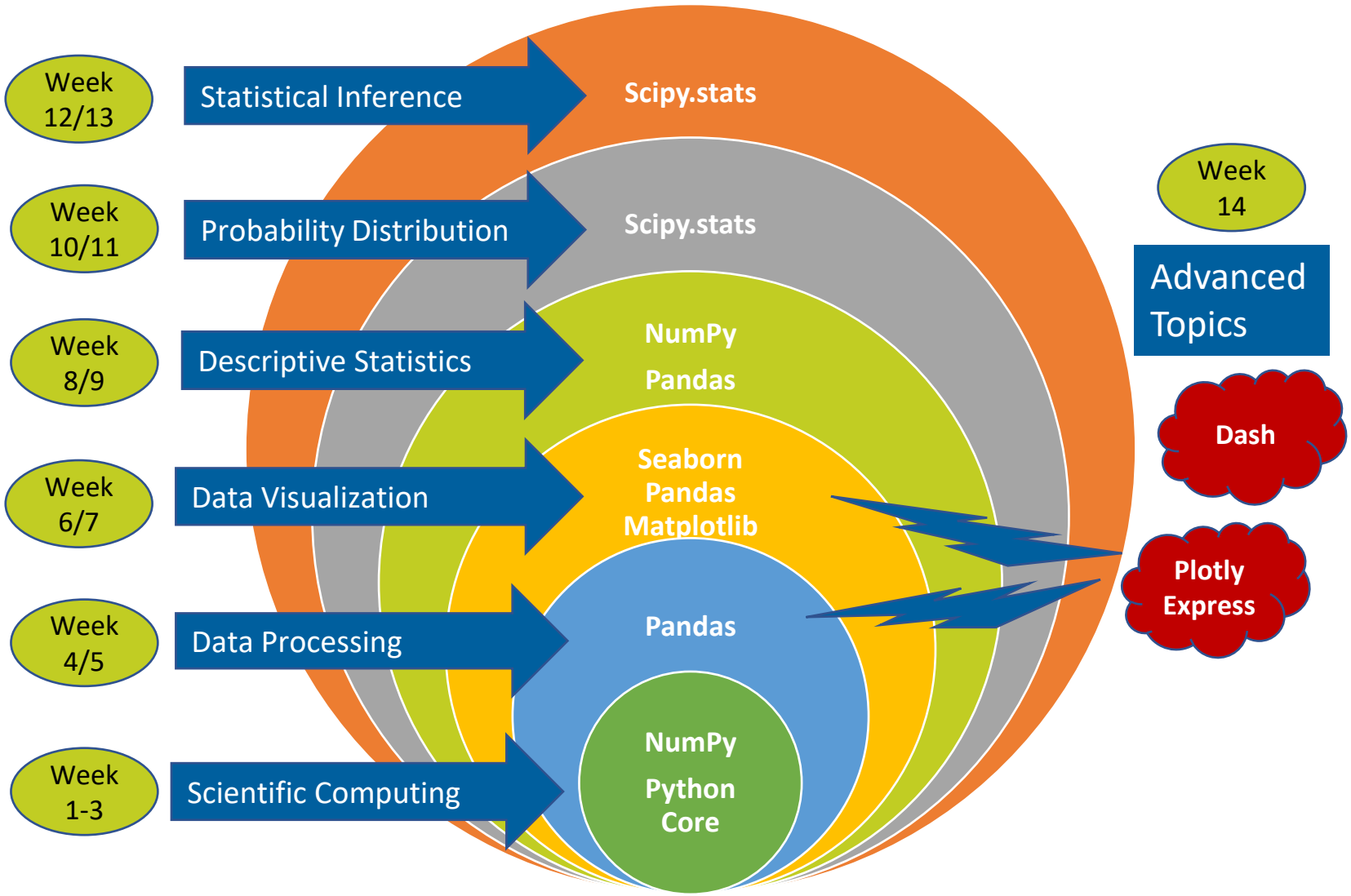
Why Data Science?

Why Python

https://redmonk.com/sogrady/2019/03/20/language-rankings-1-19/



Course Plan


Week 12/13
Statistical Inference
Week 10/11
Probability Distribution
Week 8/9
Descriptive Statistics
Week 6/7
Data Visualization
Week 4/5
Data Processing
Week 1-3
Scientific Computing
Scipy.stats
Scipy.stats
NumPy
Pandas
Seaborn
Pandas
Matplotlib
Pandas
NumPy
Python
Core
Week 14
Advanced Topics
Dash
Plotly Express

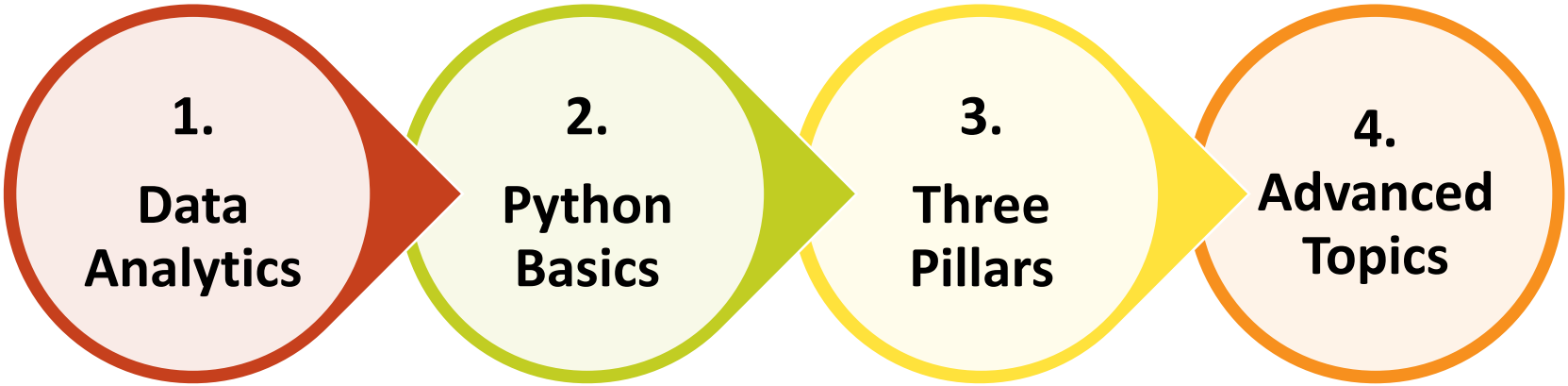
Practice Makes Perfect

knowledge cannot be taught;

It must be learned, practiced,

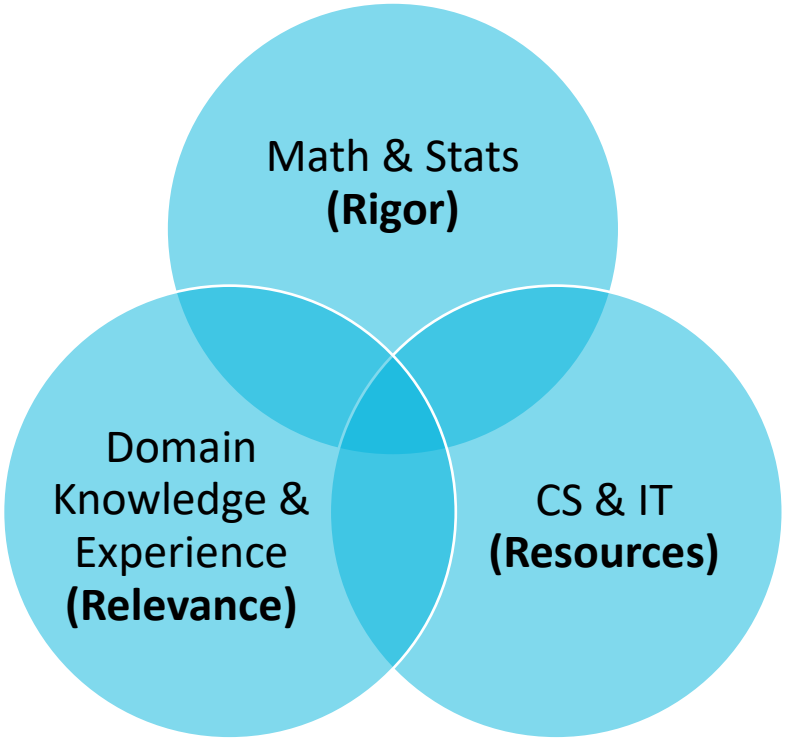
applied, and shared.

Agenda


1.
Data Analytics
2.
Python Basics
3.
Three Pillars
4.
Advanced Topics

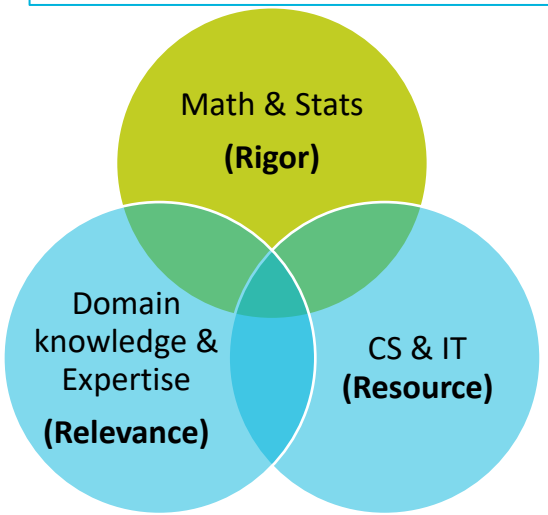
1. Data Analytics

Three R's of Data Analytics
- Relevance, Rigor, and Resources



Three R's of Data Analytics (Cont'd)

Challenge: Data Scarcity
Technique: Sampling & Inference



Challenge: Decision Support
Technique: Data Analytics/BI

Challenge: Data Deluge
Technique: Machine Learning

Goal: Help understand data, reduce info overload, and formulate hypothesis

Exploratory Data Analysis: Foundation for Data Analytics

Identify patterns, trends, and, outliers, anomalies
Compute descriptive statistics, plot visual charts

Reflections

- **Statistical Inference**
  - I know exactly what the formula does even though I don't quite understand the math behind it.
- **Machine Learning**
  - I don't know what the machine does (especially with neural network) and whether the results make sense or not. If it predicts, it works.
- **Exploratory Data Analysis (EDA)**
  - I look at data in many ways (raw data, summary statistics, tabular, graphical). I know what they are and what they look like. I can explore; I can make educated guesses.
- **EDA Support both Statistical Inference and Machine Learning**

Statistical Inference vs Machine Learning

Statistical Inference

- Mathematical Proof

- Relationship/association
- Significance of the effect
- Size of the effect

Machine Learning

- Computational Power

- Prediction
- If it predicts, it works
- Don’t ask me why

https://blog.thedataincubator.com/2017/11/scikit-learn-vs-statsmodels/
https://healthcare.ai/machine-learning-versus-statistics-use/
https://towardsdatascience.com/the-actual-difference-between-statistics-and-machine-learning-64b49f07ea3

Foundation of Inferential Statistics

- **The Law of Large Numbers (LLN)**
  - Sample means converge to the population mean
- **The Central Limit Theorem (CLT)**
  - Sample means have a near normal distribution regardless of population distribution

- **Provide Mathematical Rigor**
  - Applied statistics
  - Probability theory
- **Deal with Data Scarcity**
  - Sampling technique
  - Experimental design
  - Example: Survey

Development of EDA

- **John W. Tukey wrote the book Exploratory Data Analysis in 1977. Tukey held that too much emphasis in statistics was placed on statistical hypothesis testing (confirmatory data analysis); more emphasis needed to be placed on using data to suggest hypotheses to test. In particular, he held that confusing the two types of analyses and employing them on the same set of data can lead to systematic bias owing to the issues inherent in testing hypotheses suggested by the data.**

Objectives of EDA

- **Suggest hypotheses about the causes of observed phenomena**
- **Assess assumptions on which statistical inference will be based**
- **Support the selection of appropriate statistical tools and techniques**
- **Provide a basis for further data collection through surveys or experiments**

http://cll.stanford.edu/~willb/course/behrens97pm.pdf

Anscombe’s Quartet

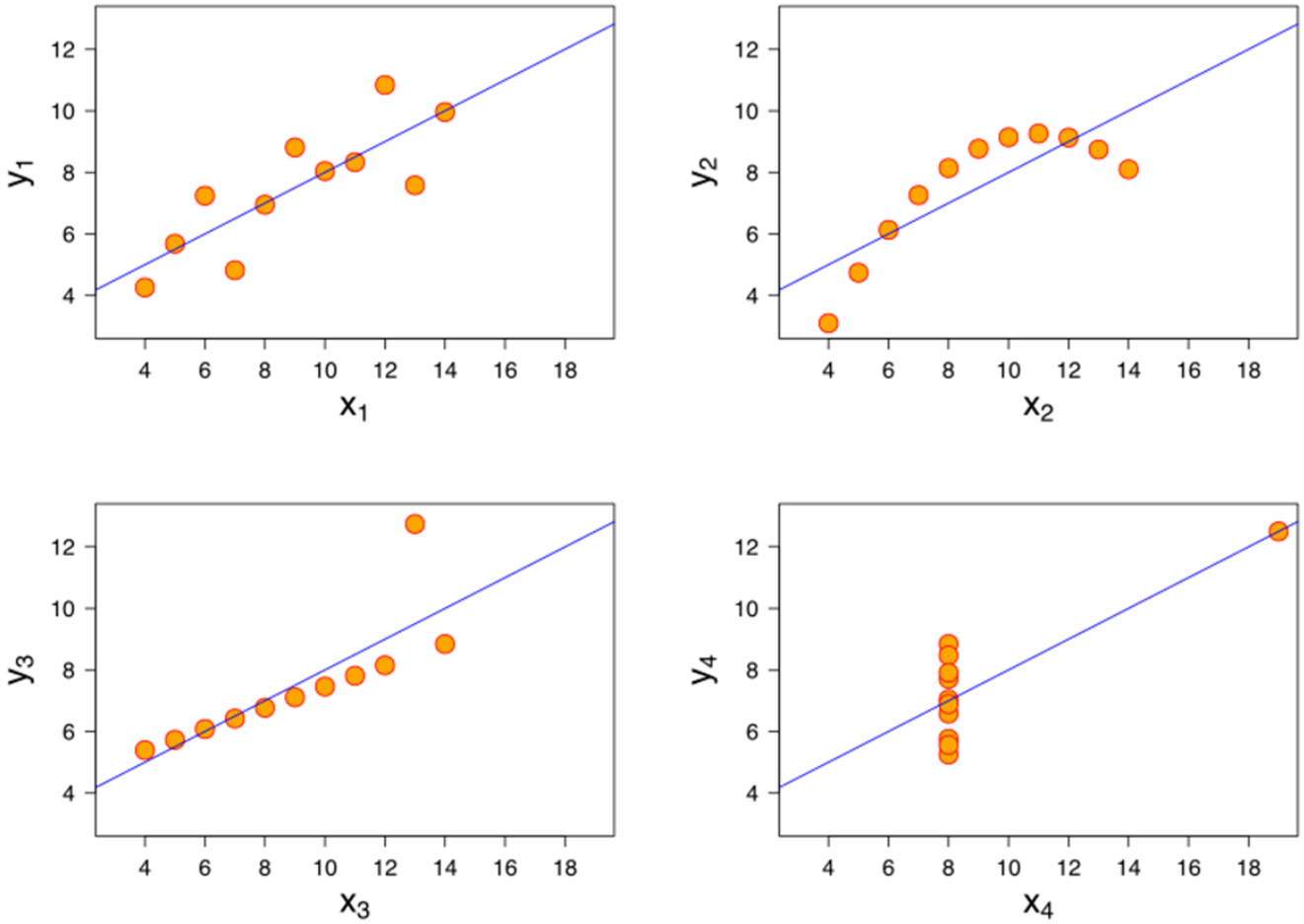
“Anscombe's quartet comprises four datasets that have nearly identical simple descriptive statistics, yet appear very different when graphed. Each dataset consists of eleven (x, y) points. They were constructed in 1973 by the statistician Francis Anscombe to demonstrate both the importance of graphing data before analyzing it and the effect of outliers on statistical properties. He described the article as being intended to counter the impression among statisticians that "numerical calculations are exact, but graphs are rough."

https://en.wikipedia.org/wiki/Anscombe%27s_quartet

Four Different Datasets Share the Same Properties

	I		II		III		IV	
	x	y	x	y	x	y	x	y
	10	8,04	10	9,14	10	7,46	8	6,58
	8	6,95	8	8,14	8	6,77	8	5,76
	13	7,58	13	8,74	13	12,74	8	7,71
	9	8,81	9	8,77	9	7,11	8	8,84
	11	8,33	11	9,26	11	7,81	8	8,47
	14	9,96	14	8,1	14	8,84	8	7,04
	6	7,24	6	6,13	6	6,08	8	5,25
	4	4,26	4	3,1	4	5,39	19	12,5
	12	10,84	12	9,13	12	8,15	8	5,56
	7	4,82	7	7,26	7	6,42	8	7,91
	5	5,68	5	4,74	5	5,73	8	6,89
SUM	99,00	82,51	99,00	82,51	99,00	82,50	99,00	82,51
AVG	9,00	7,50	9,00	7,50	9,00	7,50	9,00	7,50
STDEV	3,32	2,03	3,32	2,03	3,32	2,03	3,32	2,03

The Scatter Plots Tell the Different Stories

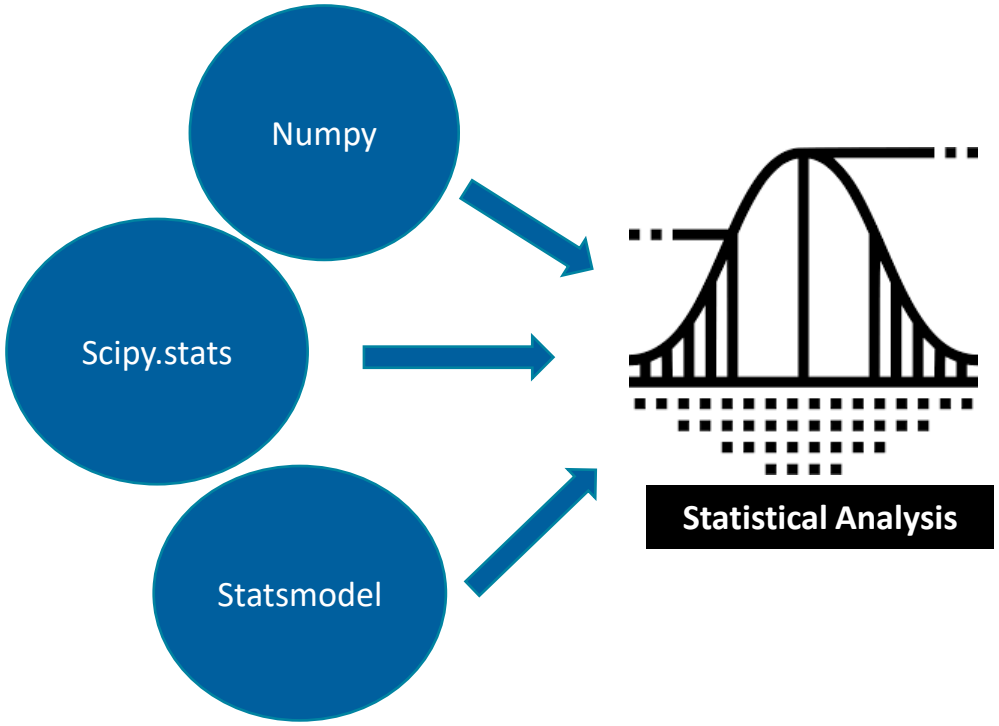


Tufte's Six Fundamental Principles of Analytical Thinking

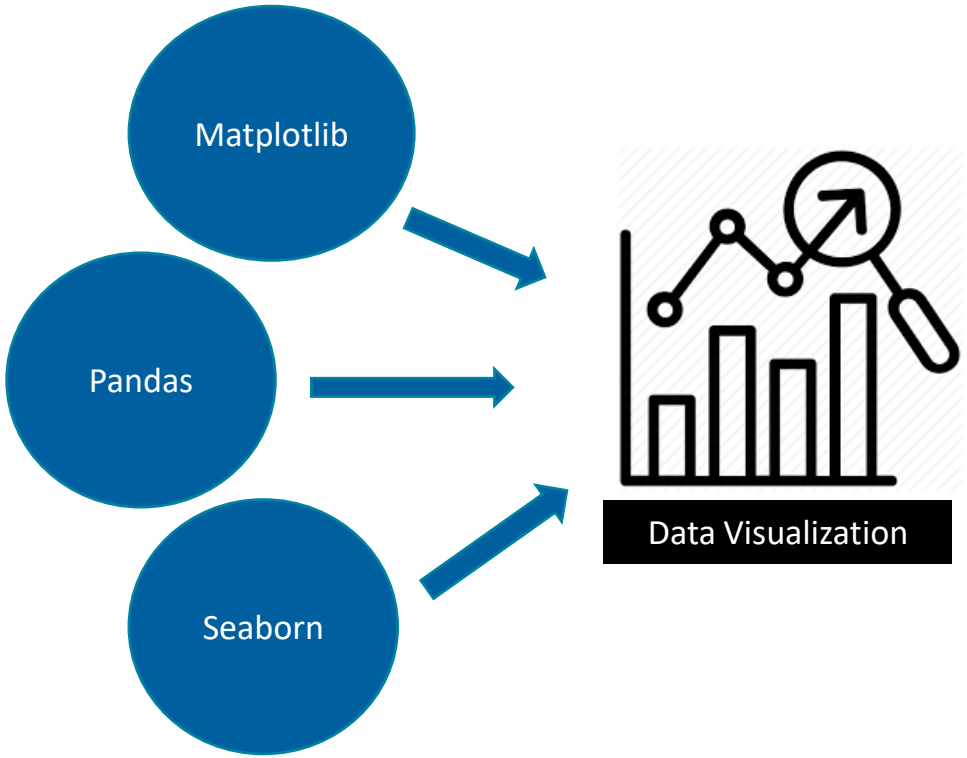
- **Comparisons**
- **Causality, Mechanism, Structure, Explanation**
- **Multivariate Analysis**
- **Integration of Evidence**
- **Documentation**
- **Contents Counts Most of All**

https://medium.com/open-machine-learning-course

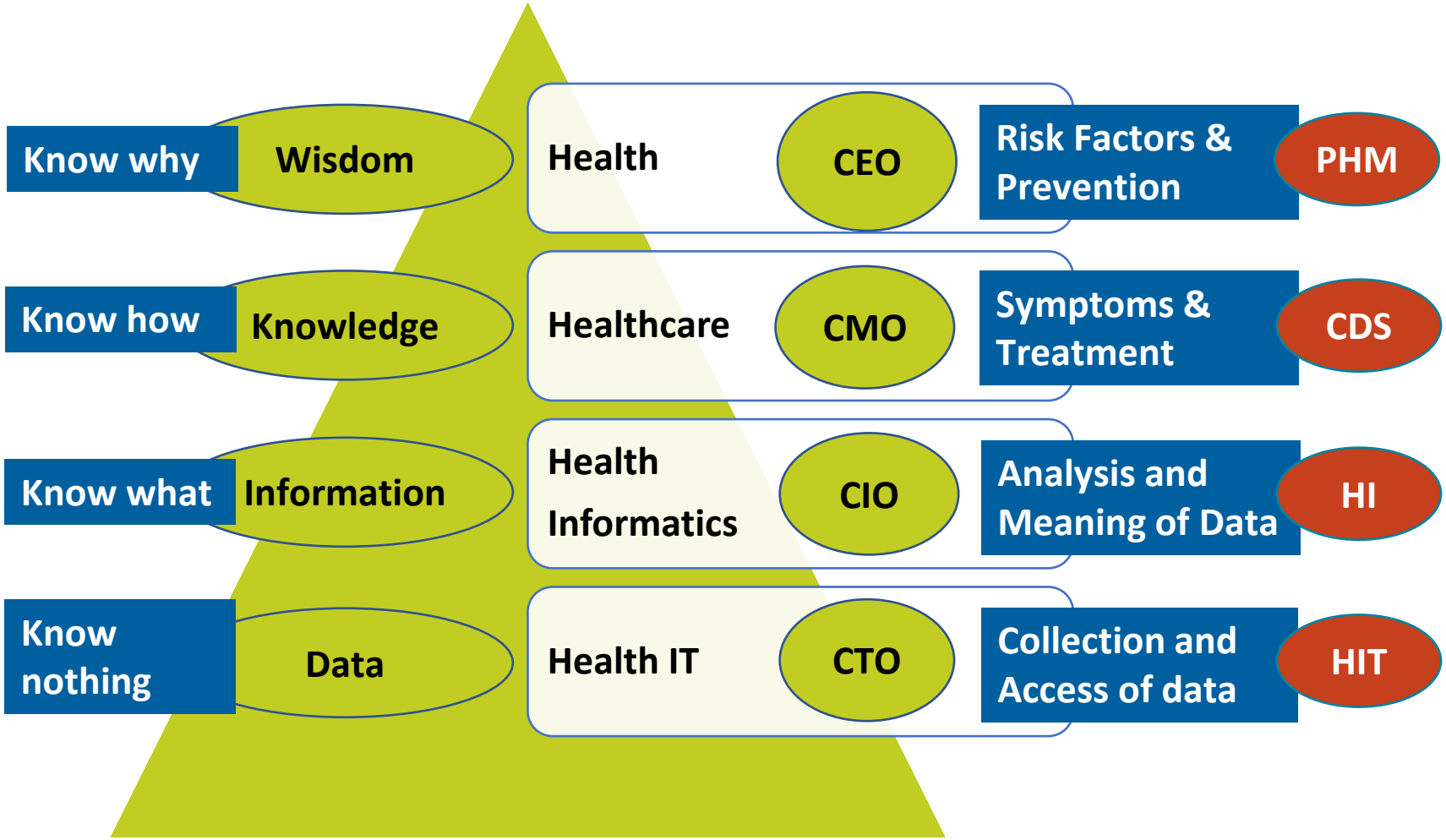
Statistical Analysis Using Python



Data Visualization Using Python



Wisdom Hierarchy & Healthcare Informatics



PHM – Population Health Management
CDS – Clinical Decision Support
HI – Health Informatics
HIT – Health Information

2. Python Basics

What is Python?

- **"Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together."**

- python.org

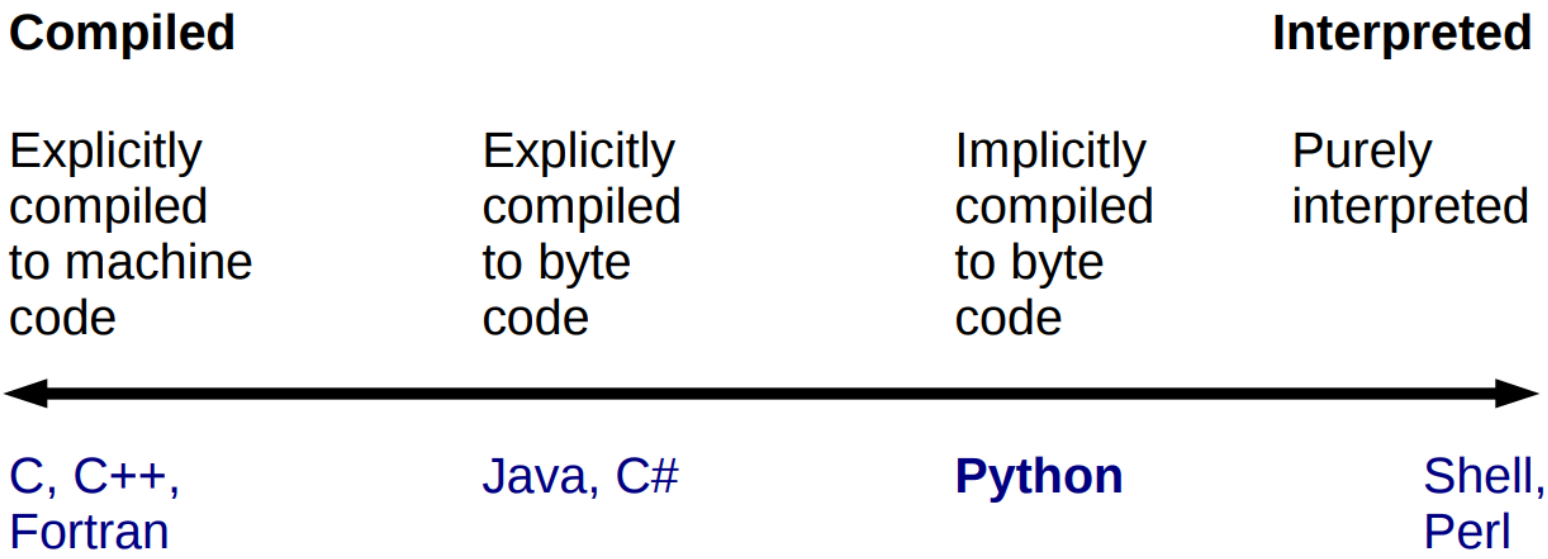
What is Python

- **General-purpose high-level programming language**
  - Interpreted
  - Cross-platform
- **Elegant and simple**
  - No brackets (use indentation instead)
  - Dynamic typing (no need to define variable)
- **Object-oriented**
  - But support procedural programming style
- **Flexible development**
  - Interactive mode
  - Script mode

Use Interactive Python Shell to Practice from python.org

Python

What sort of language is Python?



What is Jupyter Notebooks?

- **Jupyter Notebooks offers "an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text."**

Development Environments

Development Environment

- **Desktop - Anaconda**
  - IDE - Spyder
  - Notebooks – JupyterLab
- **Web-based**
  - Interactive Shell (python.org)
  - IDE (repl.it)

Jupyter Notebooks Hosting

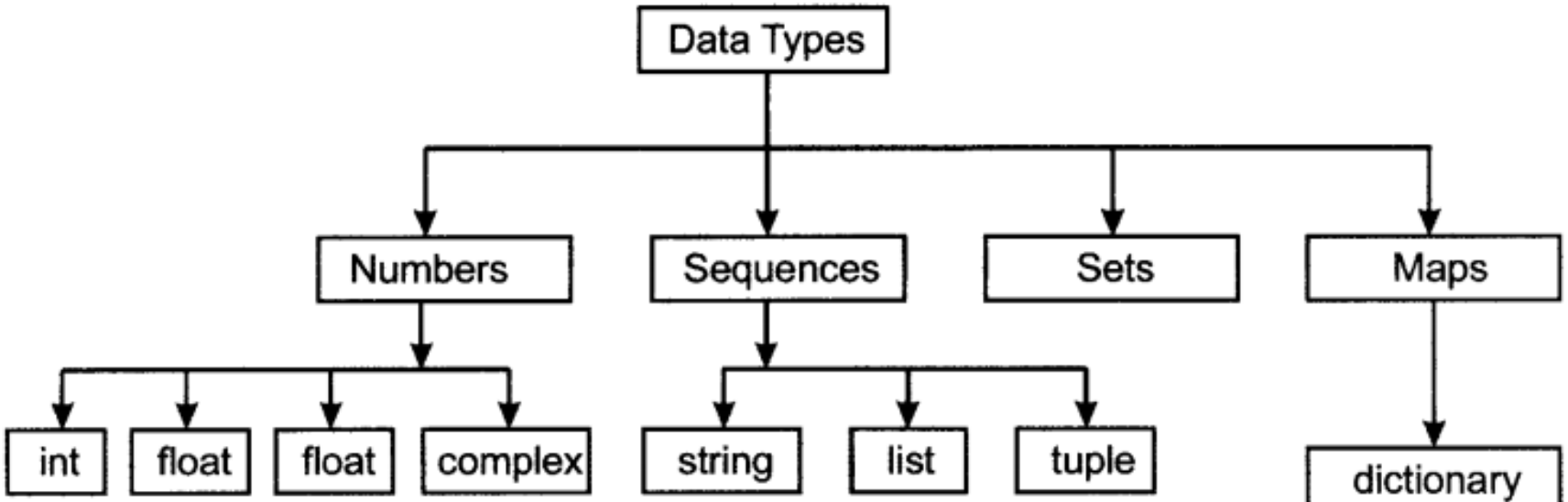
- **Google Colab**
  - https://colab.research.google.com/
- **Kaggle**
  - http://www.Kaggle.com
- **DeepNote.com**
  - https://www.deepnote.com/

Jupyter Notebooks integrate code, output, and documentation in a single document and are great for exploratory data analysis, data viz and knowledge sharing.
Tip: use ipywidgets for even more interactivity.

Python Resources

- **Python Tutorial**
  - https://docs.python.org/3/tutorial/
- **Python 3 Cheatsheet**
  - https://perso.limsi.fr/pointal/_media/python:cours:mementopython3-english.pdf
- **Python Practices**
  - https://www.practicepython.org/
- **Think Python 2nd edition**
  - https://greenteapress.com/thinkpython2/thinkpython2.pdf
- **Python 3 Patterns, Recipes, and Idioms**
  - https://python-3-patterns-idioms-test.readthedocs.io
- **Python Programing: An Intro to CS by John Zelle**
  - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.6062&rep=rep1&type=pdf

Python Data Types



Python Data types

To find out the type of a variable, use **type()** function

Basic Data Types

Data Type	Type()	Example
Integer	int	0, 9, -5, 145
Float	float	0.0, 9.5, -5.9, 1.7e-10 (scientific notation)
Complex		4+3j, 8.0+4.6j
Boolean	bool	True, False
Byte	bytes	
Date	datetime	import datetime today = datetime.datetime.now() x = datetime.datetime(2019, 6, 2)
None	NoneType	x = None, A null value.

Compound Data Types

Type	Nature	Mutable	Example	Notes
String	Ordered sequence	Immutable	Greeting = “Hello, world!”	Slicing: [start:stop:step]
range	Ordered sequence	Immutable	x = range(2, 10, 2)	range(start, stop, step) compare with np.arange()
Tuple	Ordered sequence	Immutable	x = 1, “Hi”, 10.5	
List	Ordered sequence	Mutable	X = [2, 4, 11, “New”]	Slicing: [start:stop:step] Compare with nparray
Set	Unordered collection	Mutable	S = {“Apple”, “Strawberry”]	Can’t have duplicate members
Dict	Unordered Mapping	Mutable	x = {“name” : ”Jay”, “age” : 50}	key:value pair Keys must be unique

Membership function: **in**
Example: **x in (“A”, “B”, “C”)**

Control Flows (Conditions, loops, and exceptions)

Condition-based loop

```
while condition:
    ...
else:
    ...
```

Multiway conditional

```
if condition:
    ...
elif condition:
    ...
else:
    ...
```

Exception handling

```
try:
    ...
except exception:
    ...
else:
    ...
finally:
    ...
```

No-op

```
pass
```

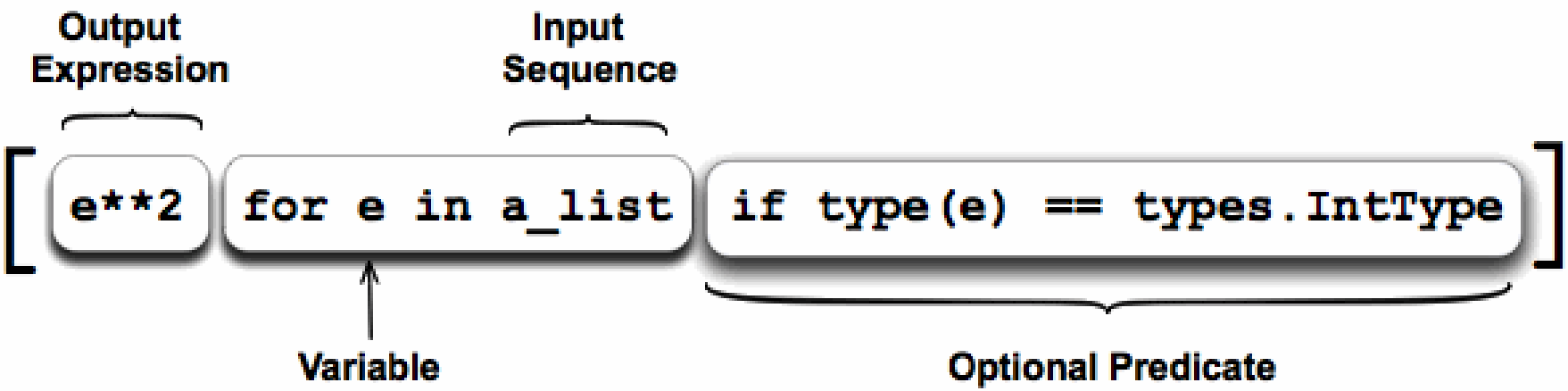
```
break
Continue
```

Value-based loop

```
for variable in sequence:
    ...
else:
    ...
```

List Comprehensions

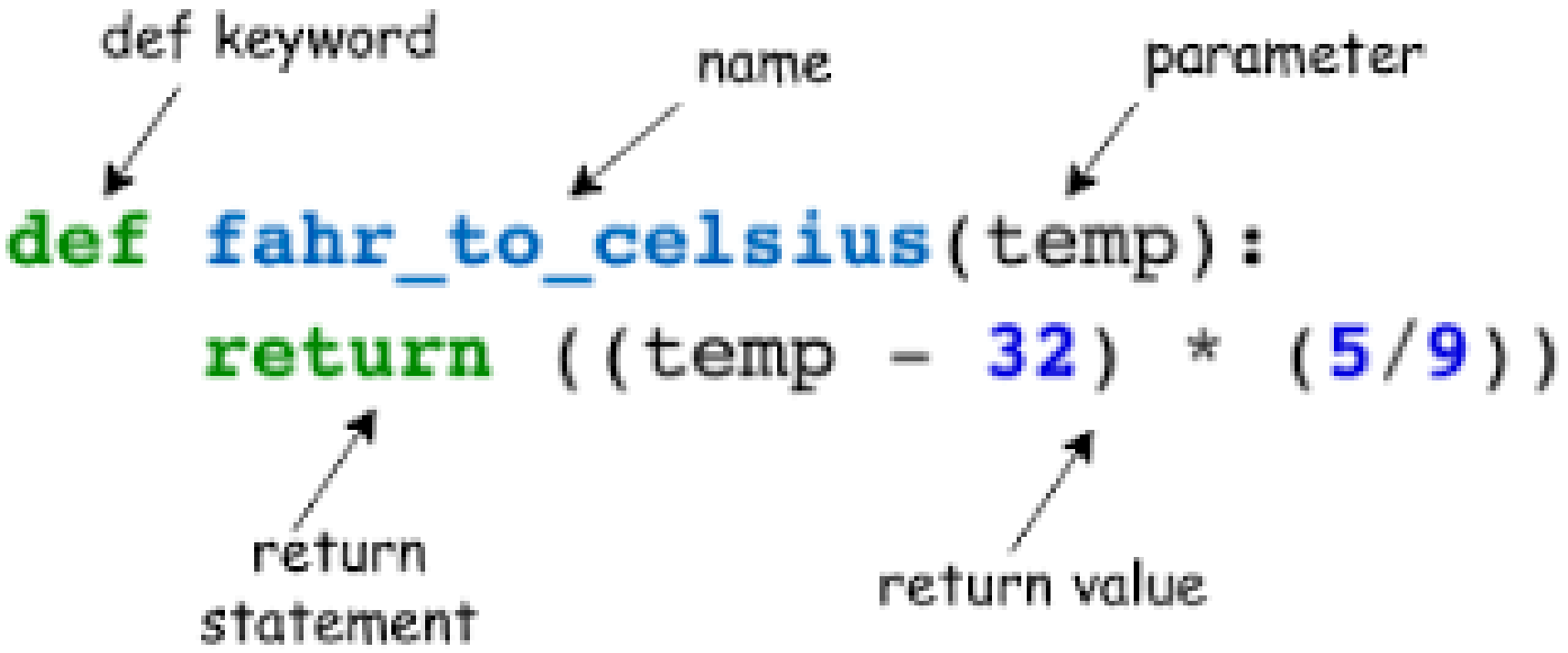
- Create a new list from an existing list without using a for loop
- Only one line of code
- Simplicity - > Beauty
- Parsimony -> Efficiency



Example

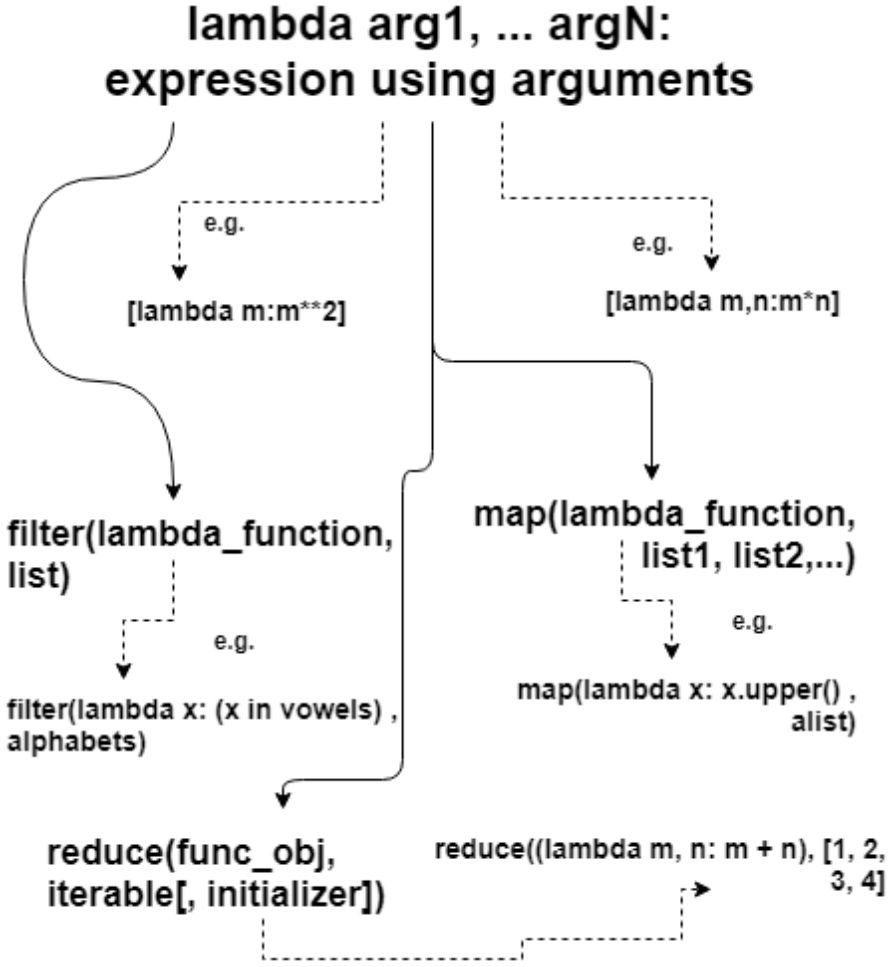
- Given a list x = [1, 4, -5, 10, 2, -6]
- Create a new list y that contains items in x that are less than 3
- Y = [a for a in x if a < 3]

Function


def keyword
name
parameter
def fahr_to_celsius(temp):
return ((temp - 32) * (5/9))
return statement
return value

Lambda Function (Anonymous Function)

- **Inline, anonymous function**
- **Multiple inputs/arguments**
- **Single output/expression**
- **Simplicity -> Beauty**
- **Parsimony -> Efficiency**

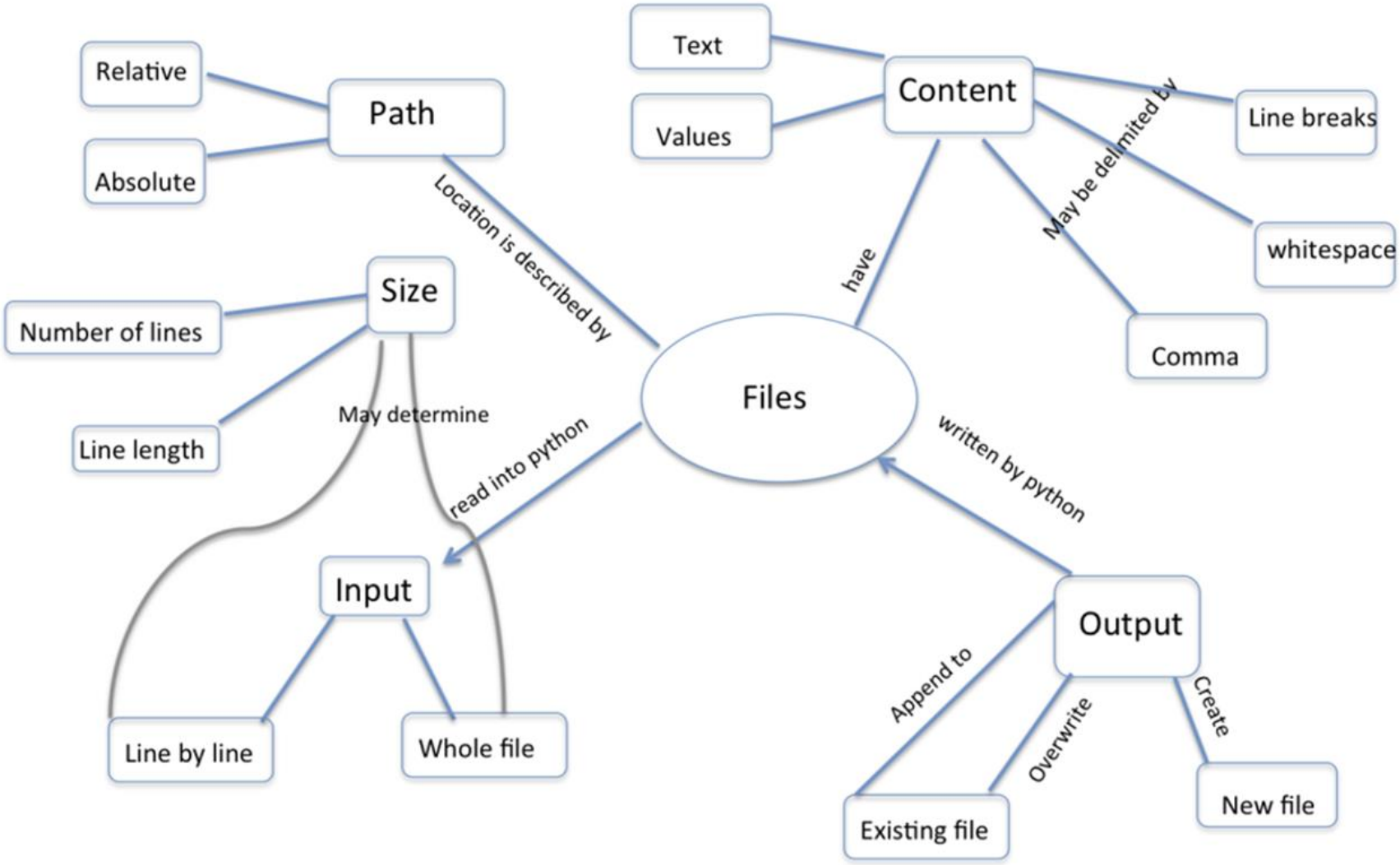


Input and Output

- **User Input**
  - X = input("Please enter a number:")
- **Output to screen**
  - print()

- **File I/O**
  - f = open("abc.txt)
  - f.read()
  - f.readline()
  - f.close()

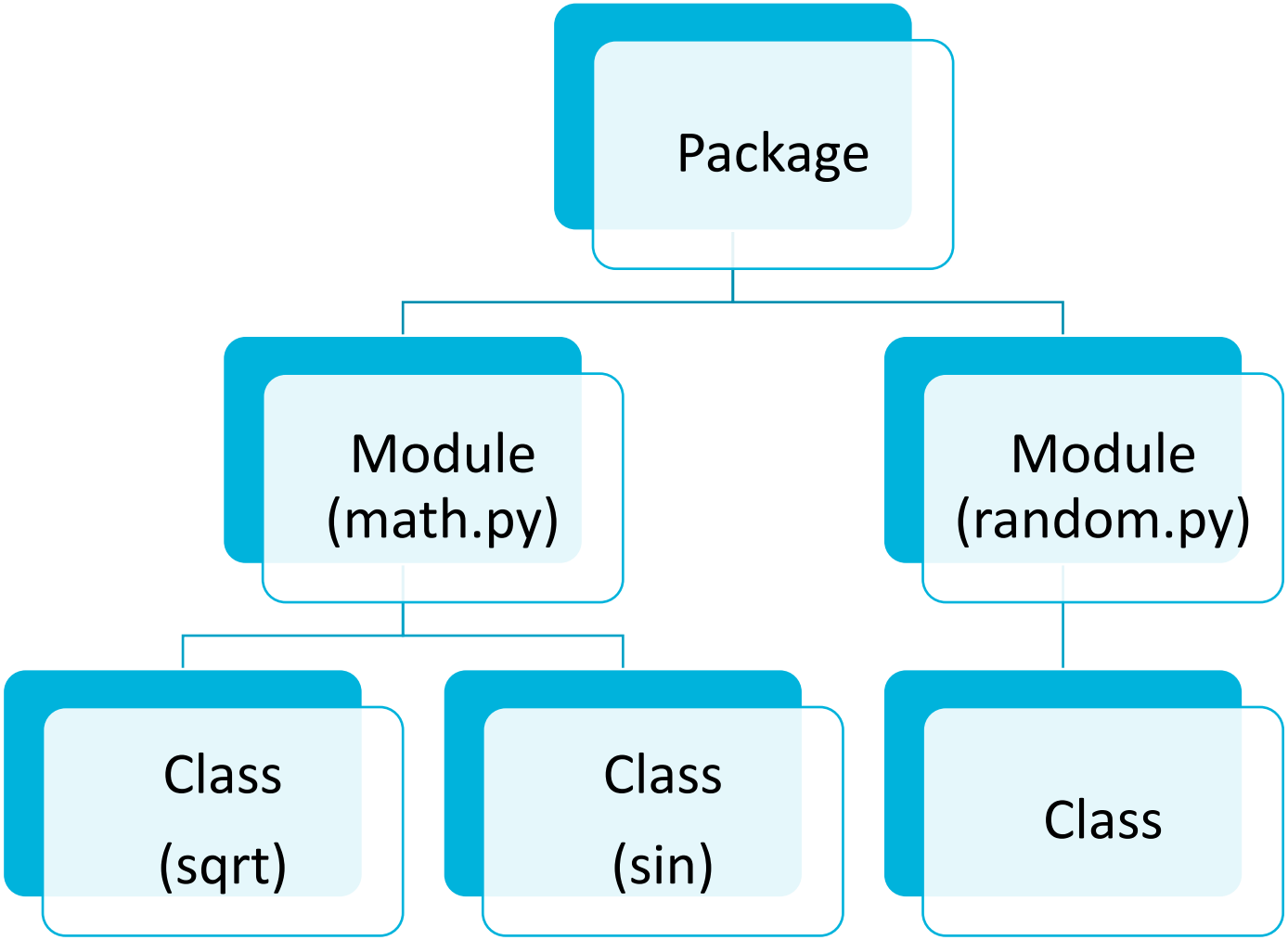
IO


Relative
Absolute
Path
Location is described by
Text
Values
Content
have
May be delimited by
Line breaks
whitespace
Comma
Size
Number of lines
Line length
May determine
Files
read into python
written by python
Input
Line by line
Whole file
Output
Append to
Overwrite
Create
Existing file
New file

Web Scraping

- **Requests**
- **BeautifulSoup**

Python Packages, Modules, Classes

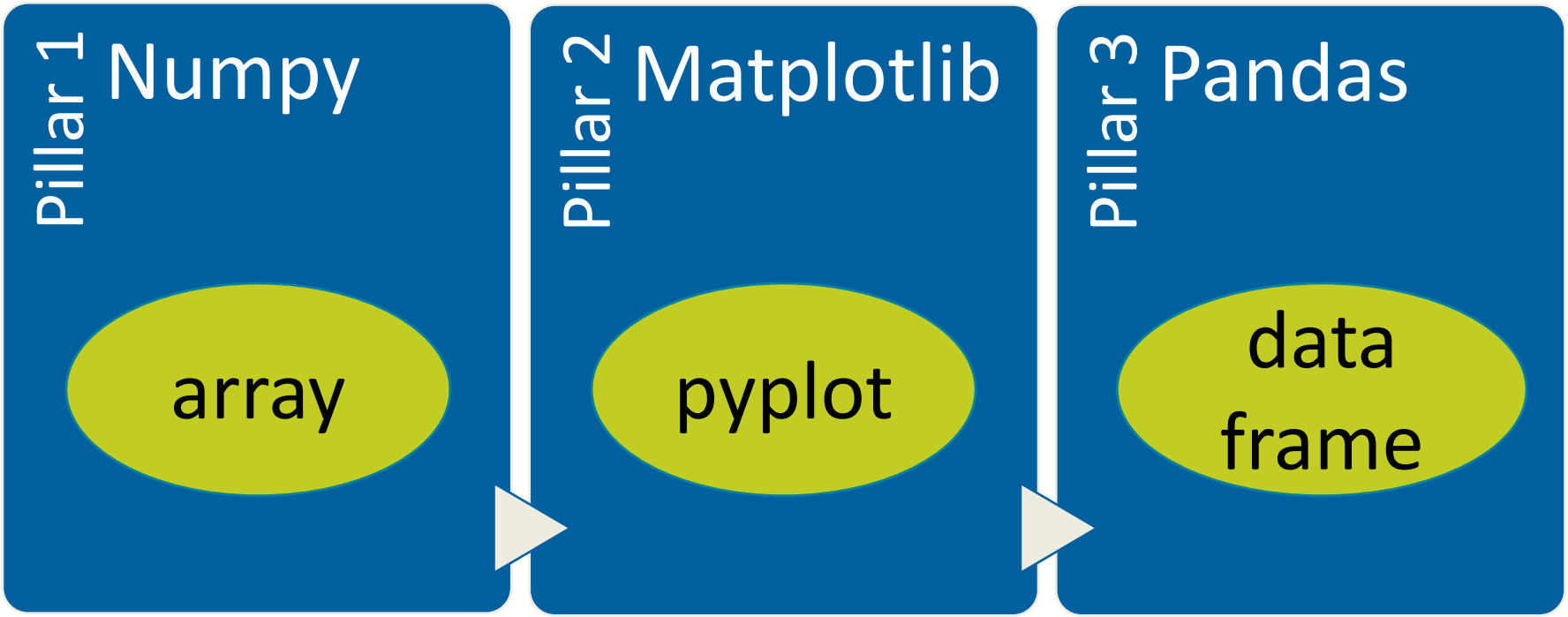


Other Useful Modules

- **pathlib**
  - import pathlib
  - Current_path = pathlib.Path().parent.resolve()
- **os**
  - import os
  - os.listdir()
  - os.getcwd()

3. Three Pillars

Three Pillars of Python for Data Analytics



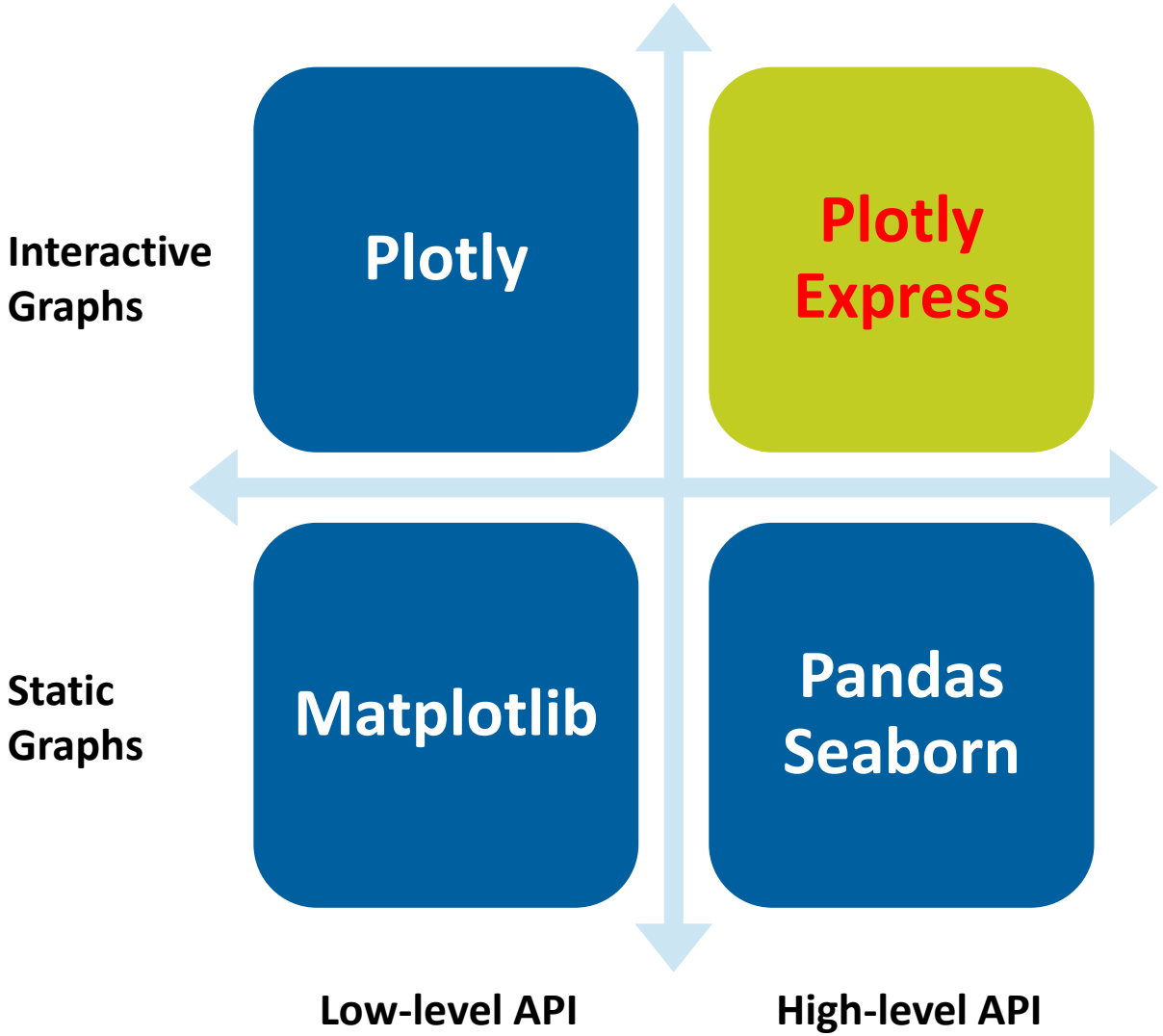
NumPy

- Homogeneous multidimensional array (nparray)
- Import numpy as np
- np.array([2,3,5,6]) – convert a list to an array
- np.arrange(start,stop, step) – produce an array

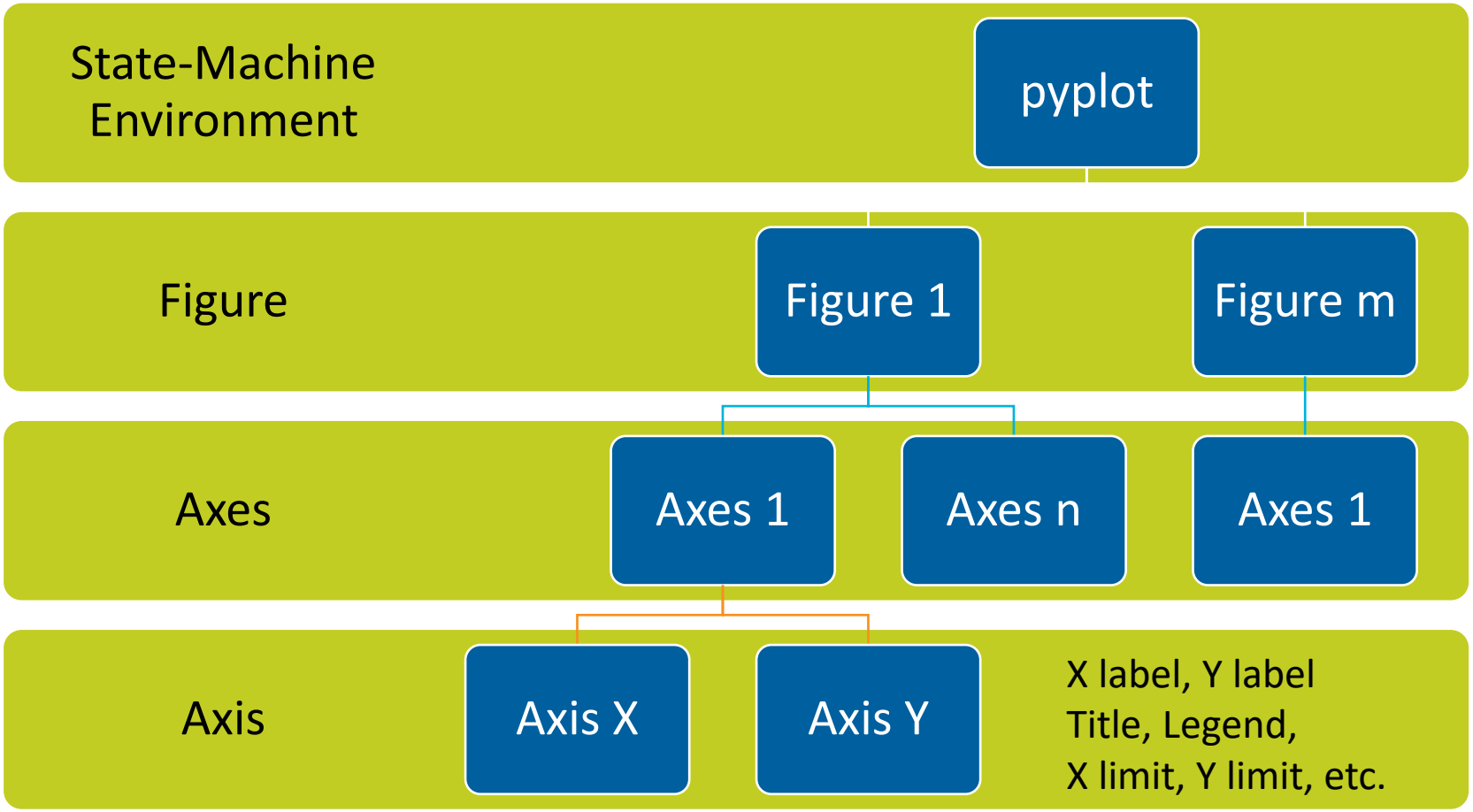
Python's Random vs NumPy Random

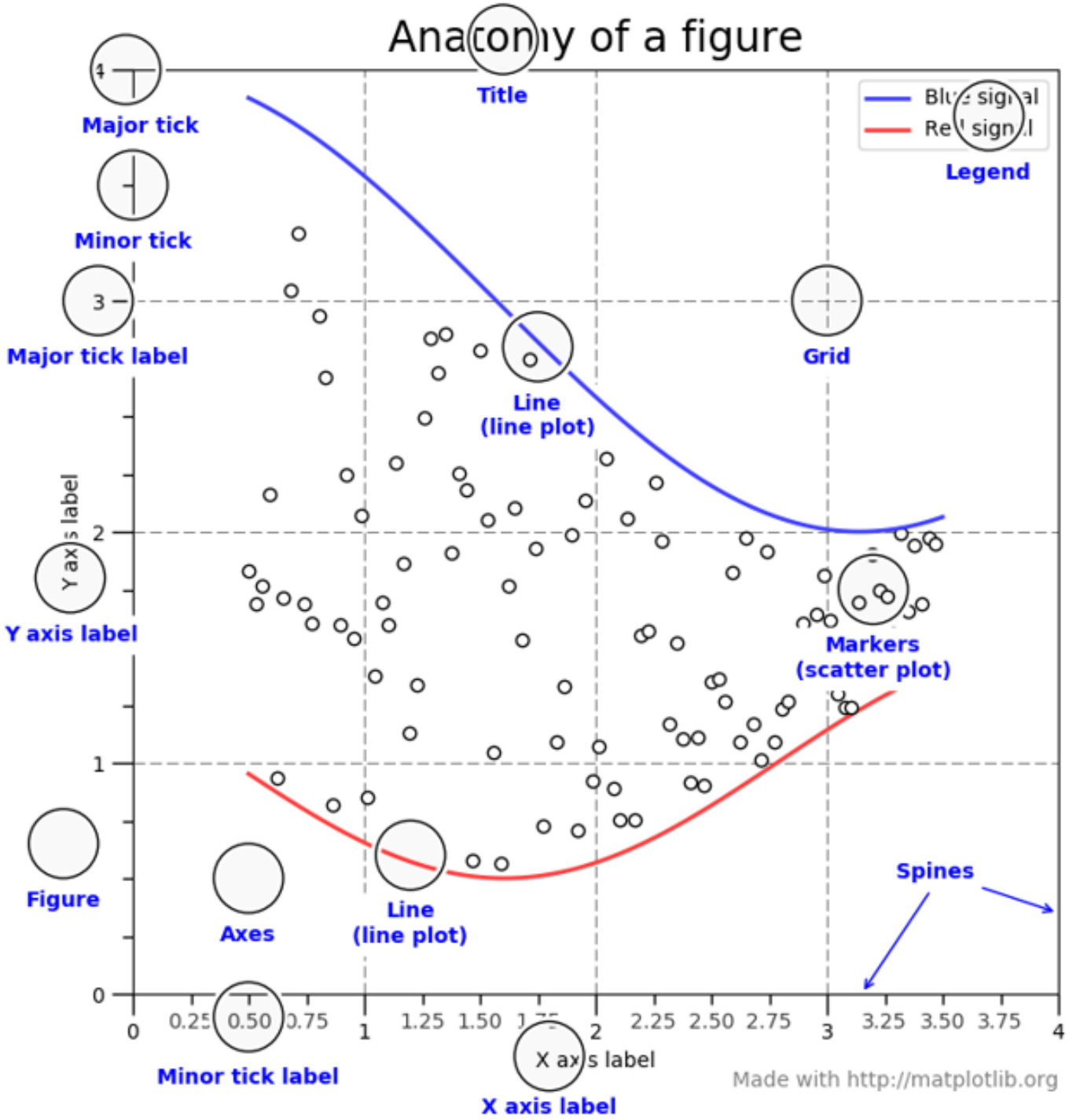
- **The former generates one random number at a time**
  - import random
  - random.randint(start, stop)
- **The latter can generate a list of random numbers at a time**
  - Optimized for performance
  - import numpy as np
  - np.random.randint

Data Visualization Libraries



Matplotlib Object Hierarchy




Anatomy of a figure
Title
Major tick
Minor tick
Major tick label
Legend
Grid
Line
(line plot)
Y axis label
Markers
(scatter plot)
Figure
Axes
Line
(line plot)
Spines
Minor tick label
X axis label
Made with http://matplotlib.org

Two Styles of Coding – Choose One

Procedural Style

- **Matlab style**
- **Methods from pyplot module**

Object Oriented Style

- **Use figure object**
- **Use axes object**
- **More control**

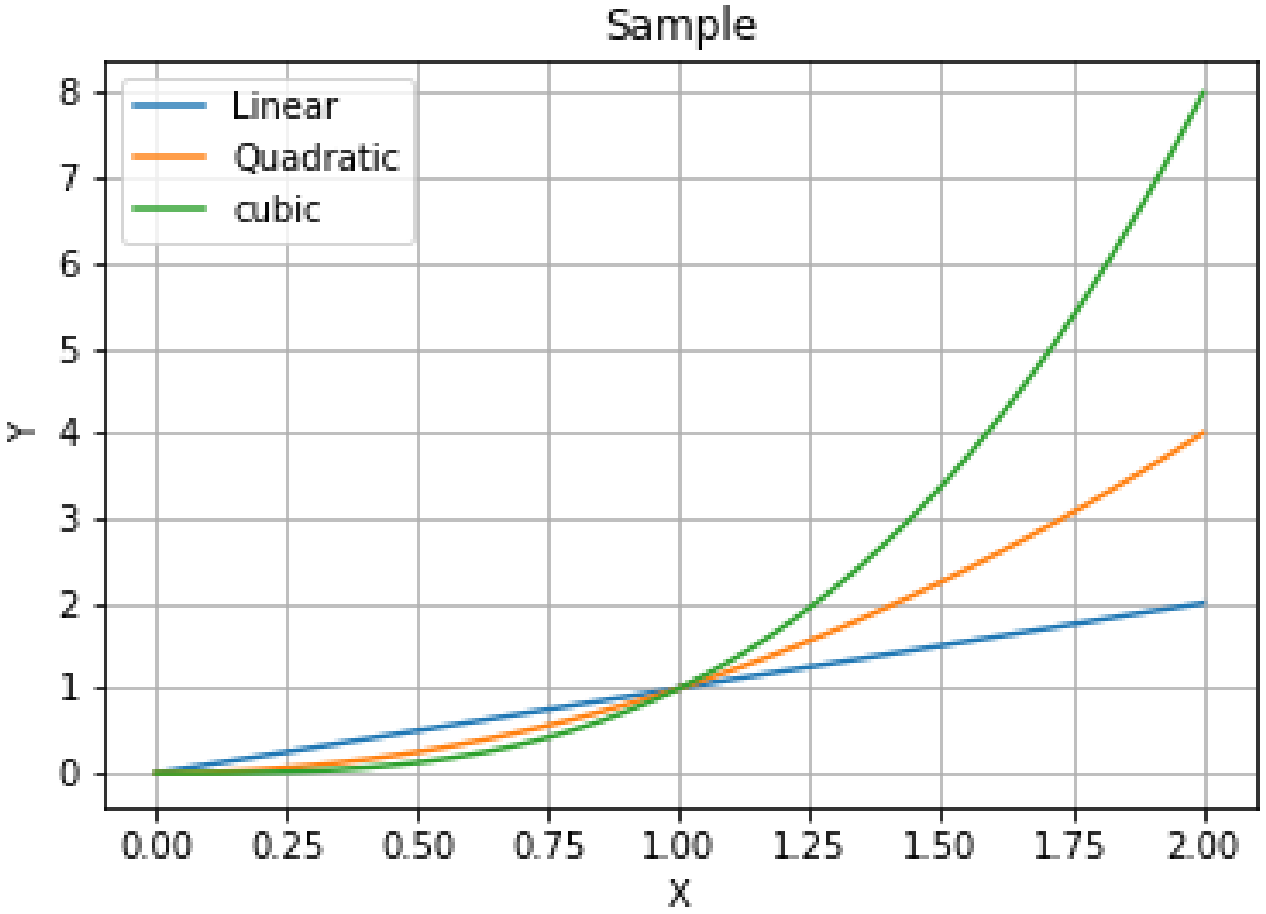
Procedural Style

- # assuming there is only one figure and one axes (default)
- x = np.linspace(0,2,100)
- plt.plot(x, x, label="linear") # legend
- plt.plot(x, x**2, label="quardratic")
- plt.plot(x, x**3, label="cubic")
- plt.xlabel("X") # X axis label
- plt.ylabel("Y") # Y axis label
- plt.title("This is a Test") # the title of the Axes
- plt.legend() # the legend of the axes

Object-Oriented Style

- x = np.linspace(0,2,100)
- # one figure can have multiple axes's. Here we create one axes.
- fig, ax = plt.subplots(4)
- Ax[0], ax[1]
- ax.xlabel = "X"
- ax.plot(x,x, label="Linear")
- ax.plot(x, x**2, label="Quadratic")
- ax.plot(x,x**3, label="cubic")
- ax.set_xlabel("X")
- ax.set_ylabel("Y")
- ax.set_title("Sample")
- ax.legend()

Different Styles, Same Results



Seaborn

- **Wrap around Matplotlib**
- **Hide complexity of Matplotlib**
- **High-level API**
- **Static Charts (not dynamic/interactive)**
- **For interactive viz, use Plotly/Plotly-Express**

Resources

- **NumPy Tutorial**
  - https://docs.scipy.org/doc/numpy/user/quickstart.html
- **Matplotlib Tutorial**
  - https://matplotlib.org/tutorials/introductory/pyplot.html#sphx-glr-tutorials-introductory-pyplot-py

4. Advanced Topics

Plotly – Interactive Data Visualization

- **plotly.graph_objects contains the main components of a plot:**
- **Figure contains all info for the visualization (data and layout)**
- **Layout contains all info for styling**
- **Scatter, Bar, Heatmap, etc, express different type of graphs.**
- **NOTE: These objects can always be swapped with python dicts**

Minimal plotly example:

```
import plotly.graph_objs as go

go.FigureWidget (data=[dict(x=[0,1,2], y=[3,4,2]
```

Plotly Express

- **Wrapper for Plotly**
- **High-level API**
- **Consistent syntax across different charts**
- **Integrate with Dash for web dashboarding**

Dashboard Development

- **Interactive Vizualization over Static Reports**
- **Team Collaboration over Individual Effort**
- **World Development Explorer**
  - http://www.worlddev.xyz
- **FOOD FOOTPRINT**
  - https://foodtprint.herokuapp.com/
  - https://github.com/InesRoque3/GroupV_project2

Dash

- **Python Framework for Analytical Web App**
- **Frontend: JS (Plotly/Plotly Express, React)**
- **Backend: Flask**

Minimal example

```
import dash
import dash_html_components as html
app = dash.Dash()
app.layout = html.Div('Hello World!' )
if __name__ == '__main__’:
  app.run_server()
```

Incorporate Visualizations

```
import dash_core_components as dcc
import plotly.graph_objs as go
app.layout = html.Div([
 html.H1('Hello EuroPython!' ),
 dcc.Graph(
 id='my-first-graph' ,
 figure=dict(data=[dict(x=[0,1,2], y=[3,4,2])]),
 )
])
```

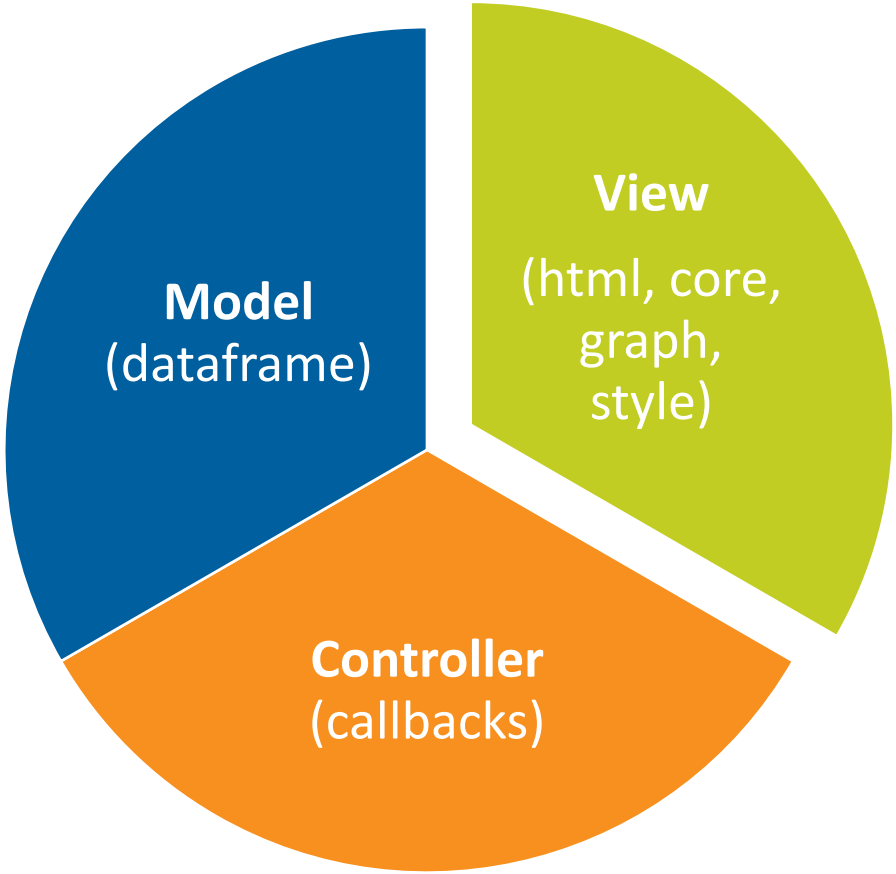
Callbacks for Dynamic Behavior

```
from dash.dependencies import Input, Output
app.layout = html.Div([
 dcc.Input(id='my-id', value='initial value' , type='text'),
 html.Div(id='my-div')
])
@app.callback (
 Output(component_id ='my-div', component_property ='children'),
 [Input(component_id ='my-id', component_property ='value')]
)
def update_output_div (input_value):
 return 'You\'ve entered "{}"' .format(input_value)
```

Style Sheet

- app.css.append_css({'external_url' : 'https://codepen.io/chriddyp/pen/bWLwg
- P.css'})
- app.layout = html.Div([
- dcc.Input(id='my-id', value='initial value' , type='text'),
- html.Div(id='my-div'),
-],
- className='container',
-)

MVC


Model
(dataframe)
View
(html, core,
graph,
style)
Controller
(callbacks)

Summary

- Html components (HTML tags)
- Core components (sliders, buttons, graphs)
- Plotly.py graph objects
- Callbacks (connect the pieces)
- CSS classes for pretty layout and styling

Deployment Options

- 1. Don't bother (1-person, local use only)
- 2. Know a little Flask
- 3. PaaS (e.g. Heroku, AWS)
- 4. Ask your engineer friend (aka Stack Overflow)
- 5. Ask Plotly (probably not for free)

Additional Considerations

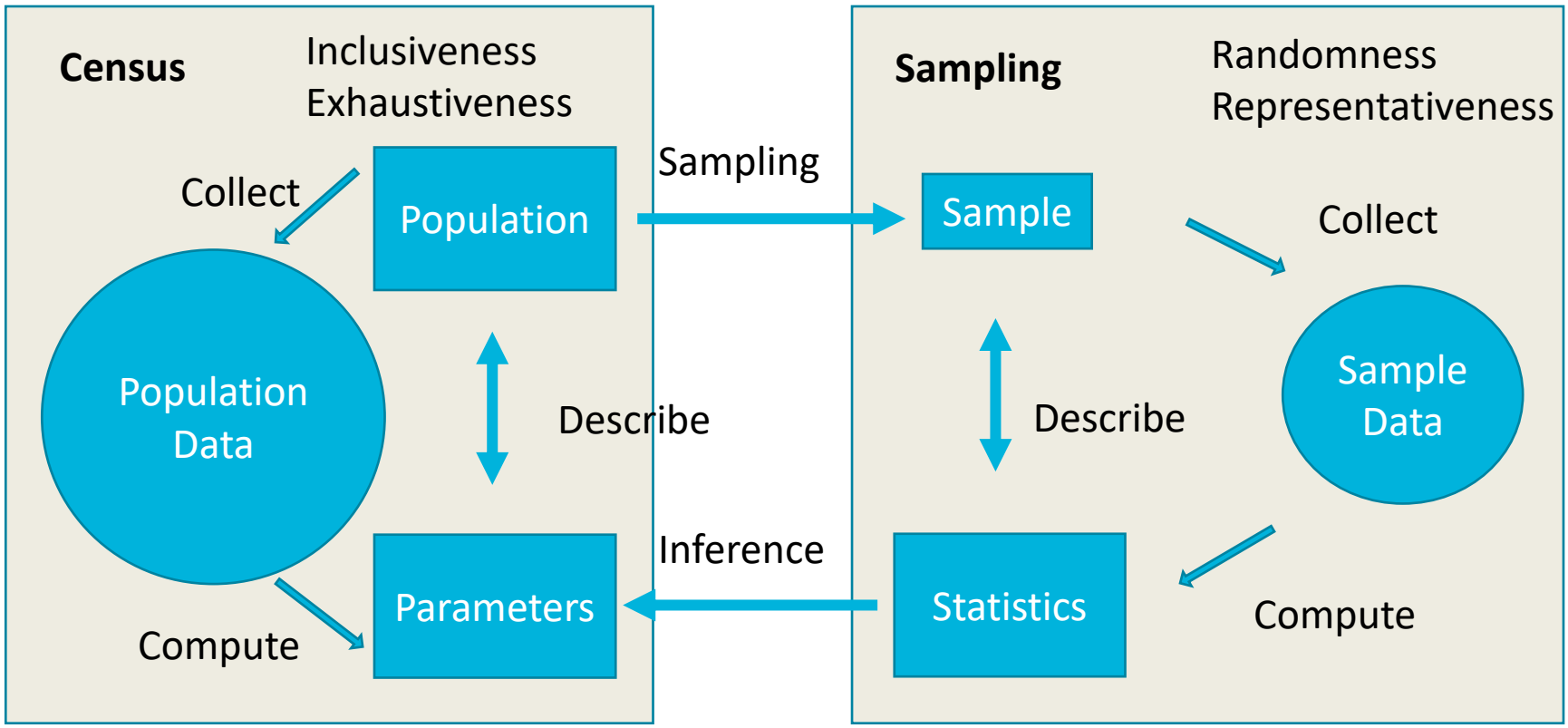
- External JS
- Caching
- Optional WebGL graphs for billion-point visualization (actually >15K)
- Live updates
- Authentication

Statistics

Three Aspects of Statistics

- **Collecting Data**
  - Census
  - Survey/Sampling
  - Design of Experiments
- **Organizing & Summarizing Data**
  - Numerical summary (descriptive statistics)
  - Tabular summary
  - Graphical summary (data visualizations)
- **Analyzing and interpreting Data**
  - Statistical inference (inferential statistics)
  - Prediction

Population vs Sample



1. μ
2. σ square
3. p

1. Centrality – mean
2. Spread – Variance
3. Proportion - Percentage

1. X bar
2. S square
3. P hat

Sampling Technique vs Design of Experiments

- **"Sampling techniques or methods deal with collecting data in the real world as it exists, namely through opinion polls, cross-sectional studies, surveys dealing with political and social issues, etc. Sampling is an essential part of everyday life and we return to it in the next section." p13**

- **"Design of Experiments is to design data collection in a more controlled setting in order to answer specific scientific questions, as in agricultural or clinical trials."**

Four Levels of Scale

- **Qualitative**
  - Nominal
  - Ordinal
- **Quantitative**
  - Interval
  - Ratio

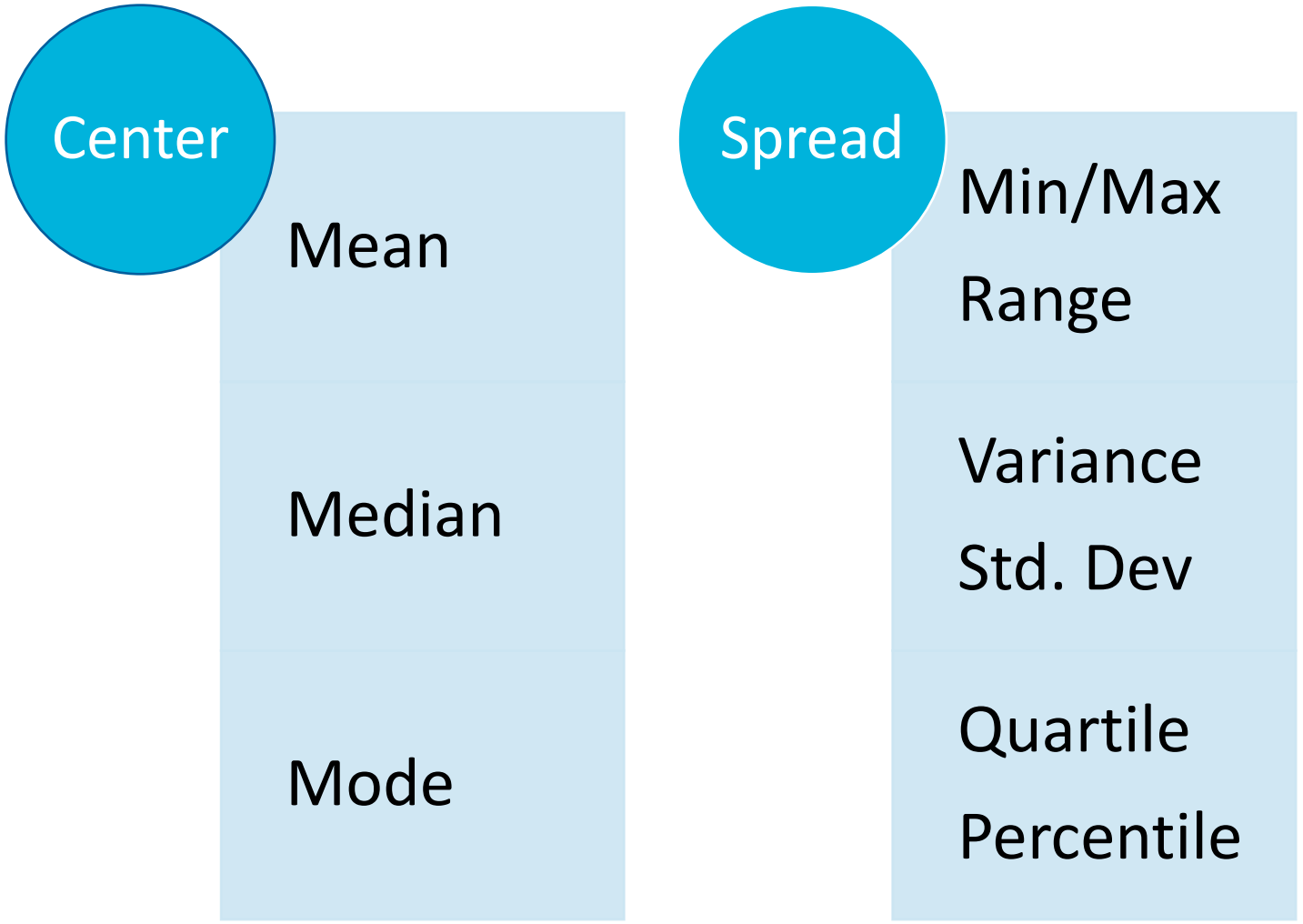
Descriptive Statistics

	Numpy	Statistics	Pandas	Scipy.stats
mean				
median				
min				
max				
Range				
Mode				
Variance				
Standard Deviation				
Percentile				
Quartile				

Percentile vs Quantile vs Quartile

- **See the link below**
- **https://stats.stackexchange.com/questions/156778/percentile-vs-quantile-vs-quartile**

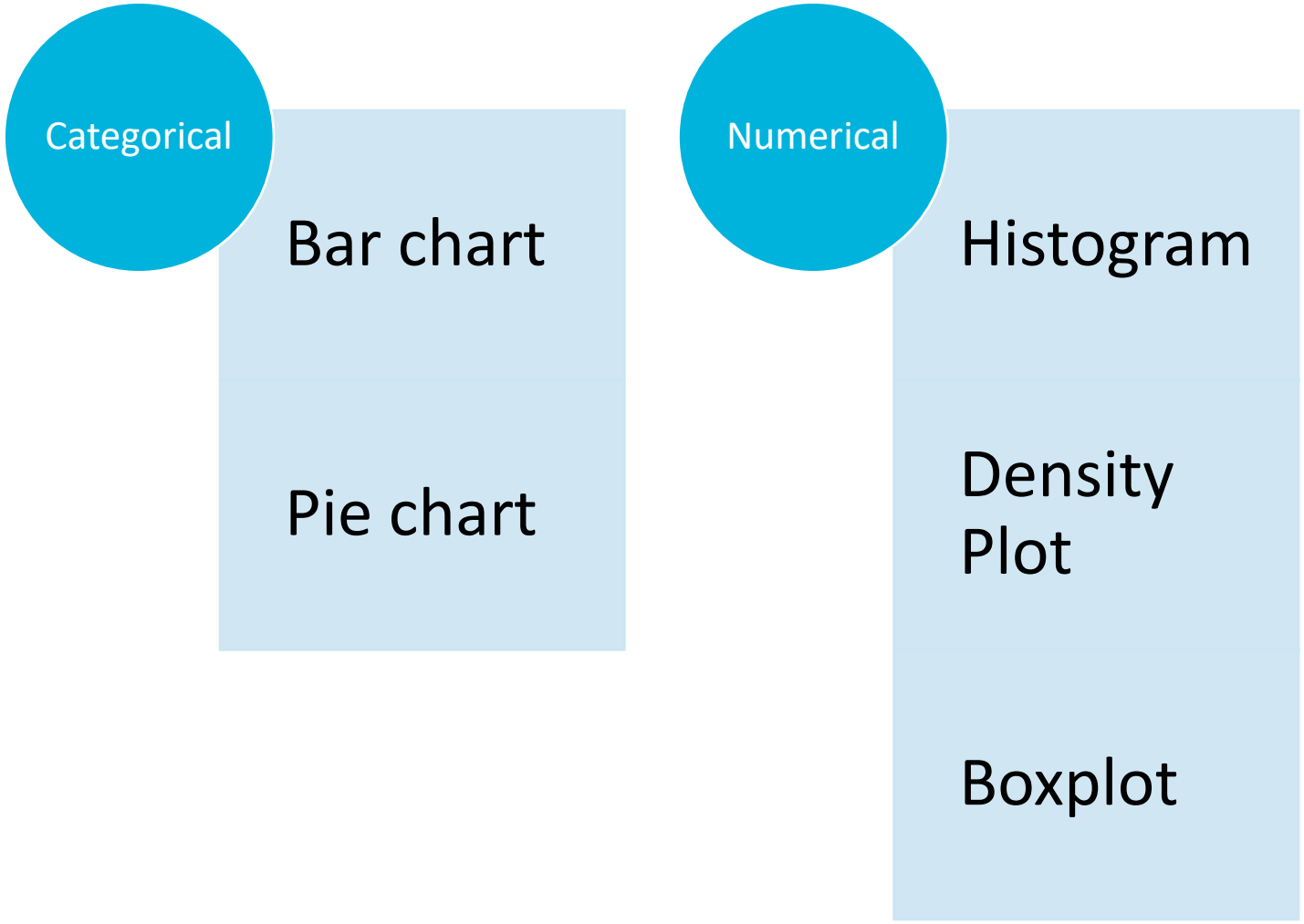
Summary Statistics – Numerical Variable



Summary Statistics – Categorial Variable

- **Frequency**
- **Proportion**

Visual Summary Statistics




thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
bedankt
dziękuję
go raibh maith agat
mochchakkeram
tapadh leat
kop khun krap
sukriya
terima kasih
grazie
arigatô
dakujem
спасибо
рахмат
Баярлалаа
mersi
barka
welalin
tack
spas
dank je
misaotra
matondo
paldies
grazzi
mahalo
faafetai lava
vinaka
спасибі
blagodaram
kiitos
dankie
dhanyavad
nanni
nandri
mauruuru
hvala
gràcie
köszönöm
akun
dankon
aciù
djiere dieuf
tau
дякую
mamnun
chokrane
murakoze
tenki
obrigada
asante
manana
хвала
enkosi
bayarlalaa
sobodi
dèkuji
mèsi
sagolun
didi madloba
kam sah hamnida
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
ありがとう
najis tuke
tanemirt
rahmet
rahmat
তোমাকে ধন্যবাদ
감사합니다
xièxie
ευχαριστώ
diolch
dhanyavadagalu
shukriya
merci
мерси
trugarez