01076103, 01076104

Programming Fundamental

Programming Project

Data visualization, Reading Data

Regular Expression

matplotlib

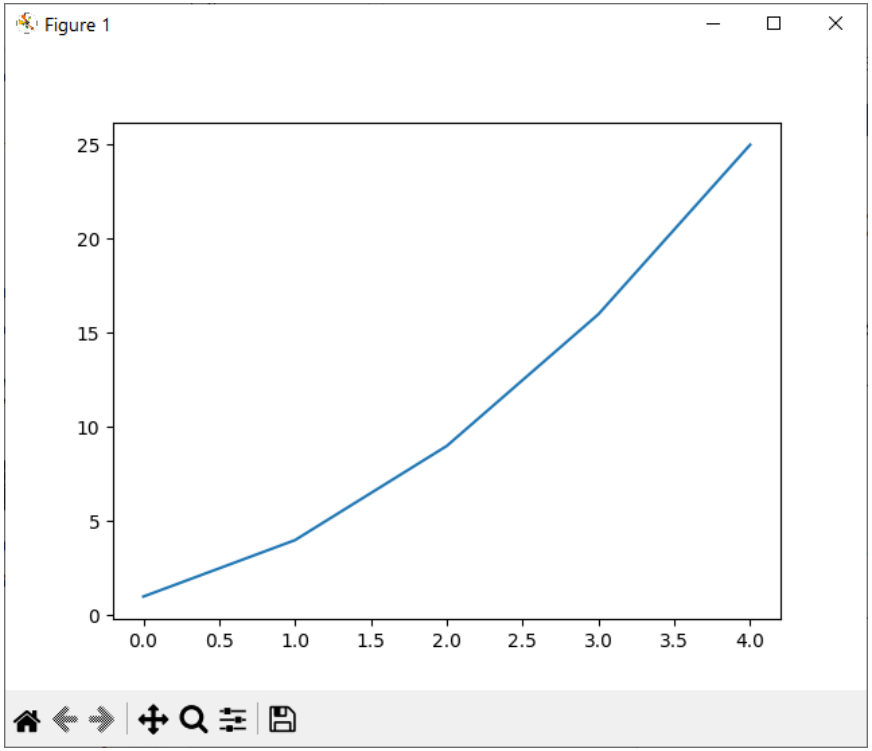
- งานที่ภาษา Python นำไปใช้มากที่สุดด้านหนึ่งคืองานด้าน Data Science
- เครื่องมือหนึ่งที่ใช้ใน Data Visualization คือ matplotlib ซึ่งไม่ได้ติดตั้งเป็น Library มาตรฐาน ดังนั้นต้องติดตั้งเข้าไปเอง
- ให้ใช้คำสั่ง pip install matplotlib จากนั้นจะ install matplotlib และ library อื่นๆ ที่มีการใช้งานร่วมกันเข้าไปเอง

```
C:\Users\khtha>pip install matplotlib
Defaulting to user installation because normal site-packages is not writeable
Collecting matplotlib
  Downloading matplotlib-3.6.1-cp310-cp310-win_amd64.whl (7.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.2/7.2 MB 25.6 MB/s eta 0:00:00
Collecting fonttools>=4.22.0
  Downloading fonttools-4.38.0-py3-none-any.whl (965 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 965.4/965.4 kB 30.8 MB/s eta 0:00:00
```

matplotlib

- ทดลองใช้งาน

```
import matplotlib.pyplot as plt

squares = [1, 4, 9, 16, 25]
plt.plot(squares)
plt.show()
```

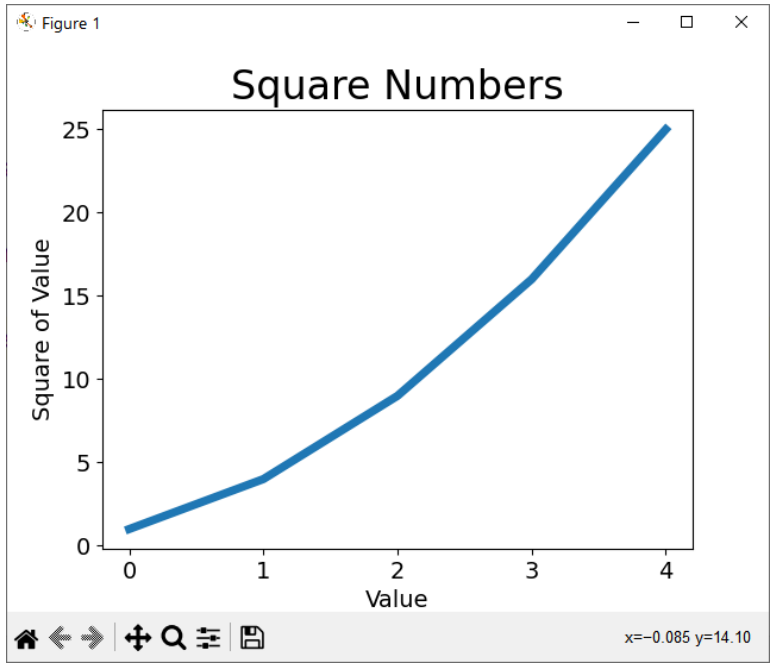


- กราฟจะเอาข้อมูลใน list มา Plot
- การแสดงผลใช้ show()

matplotlib

- เราจะปรับปรุงการแสดงผล โดยเพิ่ม (1) ความหนาของเส้น (2) ใส่ชื่อกราฟ (3) ใส่ label แกน x (4) ใส่ label แกน y

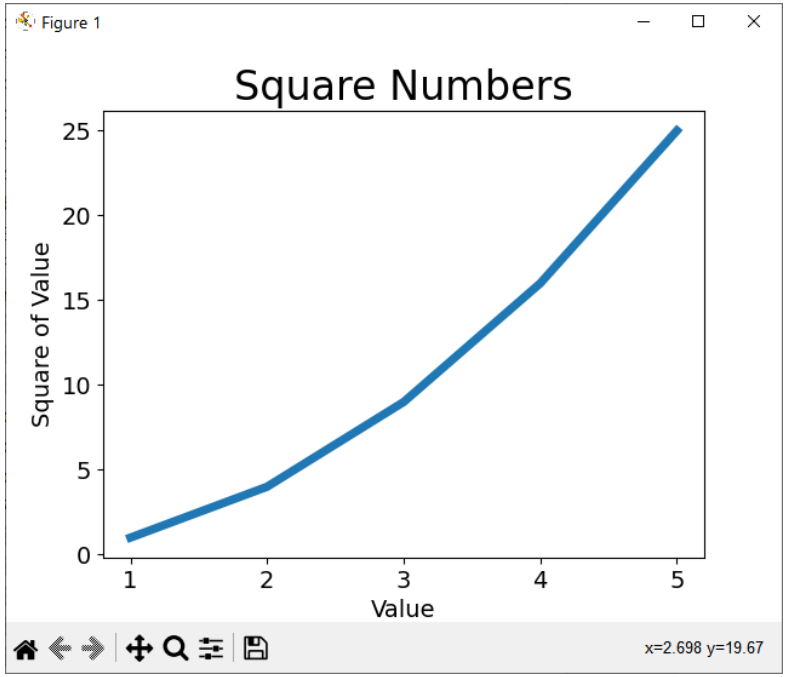
```
import matplotlib.pyplot as plt

squares = [1, 4, 9, 16, 25]
❶ plt.plot(squares, linewidth=5)

# Set chart title and label axes.
❷ plt.title("Square Numbers", fontsize=24)
❸ plt.xlabel("Value", fontsize=14)
plt.ylabel("Square of Value", fontsize=14)

# Set size of tick labels.
❹ plt.tick_params(axis='both', labelsize=14)

plt.show()
```



matplotlib

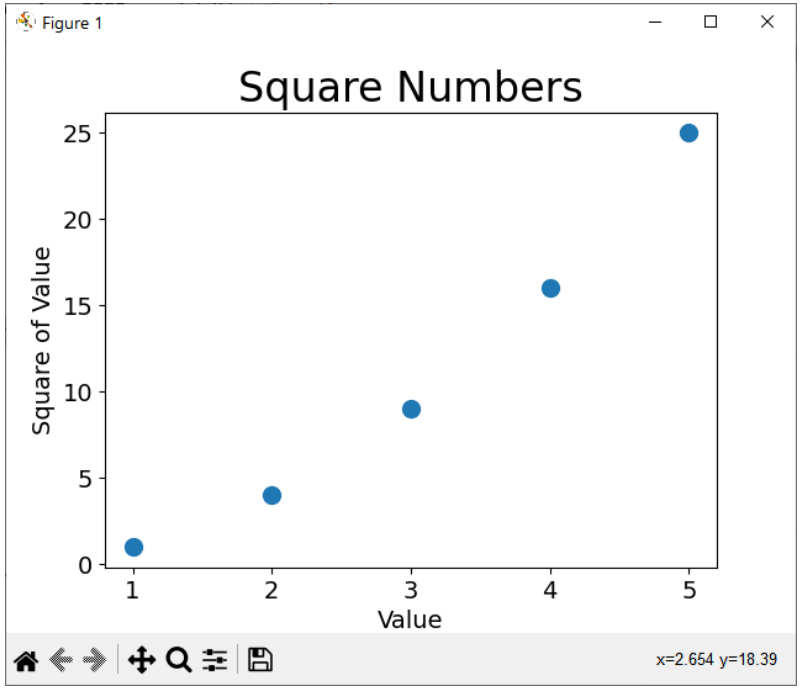
- รูปที่เกิดโปรแกรมในหน้าก่อนนี้ จะมีจุดผิดพลาด
- ถ้าดูรูปก่อนหน้าจะพบว่าค่าแกน x มีแค่ 4 และกราฟผิดพลาด
- เนื่องจากแกน x ไปเริ่มจาก 0
- ดังนั้นจึงต้องกำหนดค่าในแกน x ด้วย
- โปรแกรมจะทำงานถูกต้อง

```
import matplotlib.pyplot as plt

input_values = [1, 2, 3, 4, 5]
squares = [1, 4, 9, 16, 25]
plt.plot(input_values, squares, linewidth=5)
```



matplotlib

- ฟังก์ชันที่ใช้ plot จุด คือ scatter
- Argument = x, y และ s = ขนาด




```python
import matplotlib.pyplot as plt

x_values = [1, 2, 3, 4, 5]
y_values = [1, 4, 9, 16, 25]
plt.scatter(x_values, y_values, s=100)
# Set chart title and label axes.
plt.title("Square Numbers", fontsize=24)
plt.xlabel("Value", fontsize=14)
plt.ylabel("Square of Value", fontsize=14)
# Set size of tick labels.
plt.tick_params(axis='both', which='major', labelsize=14)
plt.show()
```

matplotlib

- เราสามารถกำหนดสีของจุดได้ โดยใช้ c=“color”

```
plt.scatter(x_values, y_values, c='red', edgecolor='none', s=40)
```

- สามารถใช้สีแบบไล่เฉดได้

```
plt.scatter(x_values, y_values, c=y_values, cmap=plt.cm.Blues,
edgecolor='none', s=40)
```

- สามารถ save เป็นไฟล์ได้

```
plt.savefig('squares_plot.png', bbox_inches='tight')
```

matplotlib

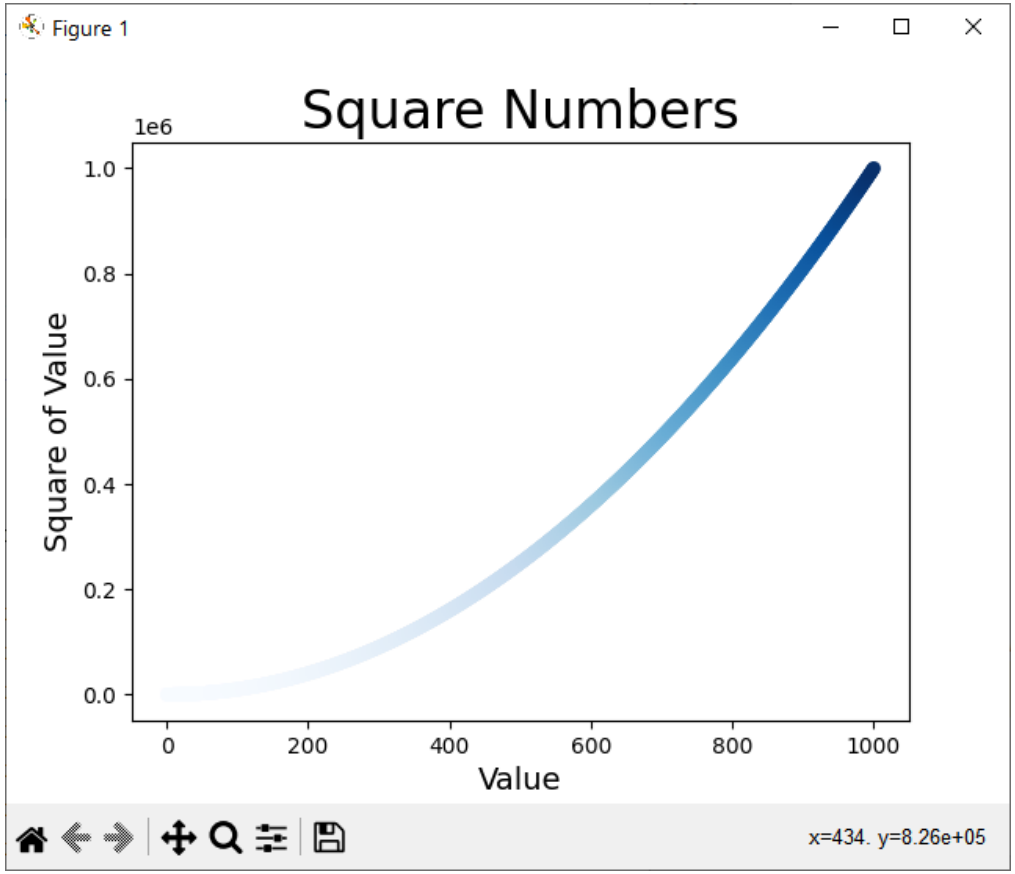
- Exercise

  สร้าง List x = 1->1000

  y = x**2

  กำหนด c = y

  cmap=plt.cm.Blues



PyGal

- PyGal เป็น Library สำหรับสร้างภาพกราฟิก ซึ่งจะต้องติดตั้งก่อนเช่นกัน

```
C:\Users\khtha>pip install pygal
Defaulting to user installation because normal site-packages is not writeable
Collecting pygal
  Downloading pygal-3.0.0-py2.py3-none-any.whl (129 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 129.4/129.4 kB 1.9 MB/s eta 0:00:00
Installing collected packages: pygal
Successfully installed pygal-3.0.0
```

- สมมติว่าจะทดลองสร้าง histogram ของลูกเต๋า จะสร้าง function roll_dice

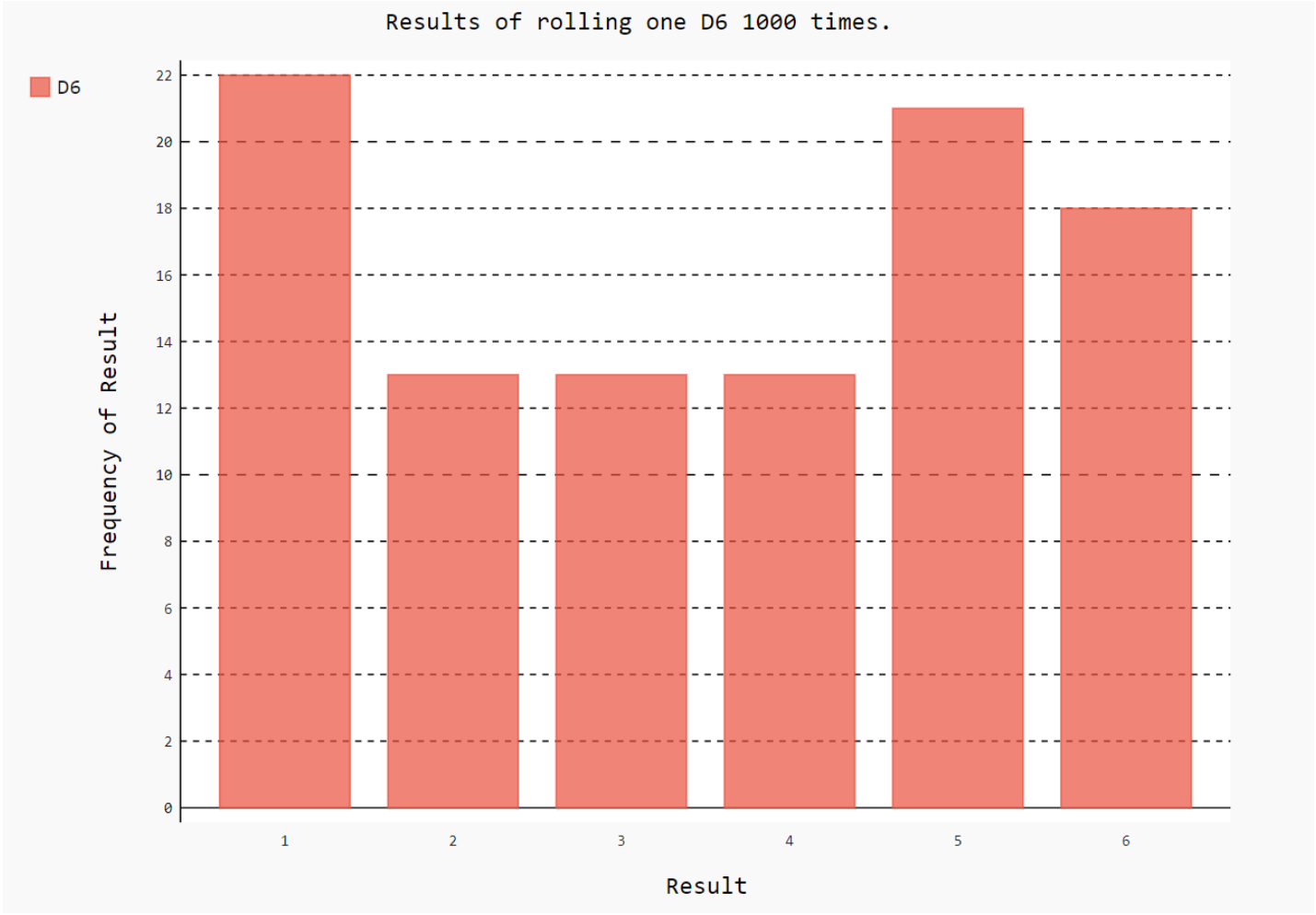
```python
from random import randint

def roll_dice():
    num_sides = 6
    return randint(1, self.num_sides)

for roll_num in range(100):
    result = roll_dice()
    results.append(result)
```

PyGal

- จากนั้นนำข้อมูลไปหาความถี่ของแต่ละหน้าลูกเต๋า
- แล้วนำไปสร้างกราฟ ซึ่งจะได้ output เป็นไฟล์

```
frequencies = []
for value in range(1, 7):
    frequency = results.count(value)
    frequencies.append(frequency)

# Visualize the results.
hist = pygal.Bar()
hist.title = "Results of rolling one D6 1000 times."
hist.x_labels = ['1', '2', '3', '4', '5', '6']
hist.x_title = "Result"
hist.y_title = "Frequency of Result"
hist.add('D6', frequencies)
hist.render_to_file('die_visual.svg')
```

PyGal

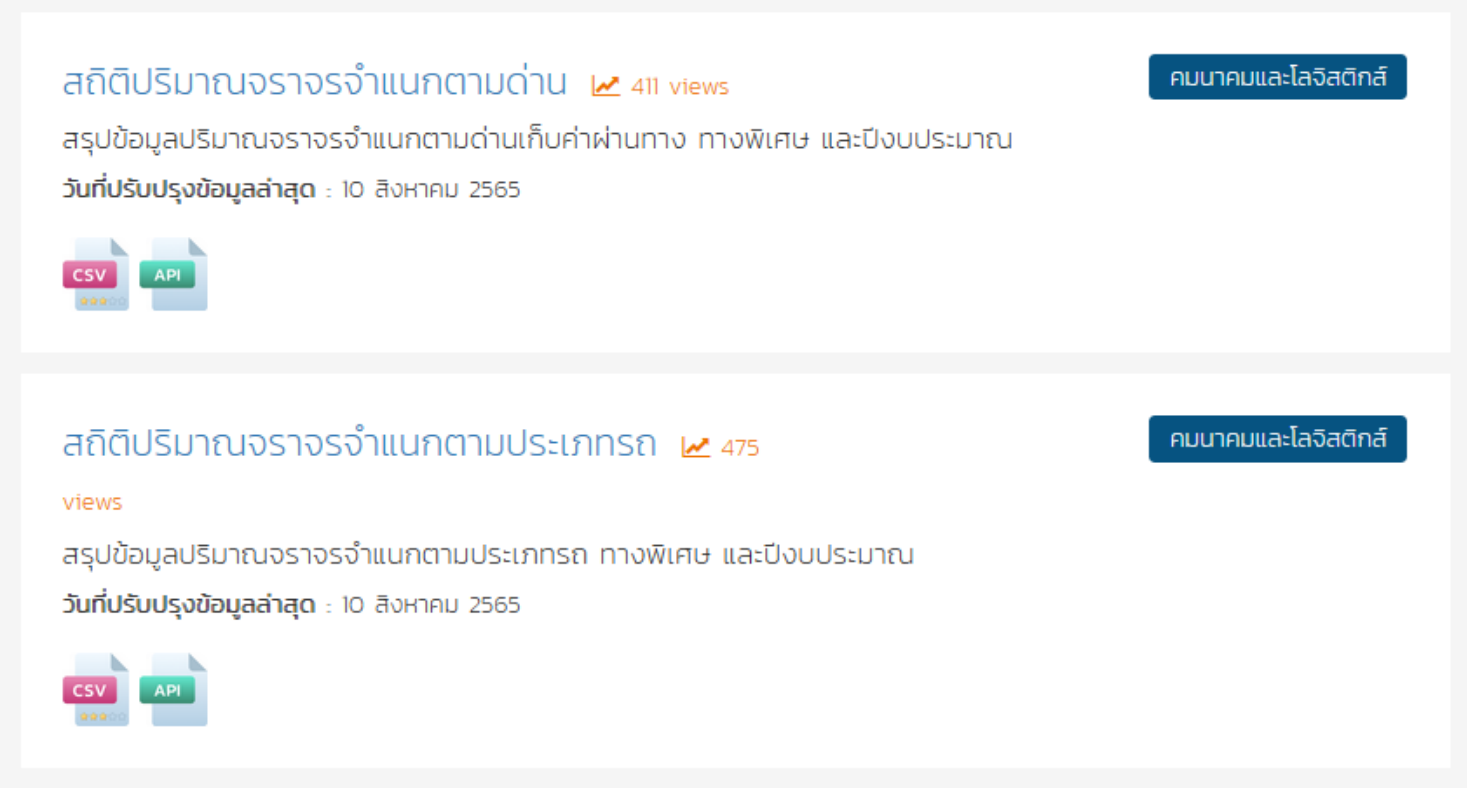

Results of rolling one D6 1000 times.
D6
Frequency of Result
0
2
4
6
8
10
12
14
16
18
20
22
1
2
3
4
5
6
Result

PyGal

- **Exercise** ให้เขียนโปรแกรมตามตัวอย่าง แต่ให้เพิ่มลูกเต๋าเป็น 2 ลูก

Reading Data

- ในประเทศไทยเริ่มมีการเผยแพร่ข้อมูลสู่สาธารณะมากขึ้น เช่น ในเว็บ www.data.go.th, https://data.bangkok.go.th/



Reading Data

- ชนิดข้อมูลที่มีการเผยแพร่มาก คือไฟล์ csv
- CSV ย่อมาจาก comma-separated values
- ตัวอย่างไฟล์

```
year,highway_name,car_type,traffic
2564,ฉลองรัช,4 ล้อ,62223742
2564,ฉลองรัช,6-10 ล้อ,552568
2564,ฉลองรัช,> 10 ล้อ,143909
2564,ทางพิเศษกาญจนาภิเษก(บางพลี-สุขสวัสดิ์),4 ล้อ,60217745
2564,ทางพิเศษกาญจนาภิเษก(บางพลี-สุขสวัสดิ์),6-10 ล้อ,6604450
2564,ทางพิเศษกาญจนาภิเษก(บางพลี-สุขสวัสดิ์),> 10 ล้อ,3921853
2564,บูรพาวิถี,4 ล้อ,37323858
2564,บูรพาวิถี,6-10 ล้อ,1639427
2564,บูรพาวิถี,> 10 ล้อ,411080
2564,ศรีรัช,4 ล้อ,171587253
```

Reading Data

- ในการอ่านไฟล์ CSV จะมี Library ให้ใช้งานชื่อ csv
- โปรแกรมจะเปิดไฟล์ จากนั้นส่งไฟล์ต่อให้ csv อ่านต่อ โดยข้อมูลจากการอ่านจะอยู่ใน reader โดยจะเป็น object ที่มีชุดของ List ที่มีข้อมูลแต่ละบรรทัด
- ฟังก์ชัน next เป็นฟังก์ชันพิเศษที่ใช้กับ object โดยจะคืนข้อมูลถัดไปมาให้ ซึ่งกรณีนี้คือ บรรทัดแรก ซึ่งเป็น header นั่นเอง

```
import csv

filename = 'tbl_traffic_bycar.csv'
with open(filename, encoding="utf8") as f:
    reader = csv.reader(f)
    header_row = next(reader)
    for index, column_header in enumerate(header_row):
        print(index, column_header)
```

```
0 year
1 highway_name
2 car_type
3 traffic
```

Reading Data

- ในบรรทัดต่อจาก header จะเป็นข้อมูล
- จากรูปแบบข้อมูลจะเห็นว่ามี ชื่อด่าน ขนาดรถ และ จำนวนคันที่ผ่าน
- ซึ่งรูปแบบข้อมูล จะเห็นว่าเหมาะสมจะเก็บใน list หรือ dictionary

เช่น [‘ฉลองรัช’, [‘4 ล้อ’, 62223742], [‘6-10 ล้อ’, 552568], [‘> 10 ล้อ’, 143909]]

หรือ {‘ฉลองรัช’ : {‘4 ล้อ’: 62223742},{‘6-10 ล้อ’: 552568},{‘> 10 ล้อ’: 143909}}

Reading Data

- สมมติว่าเลือกเป็น list จะได้ข้อมูลดังนี้

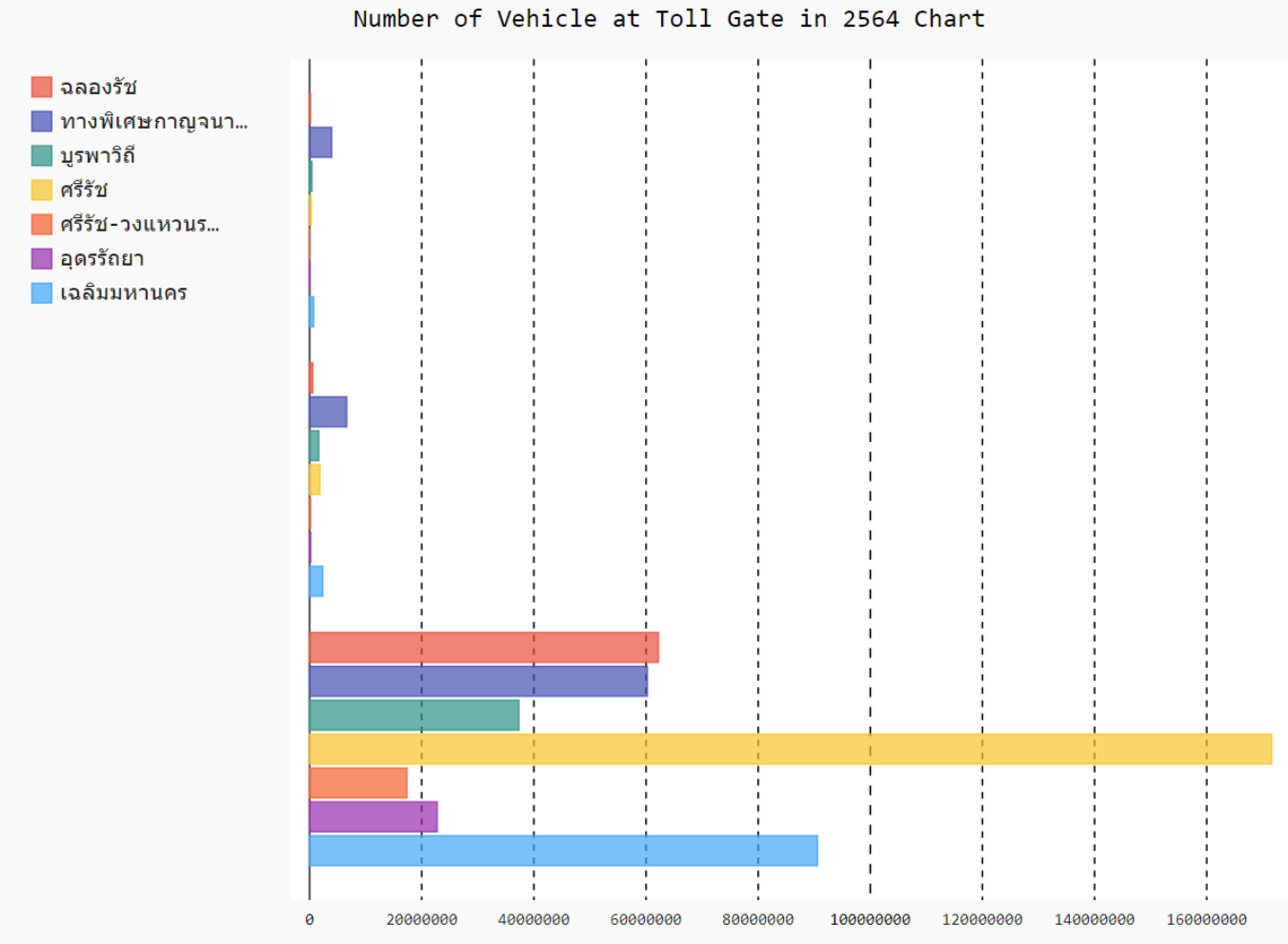
[['ฉลองรัช', ['4 ล้อ', '62223742'], ['6-10 ล้อ', '552568'], ['> 10 ล้อ', '143909']],
['ทางพิเศษกาญจนาภิเษก(บางพลี-สุขสวัสดิ์)', ['4 ล้อ', '60217745'], ['6-10 ล้อ', '6604450'], ['> 10 ล้อ', '3921853']],
['บูรพาวิถี', ['4 ล้อ', '37323858'], ['6-10 ล้อ', '1639427'], ['> 10 ล้อ', '411080']],
['ศรีรัช', ['4 ล้อ', '171587253'], ['6-10 ล้อ', '1838074'], ['> 10 ล้อ', '271263']],
['ศรีรัช-วงแหวนรอบนอกกรุงเทพมหานคร', ['4 ล้อ', '17358529'], ['6-10 ล้อ', '160722'], ['> 10 ล้อ', '11390']],
['อุดรรัถยา', ['4 ล้อ', '22749340'], ['6-10 ล้อ', '189358'], ['> 10 ล้อ', '41408']],
['เฉลิมมหานคร', ['4 ล้อ', '90569430'], ['6-10 ล้อ', '2335498'], ['> 10 ล้อ', '718874']]]

Reading Data

- จากนั้นนำมาทำเป็นกราฟ สมมติว่าเราจะใช้ Bar Chart

```
horizontal_chart = pygal.HorizontalBar()
horizontal_chart.title = 'Number of Vehicle at Toll Gate in 2564 Chart'
for i in tolls:
    horizontal_chart.add(i[0],i[1:])
horizontal_chart.render_to_file('horizontal_line_chart.svg')
```

Reading Data



Reading Data

- ดูอีกตัวอย่าง คราวนี้จะใช้ไฟล์ค่าอุณหภูมิ (sitka_weather_2018_simple.csv)
- เก็บข้อมูล 2 ตัว คือ date กับ high

```
import csv
from datetime import datetime

from matplotlib import pyplot as plt

filename = 'sitka_weather_2018_simple.csv'
with open(filename) as f:
    reader = csv.reader(f)
    header_row = next(reader)

    # Get dates and high temperatures from this file.
    dates, highs = [], []
    for row in reader:
        current_date = datetime.strptime(row[2], '%Y-%m-%d')
        dates.append(current_date)
        high = int(row[5])
        highs.append(high)
```

Reading Data

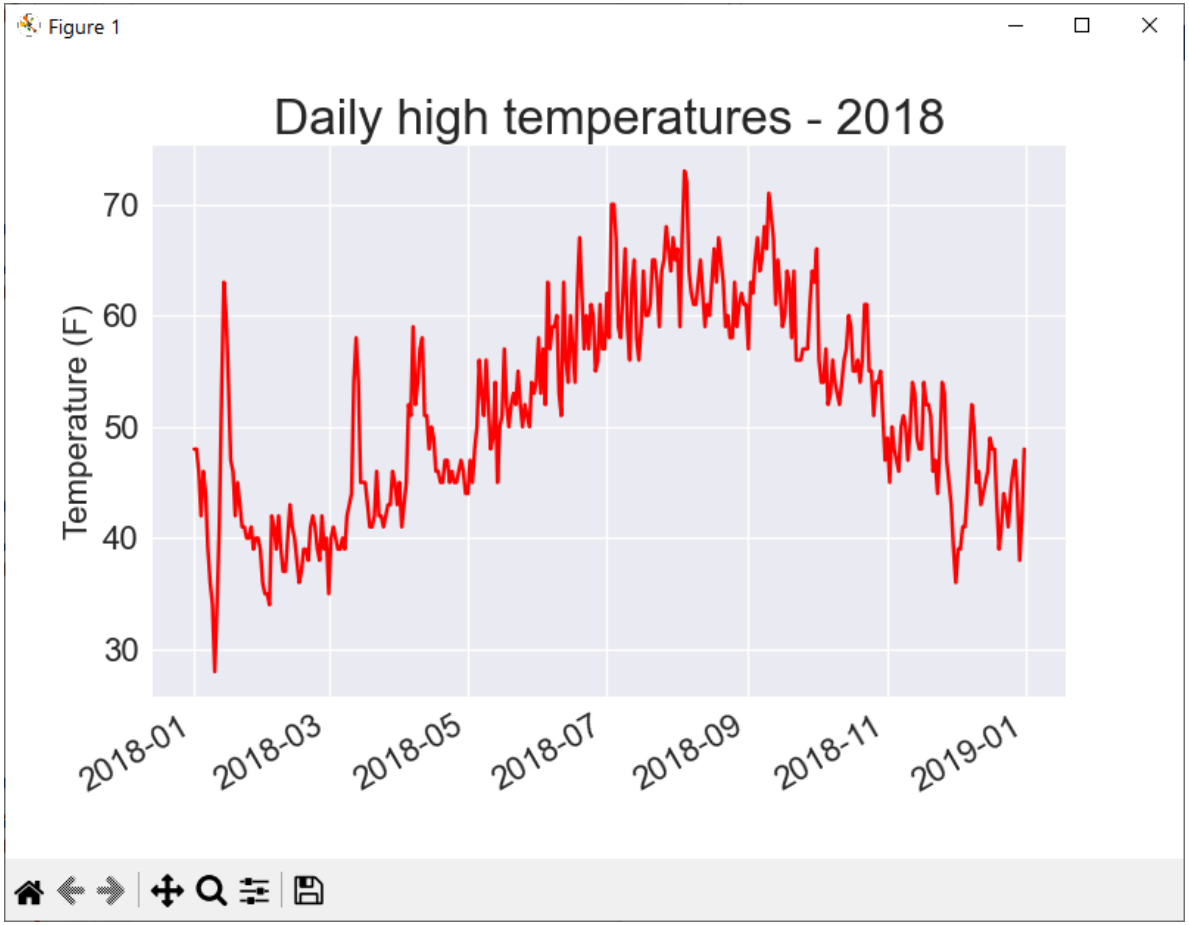
- จากนั้นใช้ matplotlib ทำหน้าที่ plot กราฟ
- รายละเอียดเกี่ยวกับ style

https://matplotlib.org/stable/tutorials/introductory/customizing.html#customizing-with-style-sheets

```
# Plot the high temperatures.
plt.style.use('seaborn-v0_8')
fig, ax = plt.subplots()
ax.plot(dates, highs, c='red')

# Format plot.
plt.title("Daily high temperatures - 2018", fontsize=24)
plt.xlabel('', fontsize=16)
fig.autofmt_xdate()
plt.ylabel("Temperature (F)", fontsize=16)
plt.tick_params(axis='both', which='major', labelsize=16)

plt.show()
```

Reading Data

- จะได้กราฟ



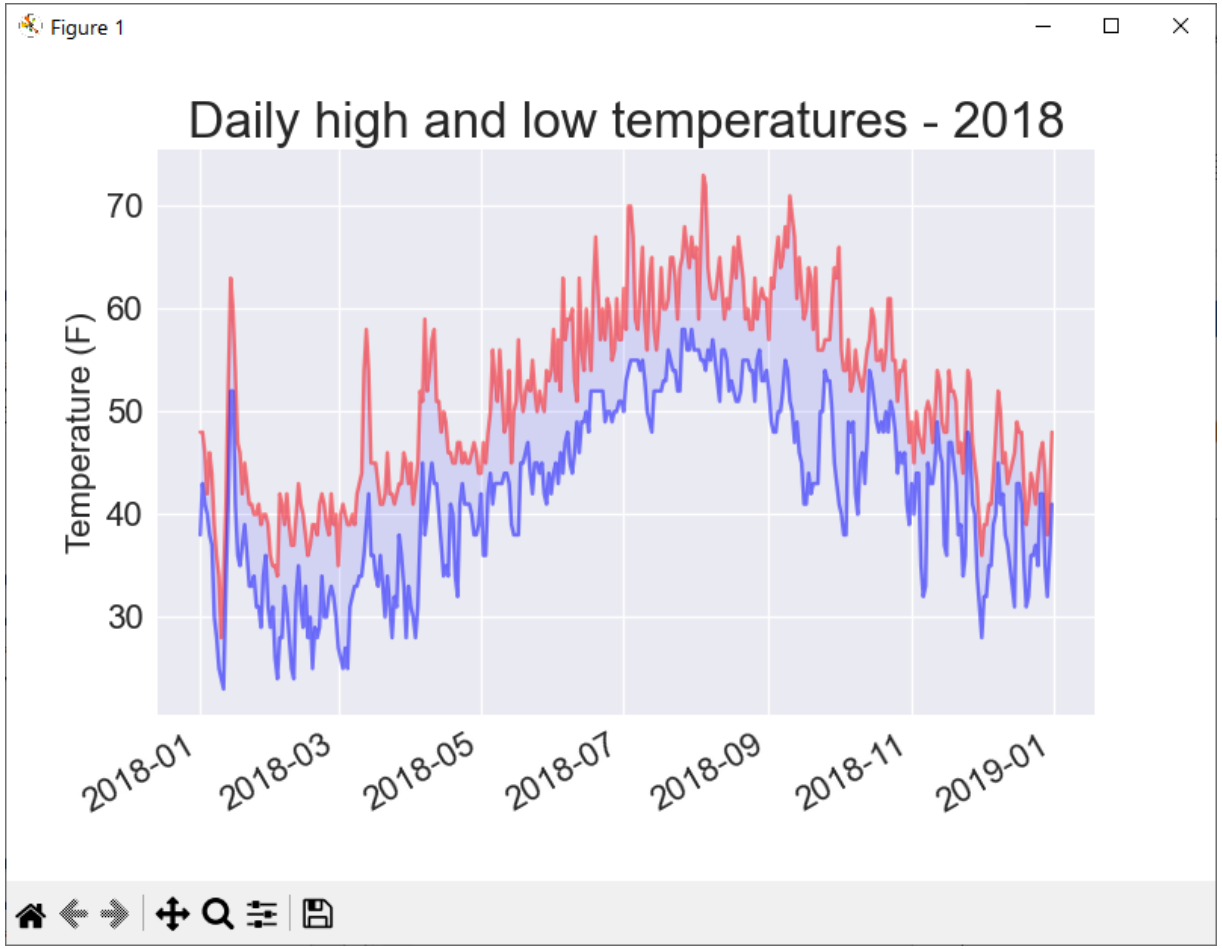
Reading Data

- กรณีที่ต้องการแสดงทั้ง High และ Low

```
# Plot the high and low temperatures.
plt.style.use('seaborn')
fig, ax = plt.subplots()
ax.plot(dates, highs, c='red', alpha=0.5)
ax.plot(dates, lows, c='blue', alpha=0.5)
plt.fill_between(dates, highs, lows, facecolor='blue', alpha=0.1)

# Format plot.
plt.title("Daily high and low temperatures - 2018", fontsize=24)
plt.xlabel('', fontsize=16)
fig.autofmt_xdate()
plt.ylabel("Temperature (F)", fontsize=16)
plt.tick_params(axis='both', which='major', labelsize=16)

plt.show()
```

Reading Data

- จะได้กราฟ (สามารถ fill ระหว่าง High และ Low ได้)



Reading Data

- การอ่านไฟล์ Excel จะใช้ Library ช่วยเช่นเดียวกัน โดยใช้ library ชื่อ openpyxl

```
import openpyxl

wb = openpyxl.load_workbook('example.xlsx')
sheetNames=wb.sheetnames

for name in sheetNames:
   print(name)

sheet = wb["Sheet1"]
max_row = sheet.max_row
max_col = sheet.max_column
print(max_col,max_row)

for i in range(1, max_row+1, 2):
    print(i, sheet.cell(row=i, column=2).value)
```

Reading Data

- นอกเหนือจากไฟล์ csv แล้ว อีกรูปแบบไฟล์ที่นิยมใช้คือ json
- JSON ย่อมาจาก JavaScript Object Notation เป็นมาตรฐานการแลกเปลี่ยนข้อมูลระหว่าง Server และ Client ที่ได้รับความนิยมในปัจจุบัน
- อักขระมาตรฐานของ JSON
  - เครื่องหมาย “:” ใช้สำหรับแยกค่า name และ value
  - เครื่องหมาย “,” ใช้สำหรับแยกข้อมูล name-value ในแต่ละคู่
  - เครื่องหมาย “{” และ “}” ระบุว่าข้อมูลเป็น Object
  - เครื่องหมาย “[” และ “]” ระบุว่าข้อมูลเป็นอาเรย์
  - เครื่องหมาย “ ” (double quotes) ใช้เขียนค่า name-value ใน JSON

Reading Data

- ข้อมูล 1 คู่

  "name" : "value"

- ข้อมูล 2 คู่ ใช้ เครื่องหมายคอมมา , (comma) ในการแยกเป็นคู่

  "name" : "value", "name" : "value", "name": "value"

- ใช้เครื่องหมาย { } ในการระบุว่าเป็น Object

```
{
  "name" : "Dwayne Johnson",
  "email" : “johnson@email.com",
}
```

Reading Data

- ชนิดข้อมูลของ JSON มี 6 ชนิด คือ

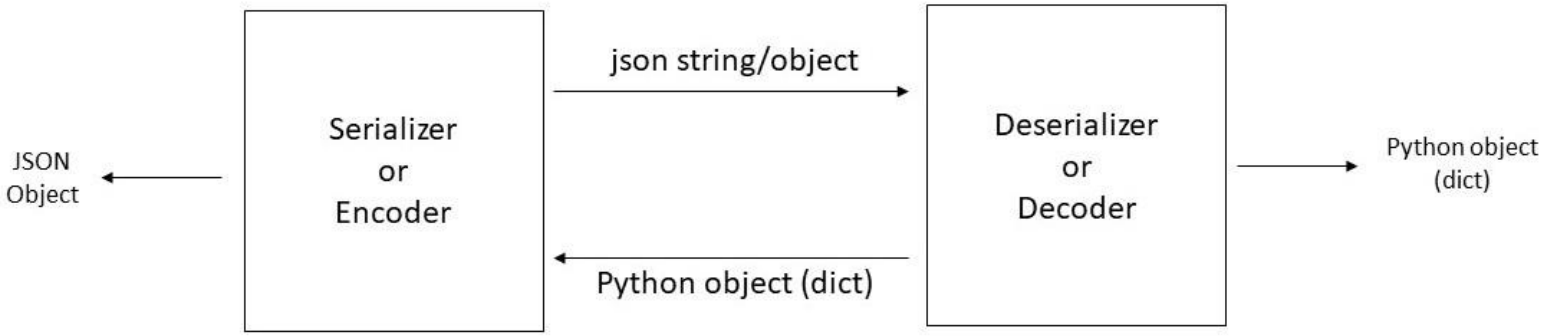
1. strings
2. numbers
3. objects
4. arrays
5. Boolean
6. null or empty

```
{
   "text" : "This is Sting",
   "number" : 210,
   "object" :  {
                    "name" : "abc",
                    "popularity" : "immense"
               },
   "arary" : ["1","2","3"],
   "empty" : ,
   "booleans" : true
}
```

Reading Data

- เนื่องจาก JSON มีลักษณะเป็นคู่ key:value คล้ายกับกับ dictionary ใน python จึงมักใช้ dictionary ในการแปลงเป็น JSON
- การแปลง Python Object ไปเป็น JSON จะเรียกว่า Serialization
- การแปลง JSON มาเป็น Python Object จะเรียกว่า Deserialization

กระบวนการ Serialization และ Deserialization



Reading Data

- การแปลง Python Object (Dict) ไปเป็น JSON String หรือ Object ทำได้โดยเรียกใช้ json.dumps()

```python
import json

test_dict = {
    "name": "Python",
    "author": "Guido Van Rossum",
    "year": 1990,
    "frameworks": ["Flask", "Django"],
    "libraries": ["Pandas", "Numpy", "Matplotlib", "Requests"]
}

print(type(test_dict))
j_string = json.dumps(test_dict)
print(j_string)
print(type(j_string))
```

Reading Data

- การแปลง JSON String ไปเป็น Python Object (Dict) ทำได้โดยเรียกใช้ json.loads()

```
import json

prog_string = '''
    {
        "name": "Python",
        "author": "Guido Van Rossum",
        "year": 1990,
        "frameworks": ["Flask", "Django"],
        "libraries": ["Pandas", "Numpy", "Matplotlib", "Requests"]
    }'''

print(type(prog_string))
prog_dict = json.loads(prog_string)
print(prog_dict)
print(type(prog_dict))
print(prog_dict["name"])
```

Reading Data

- ในการเรียกใช้ API จากที่ต่างๆ เพื่อนำข้อมูลมาใช้งาน จะใช้ Library requests
- ให้ติดตั้ง Library ชื่อ requests และรันโปรแกรมนี้

```
import requests

r = requests.get('https://covid19.ddc.moph.go.th/api/Cases/today-cases-all')
print(r)
print(dir(r))
```

- จะพบว่าได้ผลดังนี้

<Response [200]>
['__attrs__', '__bool__', '__class__', '__delattr__', '__dict__', '__dir__', '__doc__', '__enter__', '__eq__', '__exit__', '__format__', '__ge__', '__getattribute__', '__getstate__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__lt__', '__module__', '__ne__', '__new__', '__nonzero__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__setstate__', '__sizeof__', '__str__', '__subclasshook__', '__weakref__', '_content', '_content_consumed', '_next', 'apparent_encoding', 'close', 'connection', 'content', 'cookies', 'elapsed', 'encoding', 'headers', 'history', 'is_permanent_redirect', 'is_redirect', 'iter_content', 'iter_lines', 'json', 'links', 'next', 'ok', 'raise_for_status', 'raw', 'reason', 'request', 'status_code', 'text', 'url']

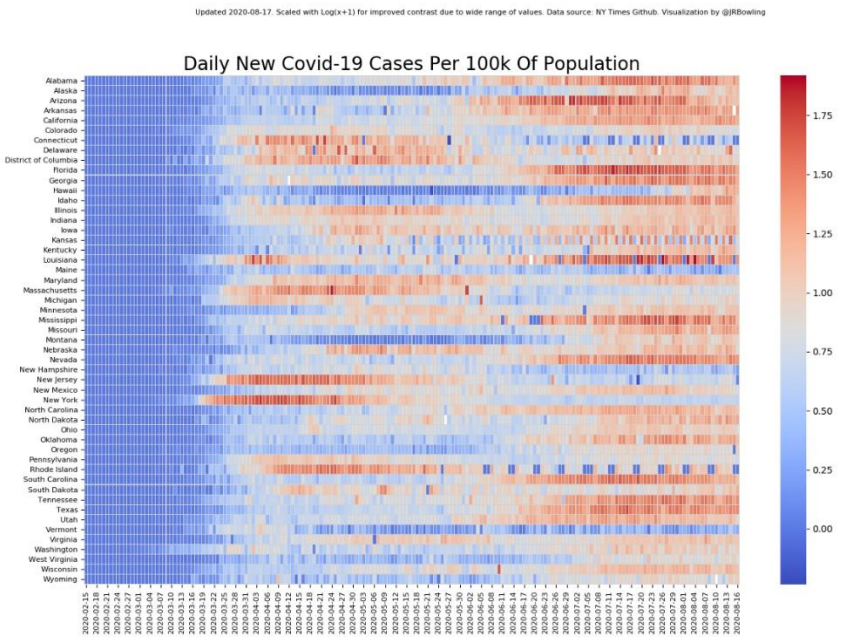
Reading Data

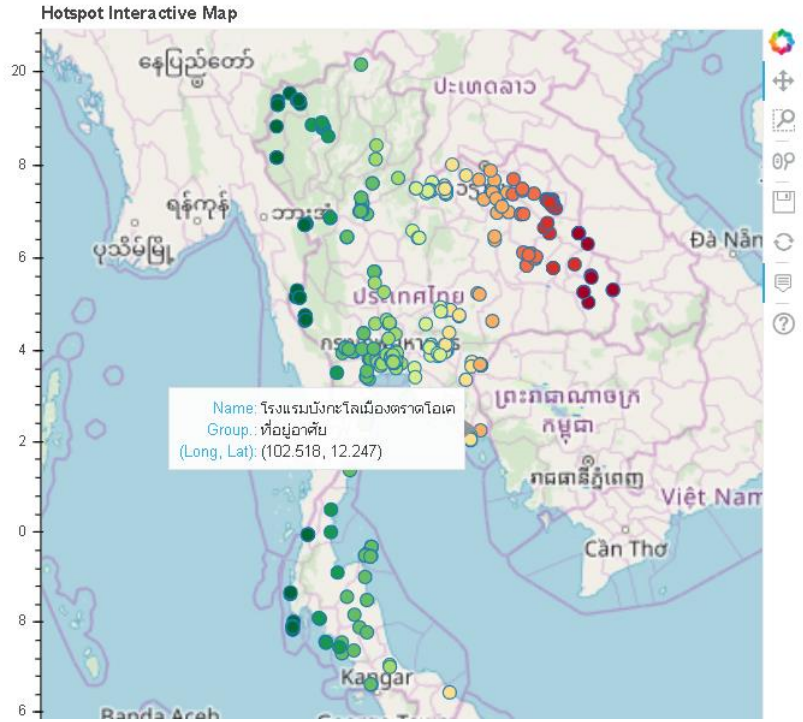
- ตัว Object ที่ส่งกลับมาเรียกว่า Response Object โดยสามารถนำมาใช้งานตามตัวอย่าง

```
import requests
import json

url = 'https://covid19.ddc.moph.go.th/api/Cases/today-cases-all'

req = requests.get(url)
data = req.json()

print("ผู้ป่วยรายวัน:",data[0]['new_case'])
print("ผู้ป่วยสะสม:",data[0]['total_case'])
print("ผู้เสียชีวิต:",data[0]['new_death'])
print("หายป่วย:",data[0]['new_recovered'])
```

Visualization

- นอกเหนือจากที่กล่าวมา Python ยังมีเครื่องมือสำหรับทำ Visualization ที่หลากหลายมาก
- เช่น แสดงเป็น Map, Heatmap





Assignment

- เป็นงานรายบุคคล
- คะแนน 15% ของส่วน Lecture และ 20% ของส่วน Lab
- ให้เขียนโปรแกรมวิเคราะห์ข้อมูลเบื้องต้น
  - ต้องใช้ GUI เป็น Interface
  - ต้องมีการอ่านข้อมูลจาก Internet โดยอาจเป็นรูปแบบ CSV หรือ Excel หรือ JSON หรือ API โดยหากเป็นข้อมูลในประเทศไทยจะดีมาก
  - ต้องมีการแสดงผลเป็นกราฟ หรือ Visualization
  - ต้องมีส่วนการวิเคราะห์หรือเปรียบเทียบข้อมูล ระหว่างชุดข้อมูล เช่น เปรียบเทียบข้อมูลระหว่างปี เดือน

Assignment

- ให้เขียนโปรแกรมวิเคราะห์ข้อมูลเบื้องต้น
  - ต้องสามารถเพิ่มข้อมูลแบบเดียวกัน ในอนาคตได้ เช่น หากมีข้อมูลปี 65, 66 ก็สามารถเพิ่มได้ เป็นต้น
  - ควรสามารถเลือกการแสดงผลได้หลายรูปแบบ เช่น มีตัวเลือกสำหรับเลือกเปรียบเทียบข้อมูลแบบต่างๆ เพื่อให้โปรแกรมมีความหลากหลายในการใช้งาน

Assignment

- การส่งงาน
  - ส่งรายงาน Proposal ประกอบด้วย
    - ข้อเสนอข้อมูลที่จะนำมาใช้งาน พร้อมทั้งแหล่งข้อมูล
    - รูปแบบการวิเคราะห์จะแสดงอะไรบ้าง
    - หน้าจอ UI คร่าวๆ
    - กำหนดส่ง 9 พ.ย. 65

Assignment

- การส่งงานฉบับสมบูรณ์
  - ส่ง Source Code ที่ใช้งานได้ โดยให้ส่งมาใน Folder ที่ขึ้นต้นด้วยรหัส นศ.
  - รายละเอียดของชิ้นงาน
    - โครงสร้างข้อมูลที่นำมาใช้งานในโครงงาน
    - ข้อเสนอข้อมูลที่จะนำมาใช้งาน พร้อมทั้งแหล่งข้อมูล
    - รูปแบบการวิเคราะห์
    - หน้าจอ UI และการใช้งานคร่าวๆ
    - โครงสร้างโปรแกรม การแบ่งเป็นฟังก์ชันอะไรบ้าง แต่ละฟังก์ชันทำอะไร
    - กำหนดส่ง 23 พ.ย. 65

For your attention