

Bài 10: Giới thiệu Bokeh

Giới thiệu

Bokeh là một thư viện trong Python giúp trực quan hóa dữ liệu có tương tác (interactive data visualization). Đặc biệt Bokeh giúp bạn mang các biểu đồ lên web một cách dễ dàng.

Vài khái niệm cơ bản

Document	Bokeh xem biểu đồ chỉ là một phần của trang web cần trình bày. Vì thế Bokeh tổ chức biểu đồ và nhiều thông tin khác nữa để xuất kết quả, thông dụng nhất thì kết quả là một website, dưới dạng Bokeh Document (tài liệu)
Application	Đây là phần mã nguồn đảm nhận phần nghiệp vụ vẽ biểu đồ và xử lý các sự kiện tương tác (như người dùng thay đổi điều kiện dữ liệu, dữ liệu nguồn có thay đổi, v.v...) để kết ra Bokeh Document.
Server	Gồm 3 thành phần Bokeh Document và Application ở trên để cung cấp dịch vụ hiển thị biểu đồ cho người dùng qua môi trường mạng.
Figure	Là đối tượng được ví như là một bức tranh mà bạn là họa sĩ sẽ vẽ biểu đồ và trang trí lên bức tranh.

Đọc thêm tài liệu tại:

https://docs.bokeh.org/en/latest/docs/user_guide/concepts.html
<https://docs.bokeh.org/en/latest/docs/reference/server.html>

Cài đặt

Tài liệu này dùng bokeh phiên bản 2.2.3 với lệnh cài đặt như sau:

```
pip install bokeh==2.2.3
```

Sử dụng bokeh plotting đơn giản

Quay lại dữ liệu mẫu GDP của Việt Nam, sử dụng thư viện bokeh để vẽ biểu đồ line như sau:

```
from bokeh.plotting import figure, output_file, show

output_file("output.html")

gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
```

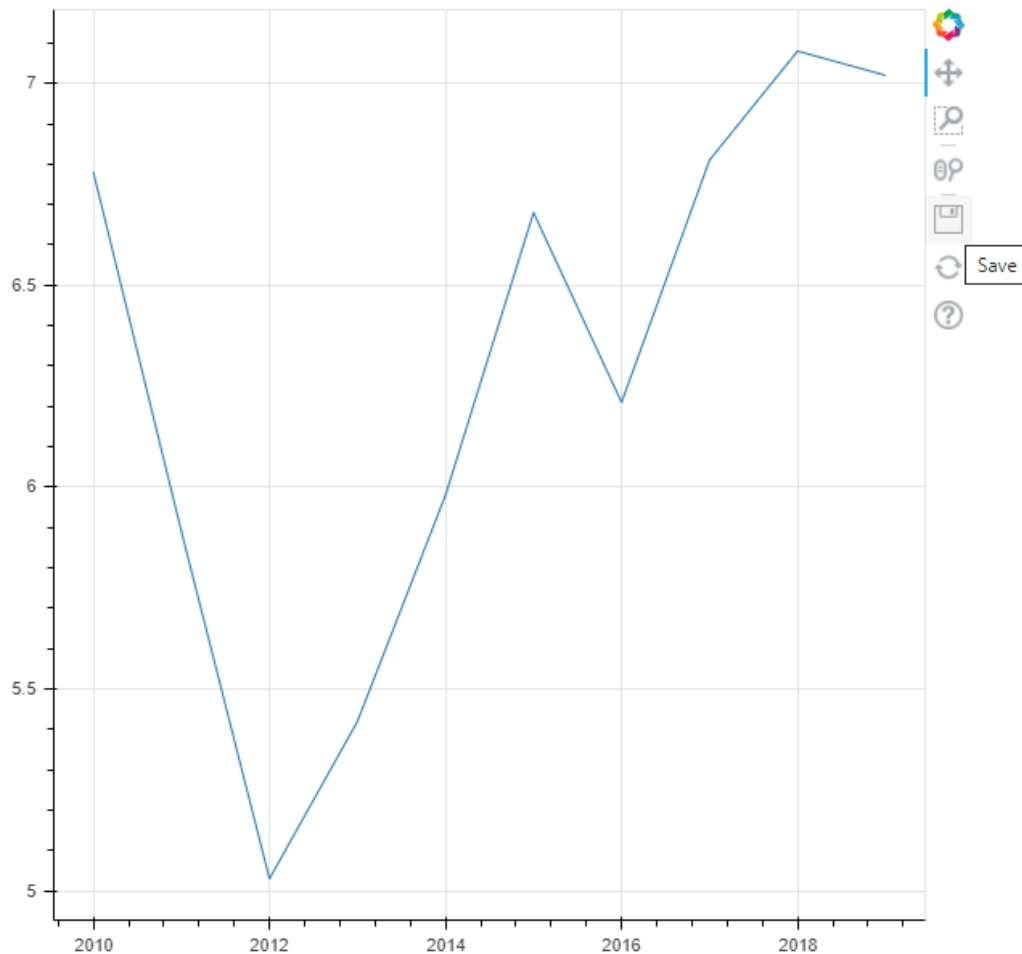
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]

p = figure()
p.line(x=year, y=gdp)

show(p)
```

Kết quả:



Bạn để ý thấy bên phải có thanh công cụ. Trong đó có biểu tượng cái đĩa (disk) cho phép bạn lưu (Save) cái hình của biểu đồ thành file ảnh.

Phân tích mã nguồn

Dòng đầu tiên để import các hàm sau trong thư viện `bokeh.plotting`:

- Hàm `output_file(file_path)` để thiết lập biểu đồ sẽ được ghi thành file trong đường dẫn `file_path`. Trường hợp không có lệnh này thì bokeh sẽ tự tạo ra file tạm.

- Hàm `figure()` để tạo ra đối tượng để vẽ biểu đồ. Hãy tưởng tượng giống như một họa sĩ sẽ căng khung vải để vẽ bức tranh. Đối tượng được lưu trong biến `p` (viết tắt của `picture` hoặc `plot`).

Nếu không truyền tham số gì hết thì kích thước của biểu đồ mặc định là 600 x 600 pixel. Muốn chỉ định rõ kích thước thì dùng thêm tham số `plot_width` và `plot_height` như sau:

```
figure(plot_width=300, plot_height=300)
```

- Hàm `p.line(x= array, y = array)` để vẽ một đường thẳng lên `picture` với thông số trục `x` và trục `y` như chỉ định.
- Hàm `show(p)` sẽ gọi trình duyệt mở file trong hàm `output_file(file_path)` ở trên để bạn xem.

Đưa biểu đồ lên Web

Sử dụng Bokeh Server bên ngoài

Thay vì trong ví dụ trên dùng hàm `output_file(...)` để lưu biểu đồ ra file HTML, chúng ta thử sửa dùng hàm `curdoc()` để thêm “bức tranh” vào tài liệu.

Hãy lưu đoạn code sau vào file. Ví dụ file: `D:\MyPython\bokeh\bokeh_gpd.py`

```
from bokeh.plotting import figure, curdoc

gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]

p = figure()
p.line(x=year, y=gdp)

curdoc().add_root(p)
```

Đứng trong thư mục `D:\MyPython\bokeh` của cửa sổ lệnh có Python và đã cài thư viện Bokeh thực hiện lệnh sau:

```
bokeh serve --show bokeh_gpd.py
```

Hoặc có thể gõ đường dẫn đầy đủ của file `.py` như sau:

```
bokeh serve --show D:\MyPython\bokeh\bokeh_gpd.py
```

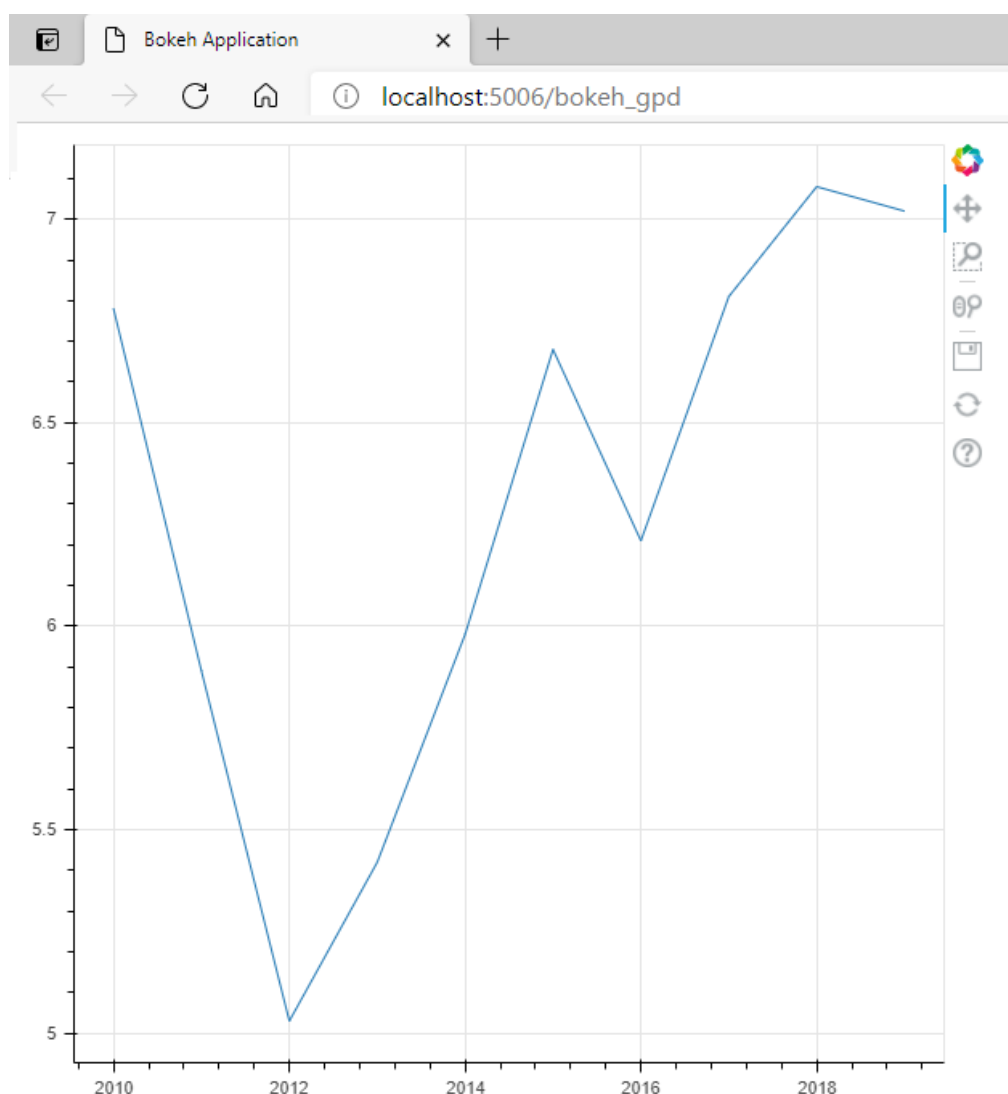
Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

Cửa sổ lệnh sẽ hiển thị các kết quả sau cho thấy là lệnh bokeh sẽ mở một Bokeh server với port 5006 và phục vụ biểu đồ ở trên tại đường link:

`http://localhost:5006/bokeh_gpd`

```
Anaconda Prompt (anaconda3) - bokeh serve --show D:\MyPython\bokeh\bokeh_gpd.py
(base) D:\MyPython\bokeh>bokeh serve --show D:\MyPython\bokeh\bokeh_gpd.py
2021-04-10 08:53:05,370 Starting Bokeh server version 2.2.3 (running on Tornado 6.0.4)
2021-04-10 08:53:05,375 User authentication hooks NOT provided (default user enabled)
2021-04-10 08:53:05,382 Bokeh app running at: http://localhost:5006/bokeh_gpd
2021-04-10 08:53:05,383 Starting Bokeh server with process id: 12428
2021-04-10 08:53:06,652 WebSocket connection opened
2021-04-10 08:53:06,652 ServerConnection created
```

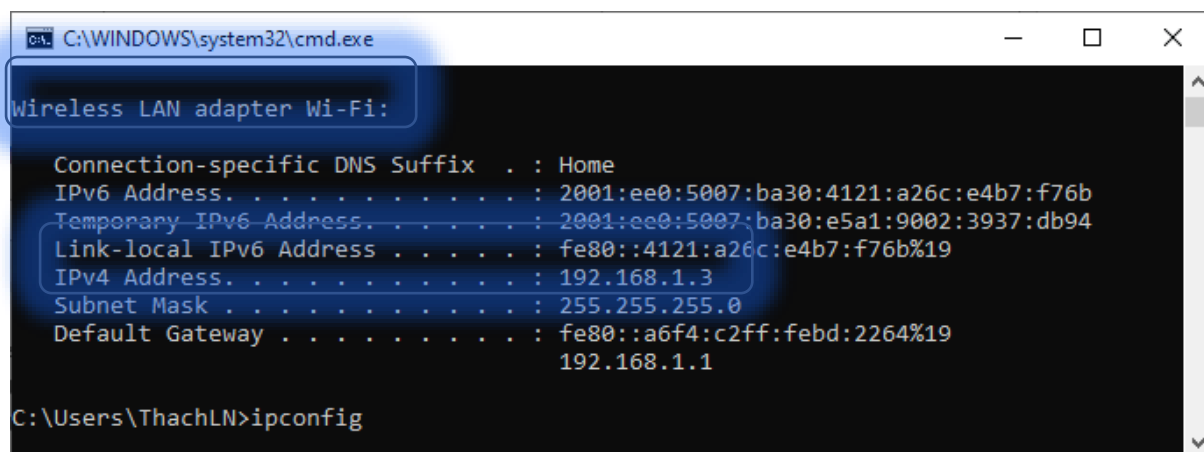
Đồng thời trình duyệt trên máy bạn sẽ tự mở được link ở trên với kết quả như sau:



Nếu bạn muốn bạn bè, đồng nghiệp của mình đang dùng chung Network (ví dụ đang cùng vào chung một WiFi) thì chạy lại lệnh bokeh với tham số cho phép truy cập mạng như sau:

`bokeh serve --show bokeh_gpd.py --allow-websocket-origin=192.168.1.3:5006`

Tham số `--allow-websocket-origin=192.168.1.3:5006` có nghĩa là cho phép máy tính của bạn có địa chỉ IP là 192.168.1.3 chạy chương trình bokeh với cổng 5006. Để kiểm tra chính xác IP mạng của máy bạn thì dùng lệnh `ipconfig`. Sau đó tìm đoạn có chữ “Wireless LAN adapter Wi-Fi” và mục IPv4 Address. Ví dụ:



```
C:\WINDOWS\system32\cmd.exe
Wireless LAN adapter Wi-Fi:
    Connection-specific DNS Suffix . : Home
    IPv6 Address. . . . . : 2001:ee0:5007:ba30:4121:a26c:e4b7:f76b
    Temporary IPv6 Address. . . . . : 2001:ee0:5007:ba30:e5a1:9002:3937:db94
    Link-local IPv6 Address . . . . . : fe80::4121:a26c:e4b7:f76b%19
    IPv4 Address. . . . . : 192.168.1.3
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::a6f4:c2ff:febd:2264%19
                               192.168.1.1

C:\Users\ThachLN>ipconfig
```

Lúc này có thể mời bạn bè, đồng nghiệp vào địa chỉ sau để xem thành quả của bạn:

http://192.168.1.3:5006/bokeh_gpd

Để tắt bokeh server thì đứng trong cửa sổ lệnh đang chạy, nhấn phím Ctrl+C và đợi vài giây, bokeh server sẽ ngưng với thông báo sau:

Interrupted, shutting down

Như vậy bạn đã biết cách vẽ một biểu đồ đơn giản với thư viện Bokeh và Bokeh server để có thể truy cập biểu đồ từ mạng. Tuy nhiên theo như cách chạy lệnh Bokeh ở trên thì có chút bất tiện là mỗi lần code xong phải chạy lại lệnh. Phần tiếp theo sẽ hướng dẫn bạn khởi chạy server bằng code Python luôn.

Sử dụng Bokeh Server bên trong

Chạy thử ví dụ sau:

```
from bokeh.plotting import figure
from bokeh.server.server import Server
from bokeh.application import Application
from bokeh.application.handlers.function import FunctionHandler

def make_document(doc):
    gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
    year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]
```

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
p = figure()
p.line(x=year, y=gdp)

doc.add_root(p)

doc.title = "Sample chart"
doc.add_root(p)

apps = {'/': Application(FunctionHandler(make_document))}

server = Server(apps, port=5000)
server.start()
```

Sau đó dùng trình duyệt mở địa chỉ sau bạn sẽ thấy kết quả tương tự như phần trước:

<http://localhost:5000>

Cấu trúc mã nguồn:

Đề ý 2 lệnh cuối cùng:

```
server = Server(apps, port=5000)
server.start()
```

để khởi động Bokeh server với port 5000. Trong đó Bokeh server cần một tham số là Application (xem lại phần khái niệm cơ bản ở trên). Tên đầy đủ của Application ở đây là “Bokeh Server Tornado application”. Tra cứu tài liệu tại:

<https://docs.bokeh.org/en/latest/docs/reference/server/tornado.html#bokeh.server.tornado.BokehTornado>

bạn sẽ thấy cách khai báo Application này là dạng Dictionary như:

```
{ '/' : applications }
```

Cụ thể trong ví dụ trên thì khai báo Application như sau:

```
apps = {'/': Application(FunctionHandler(make_document))}
```

Dấu ‘/’ có nghĩa là khi người dùng gõ đường dẫn URL trên trình duyệt như <http://localhost:5000/> (nếu không có dấu xuyệt / cuối thì coi như là có) thì Máy tính

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

(trong ngữ cảnh này gọi là máy chủ nhân, chính là cái máy tính mà bạn đang thực hành) sẽ nhận và xử lý URL gồm 2 phần:

- Phần “http://localhost:5000”: sẽ được máy có tên là localhost (tức là máy của bạn đang thực hành) có một phần mềm đang chạy ở port 5000 (trong trường này chính là Bokeh Server trong ví dụ) nhận và xử lý yêu cầu.
- Phần “/” (dấu duyệt phải ở cuối): gọi là yêu cầu (request). Dấu / gọi là root. Có thể URL đầy đủ không có dấu duyệt thì root là rỗng. Khi gặp yêu cầu root này thì đối tượng **applications** sẽ xử lý. Cụ thể applications trong ví dụ trên là:

```
Application(FunctionHandler(make_document))
```

Đây là lệnh khởi tạo đối tượng Application⁶ với tham số là đối tượng Handler⁷.
Cụ thể là:

```
FunctionHandler(make_document)
```

Lệnh này sử dụng đối tượng FuncHandler để chỉ định hàm lo xử lý việc tạo ra bokeh document. Bạn xem cách khai báo hàm make_document với tham số là doc như sau:

```
def make_document(doc):  
    p = figure()  
    ...  
    doc.add_root(p)  
  
    ...  
    doc.add_root(p)
```

Hàm make_document cập nhật tham số doc bằng cách gọi hàm add_root(p) với p là đối tượng Figure.

Stop server

Thực hiện lệnh Python sau:

⁶ Tra cứu tại:

<https://docs.bokeh.org/en/latest/docs/reference/application/application.html#bokeh.application.application.Application>

⁷

<https://docs.bokeh.org/en/latest/docs/reference/application/handlers/handler.html#bokeh.application.handlers.handler.Handler>

```
server.stop()
```

Mã nguồn đầy đủ có sẵn lệnh `server.stop()` được chú thích.

```
from bokeh.plotting import figure
from bokeh.server.server import Server
from bokeh.application import Application
from bokeh.application.handlers.function import FunctionHandler

def make_document(doc):
    gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
    year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]

    p = figure()
    p.line(x=year, y=gdp)

    doc.add_root(p)

    doc.title = "Sample chart"
    doc.add_root(p)

apps = {'/': Application(FunctionHandler(make_document))}

print('Start bokeh server with port 5000...')
server = Server(apps, port=5000)
server.start()
print('Try the application with URL: http://localhost:5000')

# server.stop()
```

Nếu bạn chạy đoạn code trên 2 lần thì sẽ bị báo lỗi như sau:

```
OSError: [WinError 10048] only one usage of each socket address (protocol/  
network address/port) is normally permitted
```

Lý do là cái port 5000 đã được sử dụng và boked server đang chạy.

Muốn tắt server thì bôi lệnh đoạn `server.stop()` và thực thi nó. Nếu dùng IDE Spyder thì bôi và nhấn F9.

Bài 11: Khai phá Bokeh

Bài trước đã giúp bạn làm quen với vài khái niệm cốt lõi và biết tổng thể về việc dùng bokeh để vẽ một biểu đồ. Đồng thời bạn cũng biết xuất biểu đồ ra file HTML hoặc đưa lên web. Bài này sẽ giúp bạn sử dụng Bokeh để phục vụ cho các nhu cầu về phân tích dữ liệu, đặc biệt khai phá sức mạnh của Bokeh để trực quan hóa dữ liệu nhằm cảm nhận dữ liệu nhanh và sâu sắc hơn. Đây là tiền đề để giúp một người phân tích dữ liệu có thể đưa ra các ý tưởng mới, hoặc tí ra khám phá được thêm các thông tin ẩn đằng sau các con số.

Xem phân bố dữ liệu với biểu đồ histogram

Xem dữ liệu mẫu:

```
import pandas as pd

df = pd.read_csv('https://thachln.github.io/datasets/bank/bank-
additional-full.csv', delimiter=';')

df_desc = df.describe()
```

df_desc - DataFrame										
	age	duration	campaign	pdays	previous	emp.var.rate	ins.price.idx	ons.conf.idx	euribor3m	r.employe
count	41188	41188	41188	41188	41188	41188	41188	41188	41188	41188
mean	40.0241	258.285	2.56759	962.475	0.172963	0.0818855	93.5757	-40.5026	3.62129	5167.04
std	10.4212	259.279	2.77001	186.911	0.494901	1.57096	0.57884	4.6282	1.73445	72.2515
min	17	0	1	0	0	-3.4	92.201	-50.8	0.634	4963.6
25%	32	102	1	999	0	-1.8	93.075	-42.7	1.344	5099.1
50%	38	180	2	999	0	1.1	93.749	-41.8	4.857	5191
75%	47	319	3	999	0	1.4	93.994	-36.4	4.961	5228.1
max	98	4918	56	999	7	1.4	94.767	-26.9	5.045	5228.1

Xem cột mô tả cột age bạn cũng hình dung phần nào độ tuổi của người tham gia chiến dịch marketing của ngân hàng trên. Ví dụ, vài thông tin có thể mô tả ra đây:

- Tổng cộng có 41.188 người tham gia.
- Độ tuổi trung bình là: 40
- Tuổi thấp nhất là 17; tuổi cao nhất là 98
- 25% số người đó độ tuổi khoảng 32
- 50% số người có độ tuổi khoảng 38
- 75% số người có độ tuổi khoảng 47

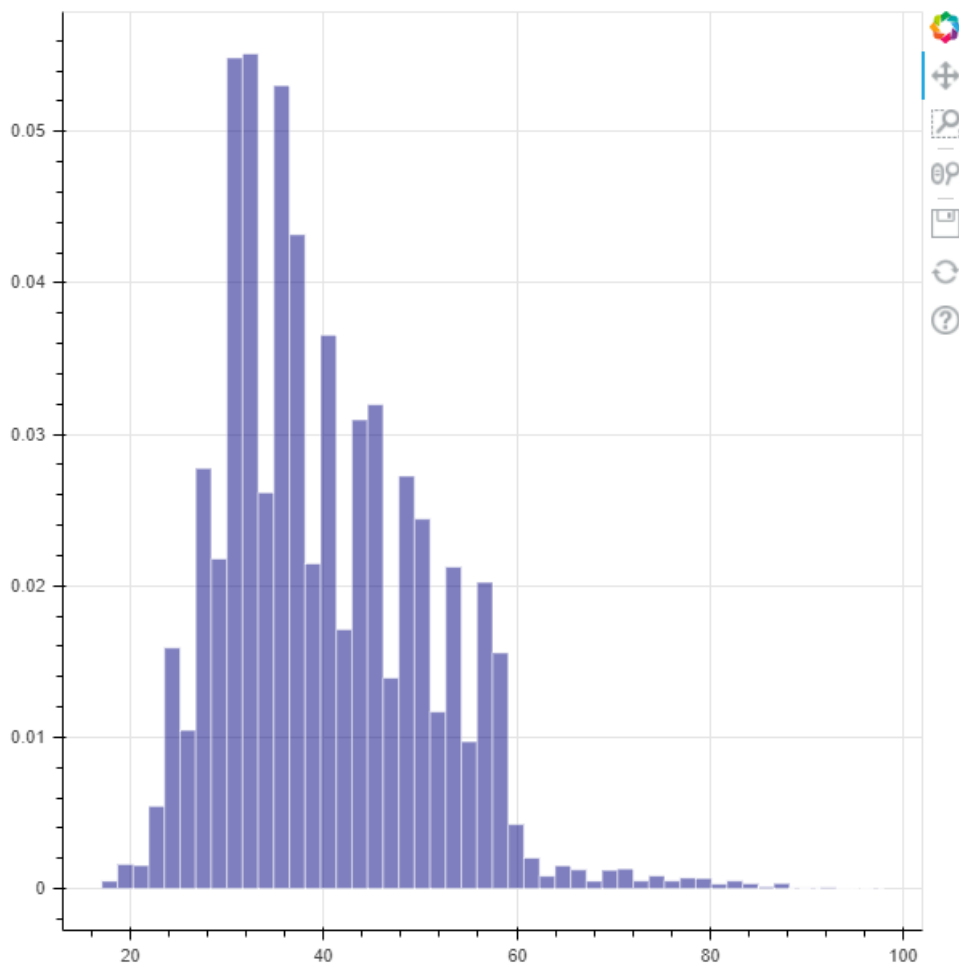
Nếu muốn xem biểu đồ phân bố tuổi của dữ liệu trên thì dùng đoạn code đầy đủ sau:

```
import numpy as np
import pandas as pd
from bokeh.plotting import figure, output_file, show

df = pd.read_csv('https://thachln.github.io/datasets/bank/bank-
additional-full.csv', delimiter=';')
df_desc = df.describe()

hist, edges = np.histogram(df['age'], density=True, bins=50)

output_file("output.html")
p = figure()
p.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:],
        fill_color="navy", line_color="white", alpha=0.5)
show(p)
```



Diễn giải:

- Đoạn code trên sử dụng hàm `numpy.histogram(...)` để tính toán mật độ phân bố dữ liệu tuổi. Tham số `bins = 50` cho biết độ mịn của các thanh đứng trong biểu đồ. Số này càng lớn thì số nhóm tuổi được thống kê sẽ càng lớn, tức là biểu đồ càng mịn.

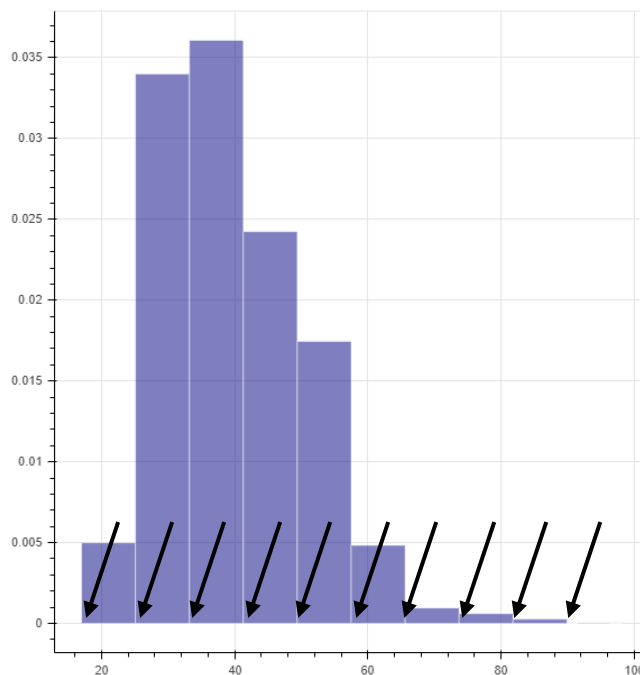
Tham số `bins` này có thể dùng chữ để chỉ phương pháp tính phân bố. Tra thêm tài liệu tại:

https://numpy.org/doc/stable/reference/generated/numpy.histogram_bin_edges.html#numpy.histogram_bin_edges.

Ví dụ sử dụng 'auto' như lệnh sau:

```
hist, edges = np.histogram(df['age'], density=True, bins='auto')
```

Kết quả `edges` chứa danh sách các giá trị trên trục x tại các vị trí biên của các thanh chữ nhật. Xem chỗ mũi tên trong hình bên dưới:



hist chứa danh sách các giá trị độ cao của các thanh chữ nhật. Thanh chữ nhật biểu diễn nhóm tuổi cần thống kê. Số thanh chữ nhật chính là số `bins` (nếu nó là số); hoặc do thuật toán xác định tự động nên `bins` là chữ (tên phương pháp cần áp dụng)

- Hàm để vẽ biểu đồ `figure.quad(...)`

```
p.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:],
        fill_color="navy", line_color="white", alpha=0.5)
```

với ý nghĩa vài tham số như sau:

- top, bottom, left, right: tương ứng với số liệu cần để vẽ biểu đồ; độ cao, điểm đáy, danh sách cạnh trái của thanh chữ nhật, danh sách cạnh phải của thanh chữ nhật.
- fill_color: màu để phủ thanh chữ nhật.
- line_color: màu để kẻ đường bao thanh chữ nhật.
- alpha: độ trong suốt (transparent) của hình vẽ)

Tham khảo thêm tài liệu tại:

<https://docs.bokeh.org/en/latest/docs/reference/plotting.html?highlight=quad#bokeh.plotting.Figure.quad>

Xem phân bố dữ liệu bằng biểu đồ Boxplot

Trong thư viện bokeh tự vẽ boxplot cho cột age của dữ liệu mẫu như sau:

```
import pandas as pd
from bokeh.plotting import figure, output_file, show

df = pd.read_csv('https://thachln.github.io/datasets/bank/bank-
additional-full.csv', delimiter=';')

output_file("output.html")

s = df['age']

q1 = s.quantile(q=0.25)
q2 = s.quantile(q=0.5)
q3 = s.quantile(q=0.75)

# Interquartile range
iqr = q3 - q1
upper = q3 + 1.5*iqr
lower = q1 - 1.5*iqr

out = s[(s > upper) | (s < lower)]

cats = ['age']
p = figure(x_range=cats)

# stems
```

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

```
p.segment(cats, upper, cats, q3, line_color="black")
p.segment(cats, lower, cats, q1, line_color="black")

# boxes
p.vbar(cats, 0.7, q2, q3, fill_color="#E08E79", line_color="black")
p.vbar(cats, 0.7, q1, q2, fill_color="#3B8686", line_color="black")

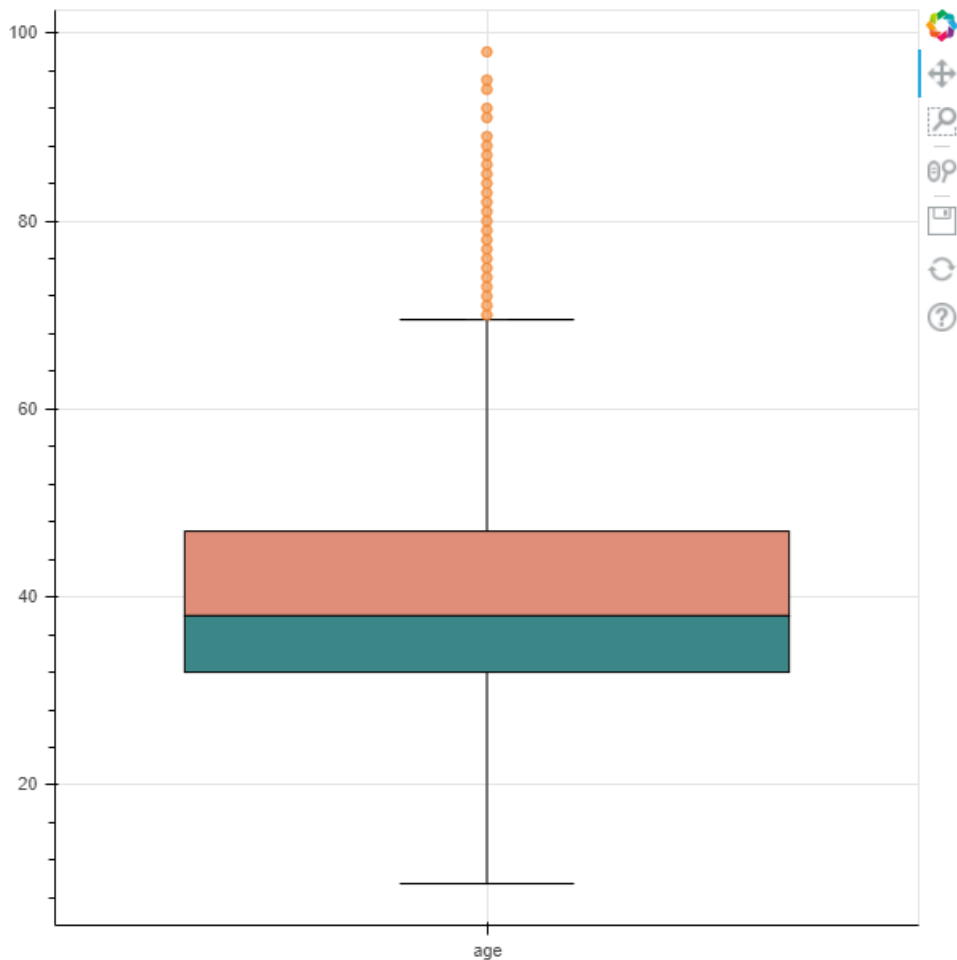
# whiskers (almost-0 height rects simpler than segments)
p.rect(cats, lower, 0.2, 0.01, line_color="black")
p.rect(cats, upper, 0.2, 0.01, line_color="black")

# prepare outlier data for plotting, we need coordinates for every
outlier.
if not out.empty:
    outy = pd.Series(out)
    outy = outy.unique()
    outx = [cats[0] for y in outy]
    # outliers
    p.circle(outx, outy, size=6, color="#F38630", fill_alpha=0.6)

# show the results
show(p)
```

Phần diễn giải mã nguồn sẽ được giải thích sau.

Tạm thời bạn xem kết quả của việc tự vẽ các thành phần của một boxplot cho cột age từ đoạn code trên như sau:



Biểu đồ so sánh 1 biến dùng Line Chart

```
import pandas as pd
from bokeh.plotting import figure, output_file, show

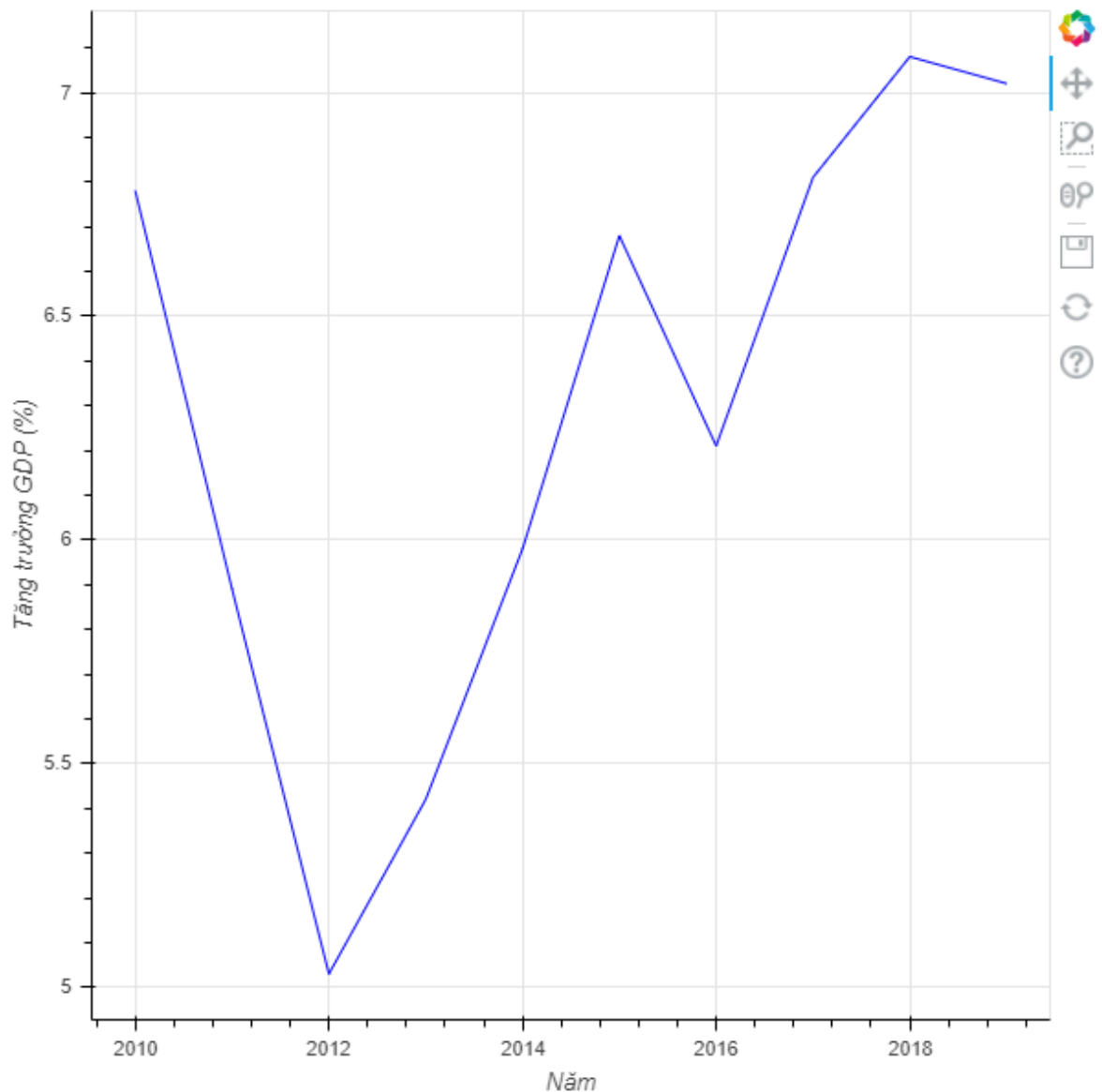
gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]

# Tạo data frame
df = pd.DataFrame({'gdp': gdp, 'year': year})

output_file("output.html")
p = figure()

p.xaxis.axis_label = 'Năm'
p.yaxis.axis_label = 'Tăng trưởng GDP (%)'
```

```
p.line(  
    x = df['year'],  
    y = df['gdp'],  
    color = 'blue'  
)  
  
show(p)
```



Biểu đồ so sánh 2 biến dùng Line Chart

```
import pandas as pd  
from bokeh.plotting import figure, output_file, show  
  
gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
```

```
year = [2010,2011,2012,2013,2014,2015,2016,2017,2018,2019]
cpi = [9.19,18.58, 9.21, 6.60, 4.09, 0.63, 2.66, 3.53, 3.54, 3.89]
df = pd.DataFrame({'gdp': gdp, 'year': year, 'cpi': cpi})

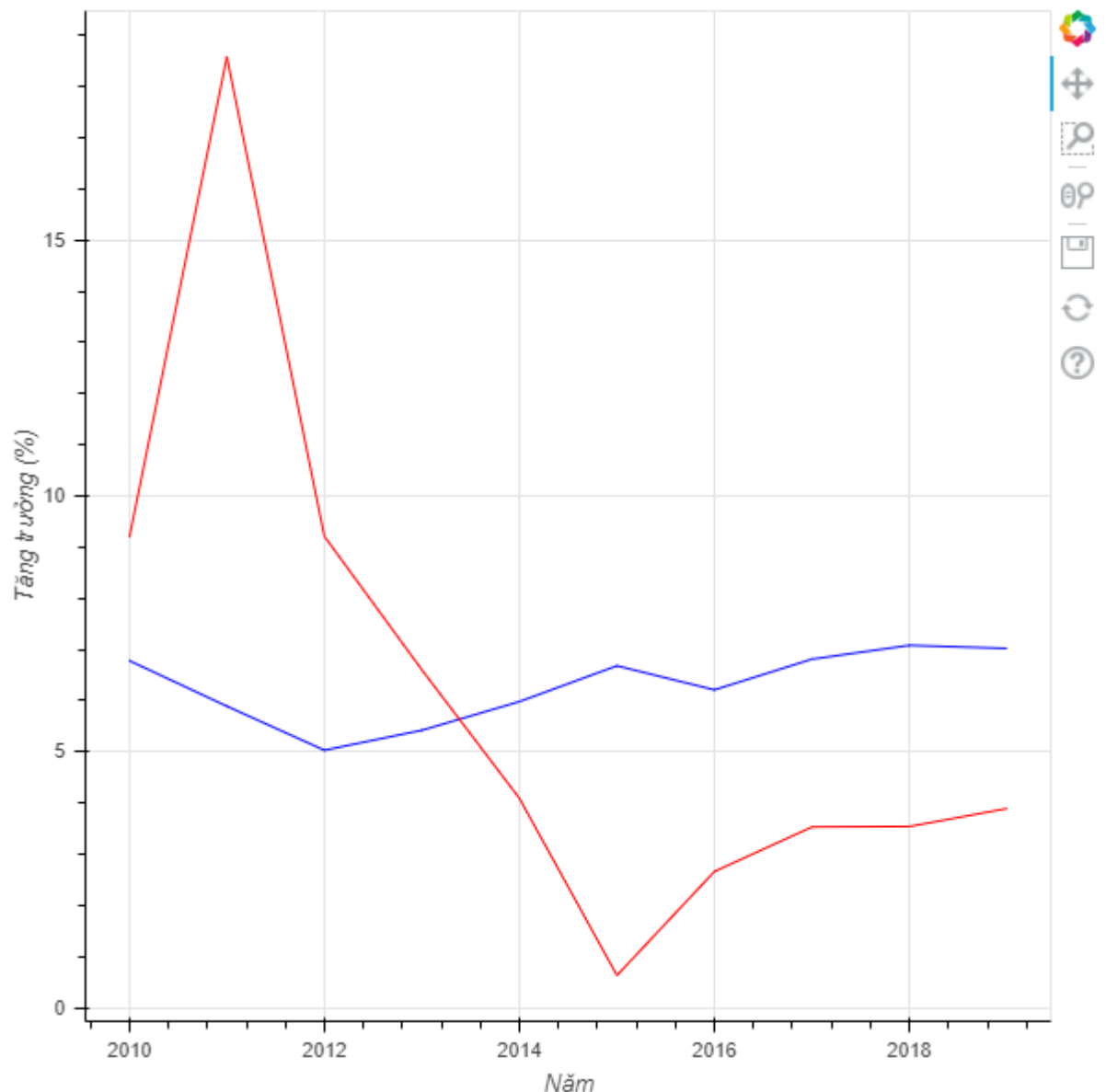
output_file("output.html")
p = figure()

p.xaxis.axis_label = 'Năm'
p.yaxis.axis_label = 'Tăng trưởng (%)'

p.line(
    x = df['year'],
    y = df['gdp'],
    color = 'blue'
)

p.line(
    x = df['year'],
    y = df['cpi'],
    color = 'red'
)

show(p)
```

Biểu đồ so sánh nhiều biến dùng Line Chart

Bokeh cung cấp hàm `multi_line` để plot dữ liệu từ các bộ dữ liệu tương ứng cho trục x và y. Trong ví dụ bên dưới hàm `multi_line` nhận tham số `xs` và `ys`.

`xs` là một array gồm 2 phần tử `[x1, x2]`. `x1` và `x2` lại mảng các năm.

`ys` là một array tương ứng gồm 2 phần tử `[y1, y2]` với `y1` là mảng các giá trị gdp (tương ứng với từng năm trong `x1`); `y2` là mảng các giá trị cpi (tương ứng với từng năm trong `x2`).

```
from bokeh.plotting import figure, output_file, show

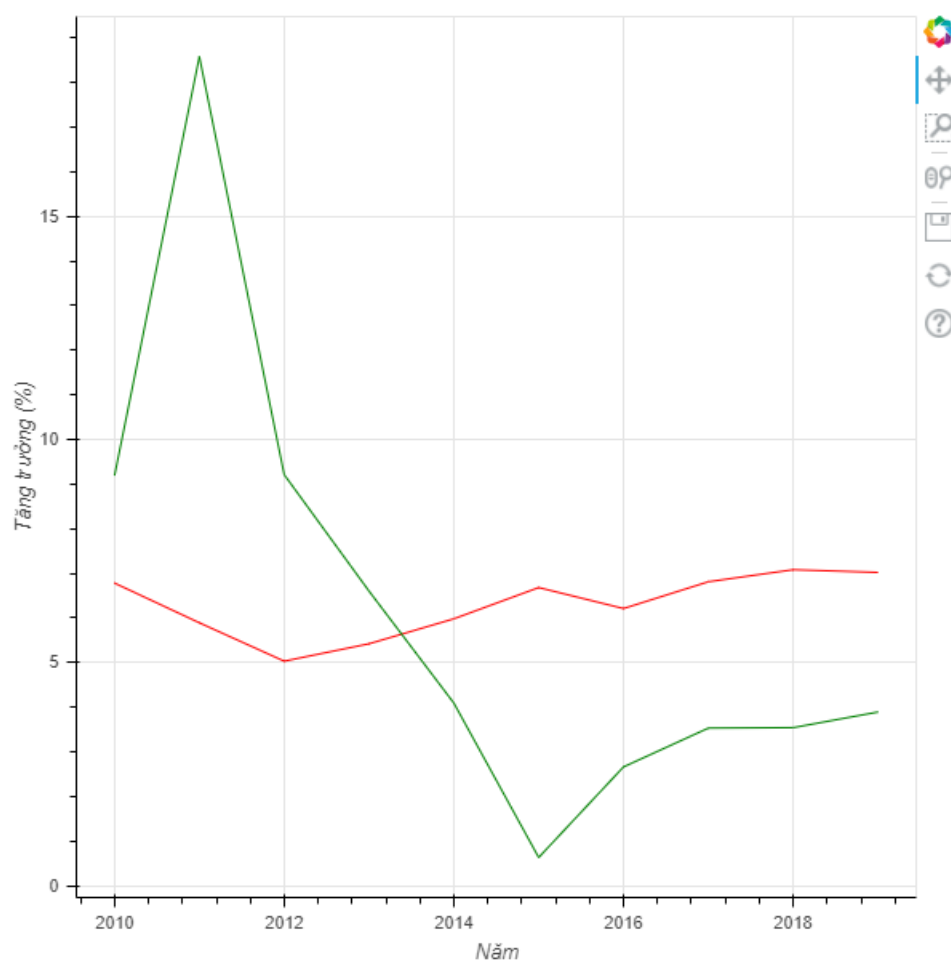
gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
year = [2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]
cpi = [9.19, 18.58, 9.21, 6.60, 4.09, 0.63, 2.66, 3.53, 3.54, 3.89]
```

```
output_file("output.html")
p = figure()

p.xaxis.axis_label = 'Năm'
p.yaxis.axis_label = 'Tăng trưởng (%)'

p.multi_line(xs=[year, year], ys=[gdp, cpi], color=['red', 'green'])

show(p)
```



Biểu đồ so sánh nhiều biến dùng Line Chart với Data Source

Trên cơ sở ý tưởng dùng các cặp giá trị (x,y) tương ứng cho nhiều line khi dùng hàm `multi_line` ở trên thì phần này cải tiến một chút bằng cách dùng Data Source.

Xem và trải nghiệm đoạn code bên dưới:

```
from bokeh.models import ColumnDataSource
from bokeh.plotting import figure, output_file, show
```

```
gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
year = [2010,2011,2012,2013,2014,2015,2016,2017,2018,2019]
cpi = [9.19,18.58, 9.21, 6.60, 4.09, 0.63, 2.66, 3.53, 3.54, 3.89]

output_file("output.html")

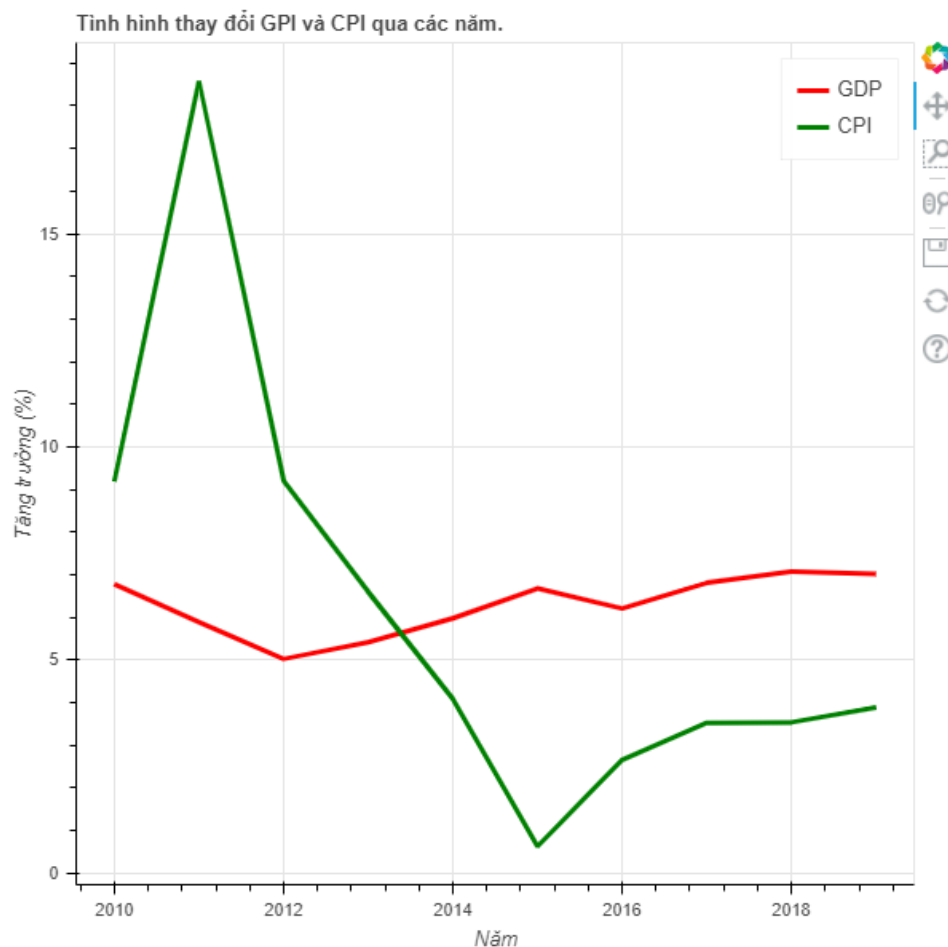
p = figure(title="Tình hình thay đổi GPI và CPI qua các năm.")

p.xaxis.axis_label = 'Năm'
p.yaxis.axis_label = 'Tăng trưởng (%)'

ds = ColumnDataSource(data={'x': [year, year], 'y': [gdp, cpi],
                             'color': ['red', 'green'],
                             'label': ['GDP', 'CPI']})

p.multi_line('x', 'y', color='color', legend='label', line_width=3,
source=ds)

show(p)
```



Giới thiệu Data Source trong Bokeh

Ý tưởng dùng Data Source nhằm mục đích cung cấp một nguồn dữ liệu cho biểu đồ. Khi nguồn dữ liệu thay đổi về nội dung thì biểu đồ sẽ được vẽ lại. Ý tưởng này sẽ giúp tạo ra các biểu đồ có tính tương tác (interactive chart). Đây là một trong các điểm mạnh của thư viện Bokeh.

Quay lại ví dụ trong phần trước, cần phân tích và điểm chính trong mã nguồn:

- Để sử dụng nguồn dữ liệu cho biểu đồ thì khai báo sử dụng lớp `ColumnDataSource` từ module `bokeh.models`:

```
from bokeh.models import ColumnDataSource
```

- Chuẩn bị một biến dạng Dictionary để ánh xạ các dữ liệu vào key như sau:

```
dict_data = {'x': [year, year], 'y': [gdp, cpi],  
            'color': ['red', 'green'],  
            'label': ['GDP', 'CPI']}
```

Phần **màu đỏ** là key, **màu xanh** là values (theo ví dụ). Giá trị theo dạng gì là tùy ý nghĩa của key và dùng vào mục gì trong hàm vẽ biểu đồ. Key là do bạn tự đặt để khai báo tương ứng cho hàm vẽ biểu đồ.

- Khởi tạo đối tượng `ColumnDataSource` với tham số là một trong các dạng dữ liệu sau:

- o Dữ liệu dạng Dictionary đã chuẩn bị ở trên:

```
ds = ColumnDataSource(data=dict_data)
```

- o Dữ liệu là DataFrame:

```
ds = ColumnDataSource(data=df)
```

- o Dữ liệu dạng GroupBy:

```
group = df.groupby(('colA', 'ColB'))  
ds = ColumnDataSource(data=group)
```

- Để vẽ biểu đồ nhiều line với hàm `multi_line` thì tham số `source` được khai báo dùng biến `ds`:

```
p.multi_line('x', 'y', color='color', legend='label',  
line_width=3, source=ds)
```

Trong đó các giá trị **'x'**, **'y'**, **'color'**, **'label'** chính là các key trong dict data đã chuẩn bị ở trên.

Thuộc tính quan trọng của ColumnDataSource

Thuộc tính data

Để truy cập dữ liệu của `ColumnDataSource` thì Bokeh cung cấp thuộc tính `data`. Ví dụ để thay đổi dữ liệu của Data Source thì dùng cú pháp sau:

```
ds.data = <new dict data>
```

Biểu đồ với dữ liệu thay đổi

Để minh họa ý tưởng sử dụng Data Source để vẽ biểu đồ với dữ liệu thay đổi theo thời gian thực thì chúng ta cải tiến ví dụ vẽ biểu đồ GDP và CPI một chút:

```
from bokeh.server.server import Server  
from bokeh.application import Application  
from bokeh.application.handlers.function import FunctionHandler
```

```
from bokeh.plotting import figure, ColumnDataSource

import random

def make_document(doc):
    gdp = [6.78, 5.89, 5.03, 5.42, 5.98, 6.68, 6.21, 6.81, 7.08, 7.02]
    year = [2010,2011,2012,2013,2014,2015,2016,2017,2018,2019]
    cpi = [9.19,18.58, 9.21, 6.60, 4.09, 0.63, 2.66, 3.53, 3.54,
3.89]

    ds = ColumnDataSource(data={'x': [year, year], 'y': [gdp, cpi],
                                'color': ['red', 'green'],
                                'label': ['GDP', 'CPI']})

    def update():
        cpi = [9.19,18.58, 9.21, 6.60, 4.09, random.randint(1, 100),
2.66, 3.53, 3.54, 3.89]
        new_ds = {'x': [year, year], 'y': [gdp, cpi],
                    'color': ['red', 'green'],
                    'label': ['GDP', 'CPI']}

        ds.data = new_ds

    doc.add_periodic_callback(update, 100)

    p = figure(title="Tình hình thay đổi GPI và CPI qua các năm.")

    p.xaxis.axis_label = 'Năm'
    p.yaxis.axis_label = 'Tăng trưởng (%)'

    p.multi_line('x', 'y', color='color', legend='label',
line_width=3, source=ds)

    doc.title = "Now with live updating!"
    doc.add_root(p)
```

```
apps = {'/': Application(FunctionHandler(make_document))}

server = Server(apps, port=5000)
server.start()
print('View dynamic chart at http://localhost:5000')
```

Hãy thực thi đoạn lệnh trên và mở trình duyệt với địa chỉ sau để xem kết quả:

<http://localhost:5000>

Để tắt bokeh server thì thực hiện lệnh sau:

```
server.stop()
```

Vẽ biểu đồ nhiều line bằng hàm `vline_stack`

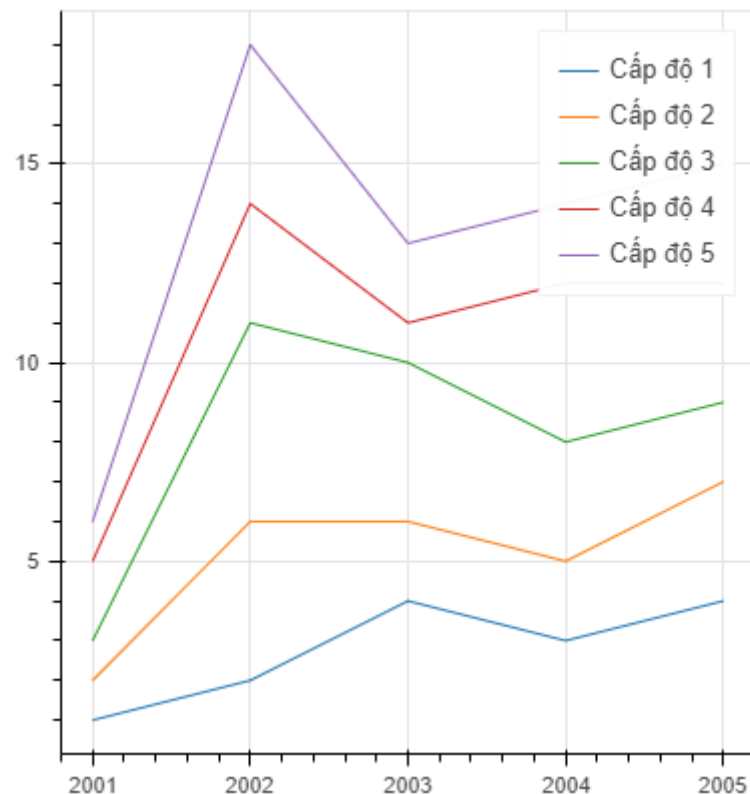
```
from bokeh.models import ColumnDataSource
from bokeh.plotting import figure, show
from bokeh.palettes import Category10_10

source = ColumnDataSource(data=dict(
    x=[2015, 2016, 2017, 2018, 2019],
    y1=[1, 2, 4, 3, 4],
    y2=[1, 4, 2, 2, 3],
    y3=[1, 5, 4, 3, 2],
    y4=[2, 3, 1, 4, 3],
    y5=[1, 4, 2, 2, 3],
))
levels = [1, 2, 3, 4, 5]
labels = ['Cấp độ %s' % x for x in levels]

p = figure(plot_width=400, plot_height=400, title='Lịch sử hoàn thành  
công việc theo cấp độ')

p.vline_stack(['y1', 'y2', 'y3', 'y4', 'y5'], x='x', source=source,
color=Category10_10[0:len(levels)], legend_label=labels)

show(p)
```



Biểu đồ so sánh - Bar Chart

Sử dụng hàm `vbar` của `figure`

Một cách đơn giản là dùng hàm `vbar` để vẽ các thanh đứng (vertical) như ví dụ bên dưới với các thông tin:

- `x` là mảng chỉ định các vị trí của thanh
- `top` là mảng chỉ định độ cao các thanh
- `width`: độ rộng các thanh

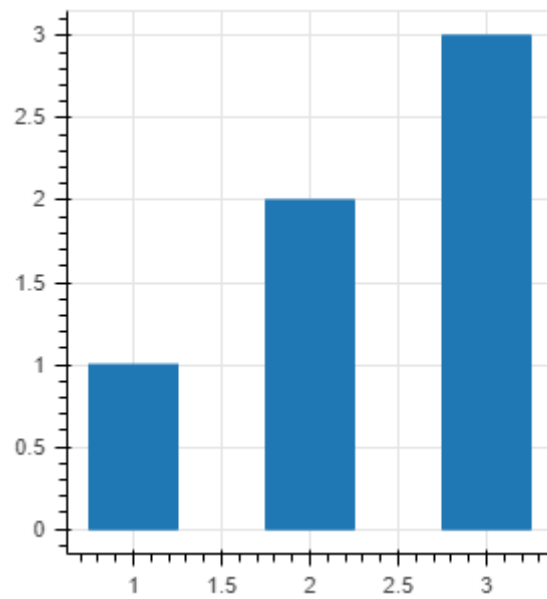
Tra cứu các tham số khác tại:

<https://docs.bokeh.org/en/latest/docs/reference/plotting.html?highlight=vbar#bokeh.plotting.Figure.vbar>

```
from bokeh.plotting import figure, show

plot = figure(plot_width=300, plot_height=300)
plot.vbar(x=[1, 2, 3], width=0.5, top=[1,2,3])

show(plot)
```

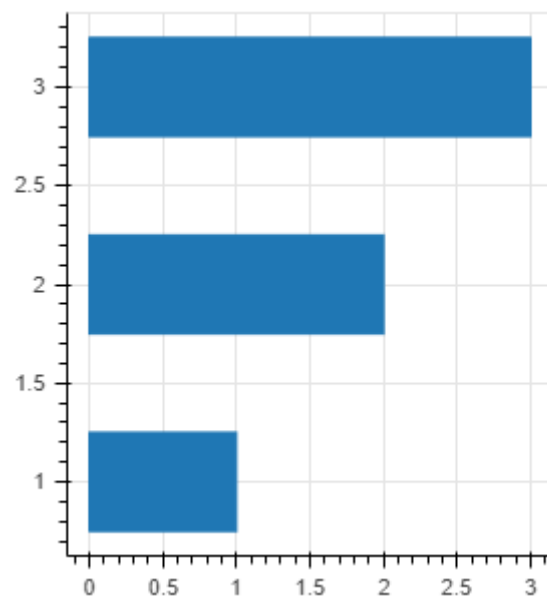



Sử dụng hàm *hbar*

```
from bokeh.plotting import figure, show

plot = figure(plot_width=300, plot_height=300)
plot.hbar(y=[1, 2, 3], height=0.5, right=[1,2,3])

show(plot)
```



Sử dụng bar chart với data source

```
from bokeh.plotting import figure, show
from bokeh.models import ColumnDataSource

# create a Python dict as the basis of your ColumnDataSource
data = {'x_values': [1, 2, 3],
        'y_values': [1, 2, 3]}

# create a ColumnDataSource by passing the dict
source = ColumnDataSource(data=data)

plot = figure(plot_width=300, plot_height=300)
plot.vbar(x='x_values', width=0.5, top='y_values', source=source)

show(plot)
```

Đưa phần Legend ra khỏi biểu đồ

Các biểu đồ trong các phần trước thì phần Legend (các chú giải màu sắc, đường nét trong biểu đồ) nếu có thì nó được đặt bên trong biểu đồ. Điều này đôi lúc nội dung của biểu đồ bị che mất. Để đưa phần Legend ra khỏi biểu đồ thì cần thiết lập layout cho biểu đồ với vị trí Legend cụ thể. Ví dụ đoạn code bên dưới thiết lập Legend được trình bày theo hàng ngang và ở phía trên của biểu đồ.

```
from bokeh.models import Legend
p = figure()
p.add_layout(Legend(orientation = 'horizontal'), 'above')
```

Xem cách sử dụng trong phần tiếp theo.

Sử dụng bar chart nhiều màu

```
from bokeh.io import show
from bokeh.models import ColumnDataSource, Legend
from bokeh.palettes import Category10_10
from bokeh.plotting import figure
from bokeh.transform import factor_cmap

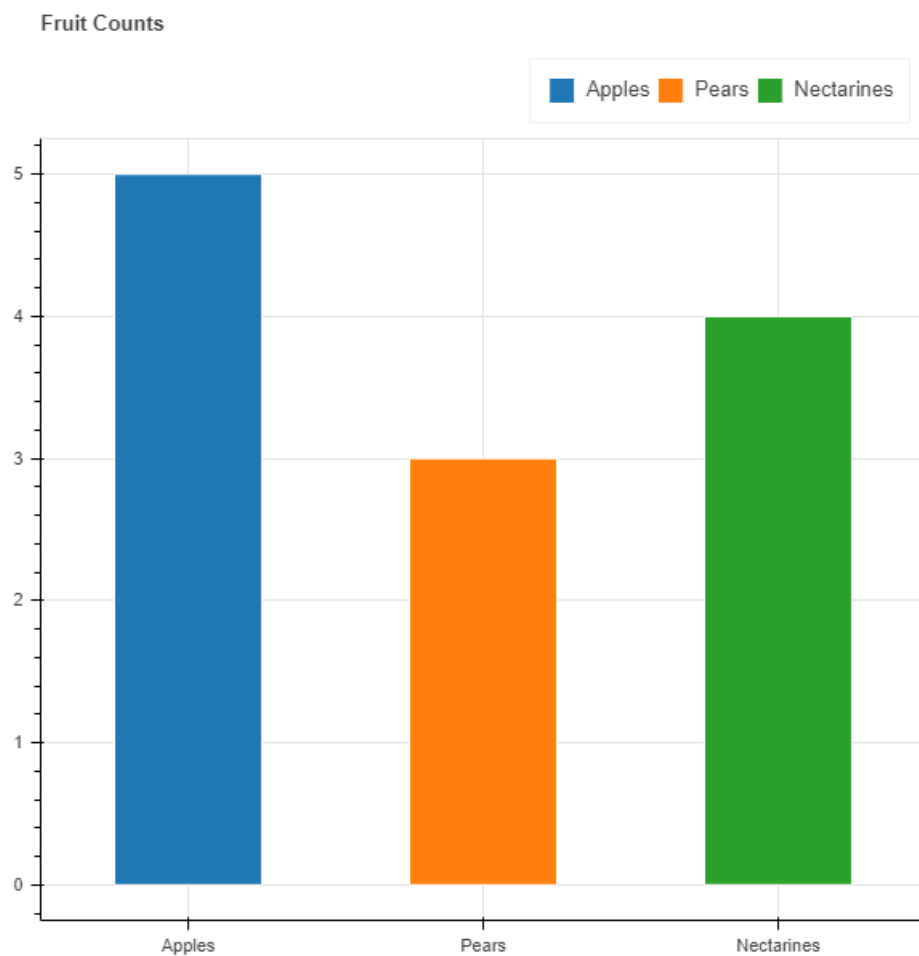
fruits = ['Apples', 'Pears', 'Nectarines']
counts = [5, 3, 4]
```

```
source = ColumnDataSource(data=dict(fruits=fruits, counts=counts))

p = figure(x_range=fruits, title="Fruit Counts")
p.add_layout(Legend(orientation = 'horizontal'), 'above')

colors = factor_cmap('fruits', palette=Category10_10, factors=fruits)
p.vbar(x='fruits', top='counts', width=0.5, source=source,
       legend="fruits",
       line_color='white', fill_color=colors)

show(p)
```



Vẽ bar chart theo nhóm

```
from bokeh.core.properties import value
from bokeh.io import show
from bokeh.models import ColumnDataSource
from bokeh.plotting import figure
from bokeh.transform import dodge
from bokeh.palettes import Category10_10

fruits = ['Apples', 'Pears', 'Nectarines']
years = ['2015', '2016', '2017']

data = {'fruits' : fruits,
        '2015'   : [2, 1, 4],
        '2016'   : [5, 3, 3],
        '2017'   : [3, 2, 4]}

source = ColumnDataSource(data=data)

p = figure(x_range=fruits, y_range=(0, 10), plot_height=250,
           title="Fruit Counts by Year")

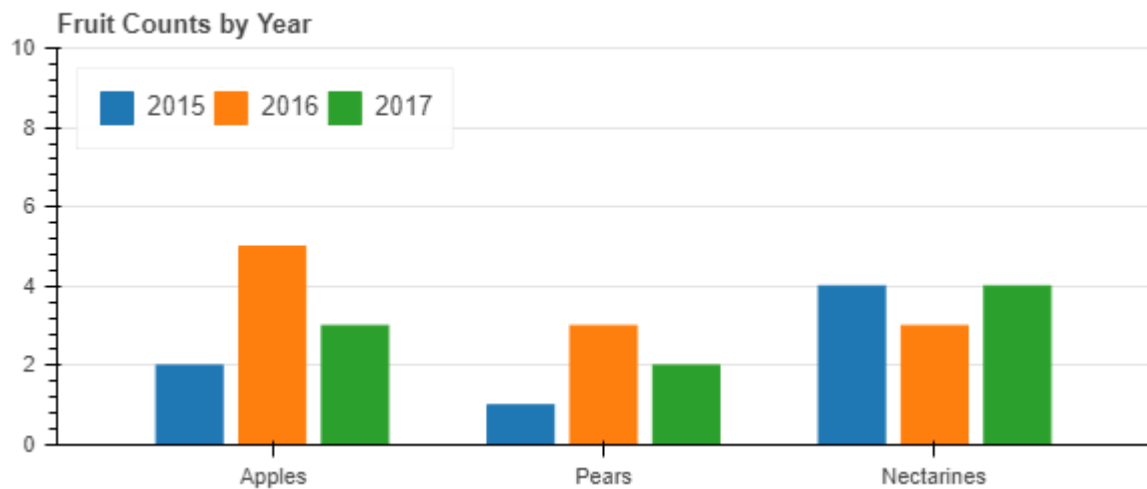
p.vbar(x=dodge('fruits', -0.25, range=p.x_range), top='2015',
       width=0.2, source=source,
       color=Category10_10[0], legend=value("2015"))

p.vbar(x=dodge('fruits', 0.0, range=p.x_range), top='2016',
       width=0.2, source=source,
       color=Category10_10[1], legend=value("2016"))

p.vbar(x=dodge('fruits', 0.25, range=p.x_range), top='2017',
       width=0.2, source=source,
       color=Category10_10[2], legend=value("2017"))

p.x_range.range_padding = 0.1
p.xgrid.grid_line_color = None
p.legend.location = "top_left"
p.legend.orientation = "horizontal"
```

```
show(p)
```



Vẽ bar chart theo nhóm chồng lên nhau

```
from bokeh.models import Legend
from bokeh.plotting import figure, show
from bokeh.palettes import Category10_10

projects = ['Dự án %s' % (x+1) for x in range(6)]
status = ['Đang làm', 'Hoàn thành', 'Tạm hoãn', 'Hủy', 'Duyệt']
colors = Category10_10[0:5]

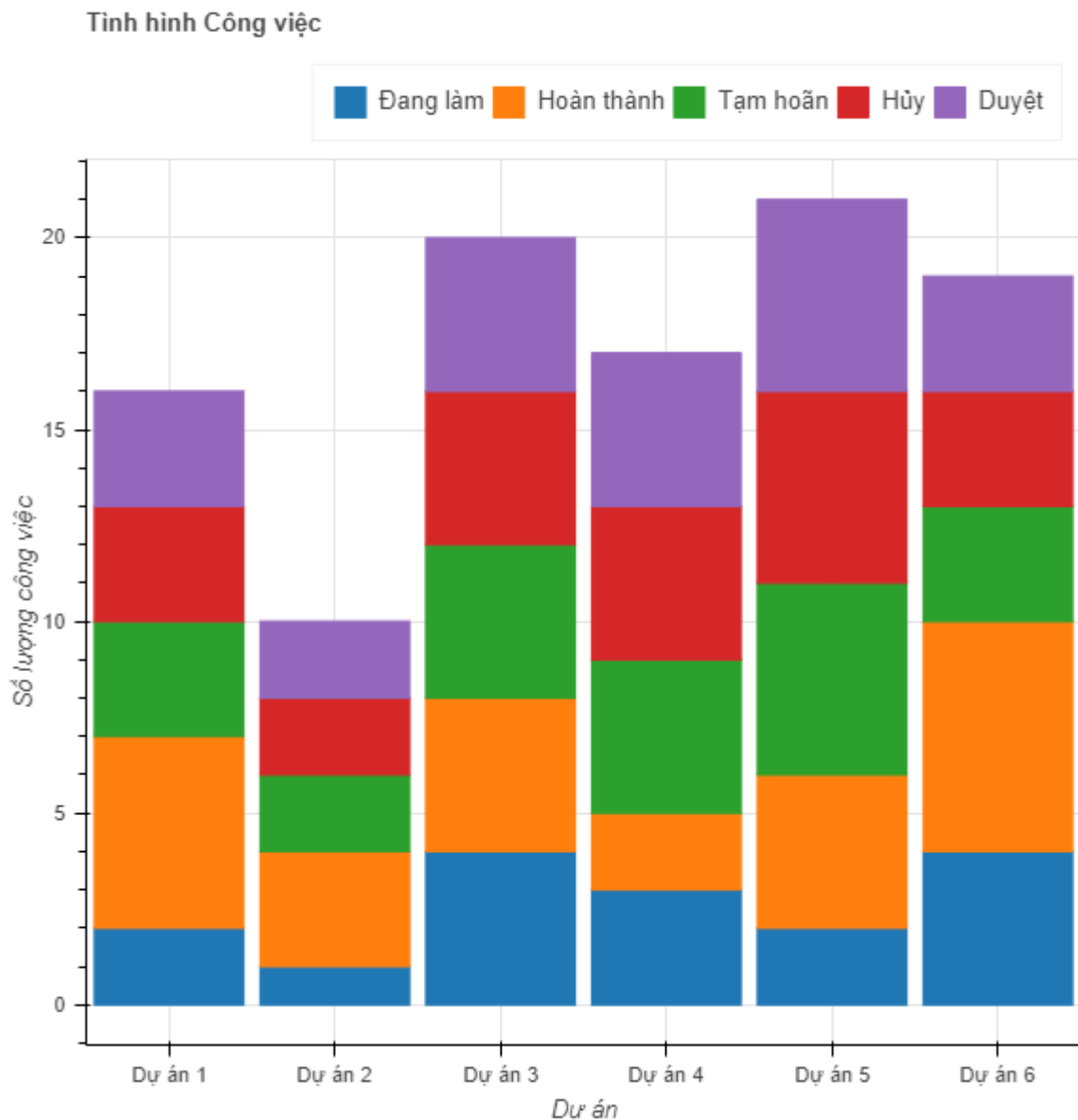
data = {'projects' : projects,
        'Đang làm'   : [2, 1, 4, 3, 2, 4],
        'Hoàn thành' : [5, 3, 4, 2, 4, 6],
        'Tạm hoãn'   : [3, 2, 4, 4, 5, 3],
        'Hủy'        : [3, 2, 4, 4, 5, 3],
        'Duyệt'      : [3, 2, 4, 4, 5, 3],
        }

p = figure(x_range=projects, title="Tình hình Công việc")
p.add_layout(Legend(orientation = 'horizontal'), 'above')

p.vbar_stack(status, x='projects', width=0.9, color=colors,
             source=data, legend_label=status)

p.xaxis.axis_label = 'Dự án'
p.yaxis.axis_label = 'Số lượng công việc'
```

```
show(p)
```



Xem thêm cách vẽ các thành phần cơ bản tại link:

https://docs.bokeh.org/en/latest/docs/user_guide/plotting.html

Xem thêm cách vẽ các loại biểu đồ tại link:

<https://docs.bokeh.org/en/latest/docs/reference/plotting.html>

Trình bày giao diện với Bokeh

Một trong các điểm mạnh của thư viện Bokeh là hỗ trợ kiến trúc giúp chúng ta có thể trình bày trực quan kết quả phân tích dữ liệu dưới dạng một ứng dụng web tương tác hoàn chỉnh.

Phần này sẽ giúp bạn làm quen với cách thiết kế một trang web gồm nhiều thành phần, trình bày dưới dạng nhiều bố cục (layout) khác nhau.

Bố cục dạng cột

Để trình bày các biểu đồ trên cùng một cột thì dùng hàm `column(..)` để liệt kê các đối tượng Figure trong các tham số như ví dụ sau:

```
from bokeh.io import output_file, show
from bokeh.layouts import column
from bokeh.plotting import figure

output_file("layout_column.html")

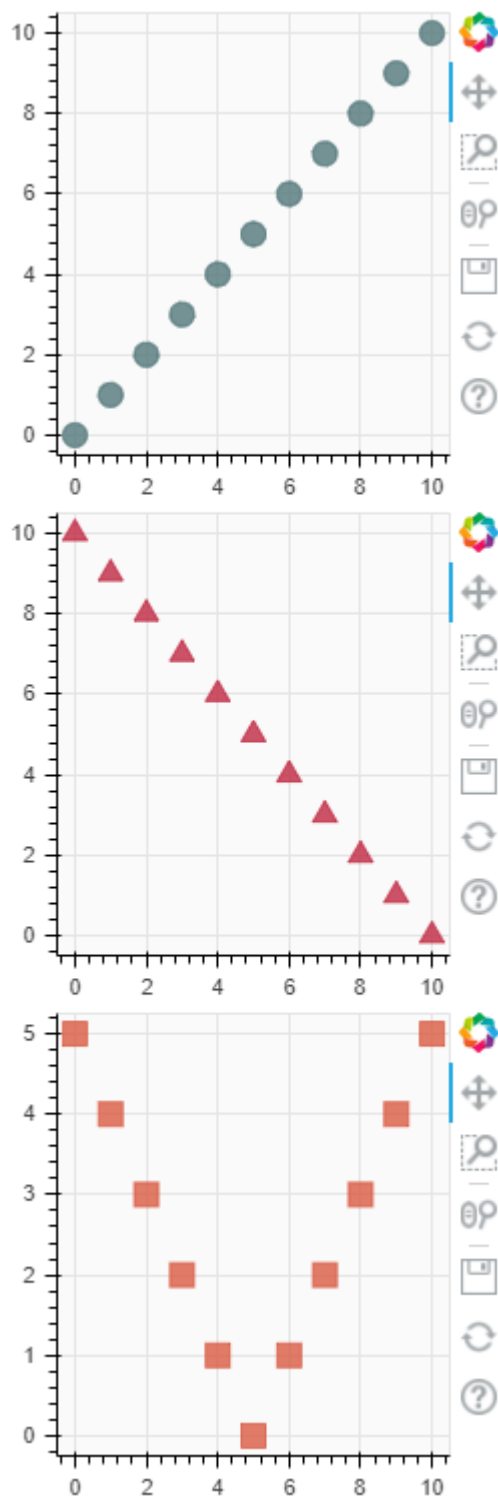
x = list(range(11))
y0 = x
y1 = [10 - i for i in x]
y2 = [abs(i - 5) for i in x]

# create three plots
s1 = figure(plot_width=250, plot_height=250,
background_fill_color="#fafafa")
s1.circle(x, y0, size=12, color="#53777a", alpha=0.8)

s2 = figure(plot_width=250, plot_height=250,
background_fill_color="#fafafa")
s2.triangle(x, y1, size=12, color="#c02942", alpha=0.8)

s3 = figure(plot_width=250, plot_height=250,
background_fill_color="#fafafa")
s3.square(x, y2, size=12, color="#d95b43", alpha=0.8)

# put the results in a column and show
layout = column(s1, s2, s3)
show(layout)
```



Bố cục dạng dòng

```
from bokeh.io import output_file, show
from bokeh.layouts import row
from bokeh.plotting import figure

output_file("layout_row.html")
```



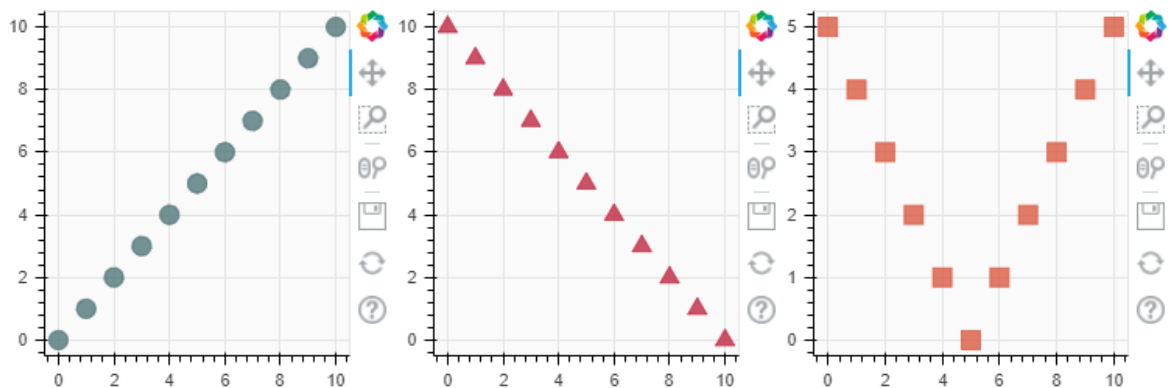
```
x = list(range(11))
y0 = x
y1 = [10 - i for i in x]
y2 = [abs(i - 5) for i in x]

# create three plots
s1 = figure(plot_width=250, plot_height=250,
background_fill_color="#fafafa")
s1.circle(x, y0, size=12, color="#53777a", alpha=0.8)

s2 = figure(plot_width=250, plot_height=250,
background_fill_color="#fafafa")
s2.triangle(x, y1, size=12, color="#c02942", alpha=0.8)

s3 = figure(plot_width=250, plot_height=250,
background_fill_color="#fafafa")
s3.square(x, y2, size=12, color="#d95b43", alpha=0.8)

# put the results in a row and show
layout = row(s1, s2, s3)
show(layout)
```



Kết hợp dòng và cột

Trang trí biểu đồ

Thiết lập tiêu đề, tên trục x và y

```
import pandas as pd
from bokeh.plotting import figure
from bokeh.models.annotations import Title
```

Ứng dụng Phân tích dữ liệu và Trí tuệ nhân tạo với Python

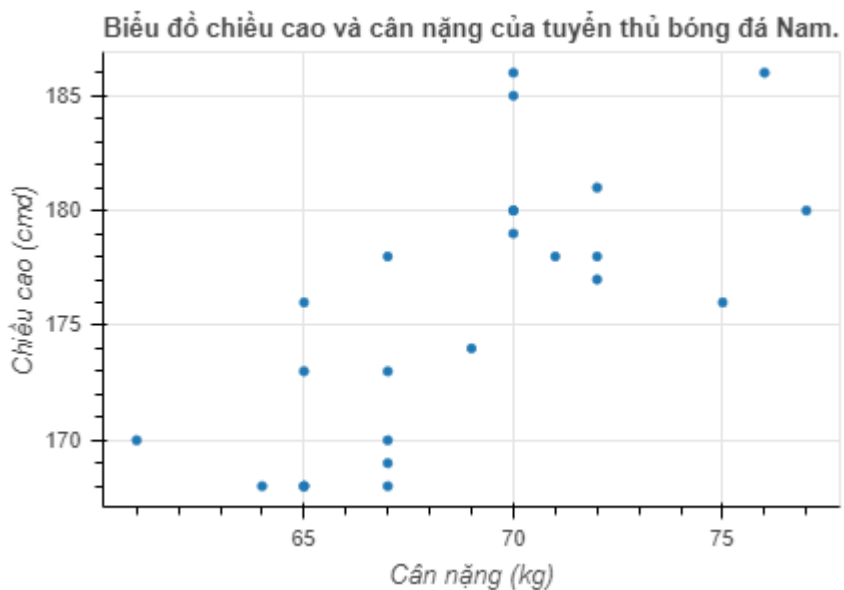
```
from bokeh.io import show

df =
pd.read_csv('https://thachln.github.io/datasets/TuyenVN_2019.csv')

x = df['Weight']
y = df['Height']

# plot it
p = figure()
p.circle(x,y)

p.xaxis.axis_label = 'Cân nặng (kg)'
p.yaxis.axis_label = 'Chiều cao (cmd)'
p.title = Title(text = 'Biểu đồ chiều cao và cân nặng của tuyển thủ bóng đá Nam.')
show(p)
```



Thay đổi vị trí của legend

```
p.legend.location = "top_left"
```