Lab 2. THỐNG KÊ MÔ TẢ

Nội dung:

1. Xây dựng histogram
2. Xây dựng scatterplot
3. Xây dựng bar chart và pie char
4. Tính các giá trị thống kê: trung bình (mean), trung vị (median), range (min, max), phương sai (varian), độ lệch chuẩn (standard deviation)
5. Xây dựng box plot
6. Kiểm tra dạng chuẩn

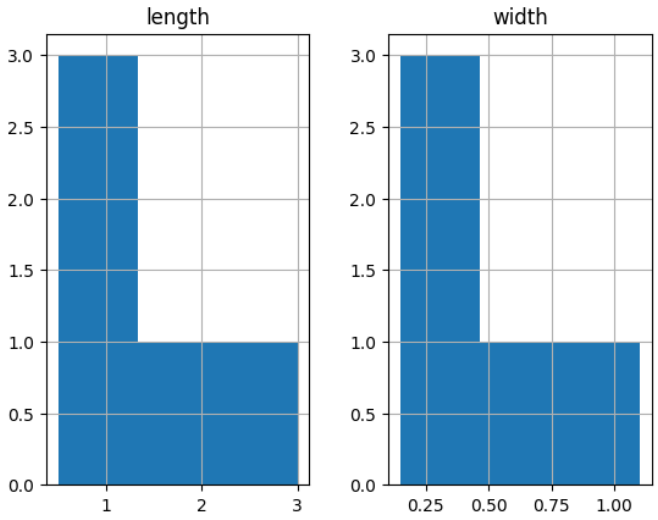
1. Xây dựng histogram

Hướng dẫn:

Cách 1: Dùng DataFrame của package Pandas

Ví dụ:

```
import pandas as pd
df=pd.DataFrame({
    'length':[1.5,0.5,1.2,0.9,3],
    'width':[0.7,0.2,0.15,0.2,1.1]
}, index=['pig','rabbit','duck','chicken','horse'])
hist=df.hist(bins=3)
```



Cách 2: dùng hàm **matplotlib.pyplot.hist**

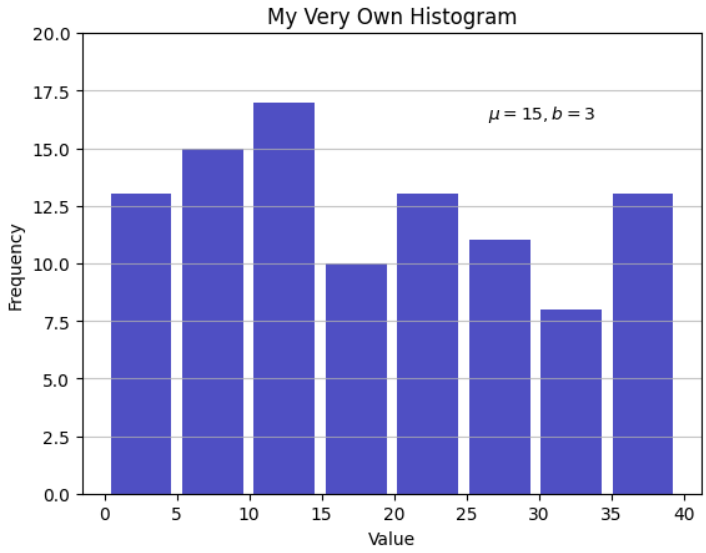
plt.hist(): Vẽ biểu đồ histogram từ dữ liệu d.

- x=d: Truyền dữ liệu cần vẽ là mảng d.
- bins='auto': Tự động chọn số lượng "bins" (khoảng chia giá trị) sao cho tối ưu.
- color='#0504aa': Màu sắc của các cột trong biểu đồ là màu xanh đậm (#0504aa).
- alpha=0.7: Độ trong suốt của các cột, giá trị từ 0 (trong suốt) đến 1 (không trong suốt).
- rwidth=0.85: Độ rộng của các cột trong biểu đồ, so với khoảng cách giữa các giá trị.

Ví dụ:

```
import matplotlib.pyplot as plt
import numpy as np
d = np.random.uniform(0, 40, 100)
#n: ds số lượng phần tử trong mỗi bin; bins: Mảng chứa các giá trị biên của mỗi
bin; patches: Đối tượng đại diện cho mỗi hình chữ nhật trong biểu đồ
n, bins, patches = plt.hist(x=d, bins='auto', color='#0504aa', alpha=0.7, rwidth=0.
85)
#Hiển thị các đường lưới trên trục y (tần suất), giúp dễ đọc giá trị hơn. alpha=0.
75 làm cho lưới có độ trong suốt
plt.grid(axis='y', alpha=0.75)
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('My Very Own Histogram')
#Thêm văn bản vào biểu đồ tại tọa độ (x=23, y=45). Ở đây $\mu=15, \sigma=3$ là
công thức toán học được viết dưới dạng LaTeX (biểu diễn ký hiệu trung bình μ và độ
lệch chuẩn σ).
plt.text(23, 45, r'$\mu=15, \sigma=3$')
maxfreq=n.max()
#set a clean upper y-axis limit
plt.ylim(ymax=np.ceil(maxfreq/10)*10 if maxfreq % 10 else maxfreq+10)
```

- plt.ylim(): giới hạn của trục y (ymax: giá trị lớn nhất của trục y)



Xây dựng histogram cho các bài tập sau:

- Old Faithful: biểu diễn thời gian (tính bằng giây) phun trào Old Faithful từ Dataset 15.
- Trọng lượng của Diet Coke: biểu diễn trọng lượng (tính bằng pound) của Diet Coke từ Dataset 17

Bài Dataset 15:

1. Duration (Thời gian phun trào):

- **Mô tả**: Cột này thể hiện **thời gian phun trào của mạch nước Old Faithful**, tính bằng giây hoặc phút (tùy thuộc vào đơn vị trong dataset). Đây là khoảng thời gian trong đó mạch nước phun trào liên tục.
- **Ý nghĩa**: Thời gian này giúp theo dõi và phân tích chu kỳ phun trào của mạch nước ngầm.

2. Interval Before (Khoảng thời gian trước phun trào):

- **Mô tả**: Đây là khoảng **thời gian từ lần phun trào trước đến lần phun trào hiện tại**, thường tính bằng phút.
- **Ý nghĩa**: Cho biết thời gian giữa hai lần phun trào liên tiếp. Điều này rất hữu ích trong việc dự đoán khi nào mạch nước sẽ phun trào tiếp theo.

3. Interval After (Khoảng thời gian sau phun trào):

- **Mô tả**: Đây là khoảng **thời gian từ lần phun trào hiện tại đến lần phun trào kế tiếp**.
- **Ý nghĩa**: Cột này giúp xác định mối liên hệ giữa thời gian phun trào và khoảng thời gian giữa các lần phun trào. Nó có thể giúp dự đoán chu kỳ phun trào tiếp theo dựa trên dữ liệu trước đó.

4. Height (Chiều cao):

- **Mô tả**: Cột này thể hiện **chiều cao của cột nước phun trào** trong mỗi lần phun trào, tính bằng mét hoặc feet (tùy thuộc vào đơn vị trong dataset).
- **Ý nghĩa**: Chiều cao của cột nước phun có thể thay đổi tùy theo áp lực và các yếu tố tự nhiên khác. Điều này giúp phân tích quy mô và sức mạnh của từng lần phun trào.

5. Prediction Error (Sai số dự đoán):

- **Mô tả**: Đây là **độ sai lệch giữa giá trị dự đoán và giá trị thực tế** của thời gian phun trào hoặc chiều cao phun trào.
- **Ý nghĩa**: Cột này thường được sử dụng khi có một mô hình dự đoán về các thuộc tính của mạch nước phun trào, ví dụ như thời gian hoặc chiều cao. Nó giúp đánh giá mức độ chính xác của mô hình.

Bài Dataset 17:

SGHG

1. CKREGWT:

- **Mô tả**: Trọng lượng của sản phẩm cola thông thường (regular cola), thường tính bằng pound.
- **Ý nghĩa**: Chỉ số này cho biết trọng lượng của sản phẩm cola thông thường, giúp so sánh với các loại cola khác, như diet.

2. CKREGVOL:

- **Mô tả**: Thể tích của sản phẩm cola thông thường, thường tính bằng ounce hoặc milliliters.
- **Ý nghĩa**: Thể tích cho phép người tiêu dùng biết được lượng sản phẩm trong mỗi chai hoặc lon.

3. CKDIETWT:

- **Mô tả**: Trọng lượng của sản phẩm Diet Coke, thường tính bằng pound.
- **Ý nghĩa**: Chỉ số này giúp xác định trọng lượng của Diet Coke, hỗ trợ phân tích sự khác biệt giữa các loại cola.

4. CKDTVOL:

- **Mô tả**: Thể tích của Diet Coke, thường tính bằng ounce hoặc milliliters.
- **Ý nghĩa**: Tương tự như CKREGVOL, cột này cho biết thể tích của sản phẩm Diet Coke trong mỗi đơn vị.

5. PPREGWT:

- **Mô tả**: Trọng lượng của sản phẩm cola thông thường trong một gói (package), thường tính bằng pound.
- **Ý nghĩa**: Cung cấp thông tin về trọng lượng sản phẩm cola thông thường khi được bán theo gói.

6. PPREGVOL:

- **Mô tả**: Thể tích của sản phẩm cola thông thường trong một gói, thường tính bằng ounce hoặc milliliters.
- **Ý nghĩa**: Cho biết thể tích của sản phẩm cola thông thường khi bán theo gói, phục vụ cho việc so sánh với các sản phẩm khác.

7. PPDIETWT:

- **Mô tả**: Trọng lượng của sản phẩm Diet Coke trong một gói, thường tính bằng pound.
- **Ý nghĩa**: Cung cấp thông tin về trọng lượng sản phẩm Diet Coke khi được bán theo gói.

8. PPDTVOL:

- **Mô tả**: Thể tích của Diet Coke trong một gói, thường tính bằng ounce hoặc milliliters.
- **Ý nghĩa**: Thể tích của Diet Coke trong một gói, giúp người tiêu dùng biết được lượng sản phẩm khi mua.

Tóm tắt:

Các cột này chủ yếu cung cấp thông tin về trọng lượng và thể tích của các sản phẩm cola khác nhau, bao gồm cola thông thường và Diet Coke, cả ở dạng đơn lẻ và trong gói. Thông tin này rất hữu ích cho các nhà nghiên cứu và người tiêu dùng để so sánh và phân tích các sản phẩm cola khác nhau

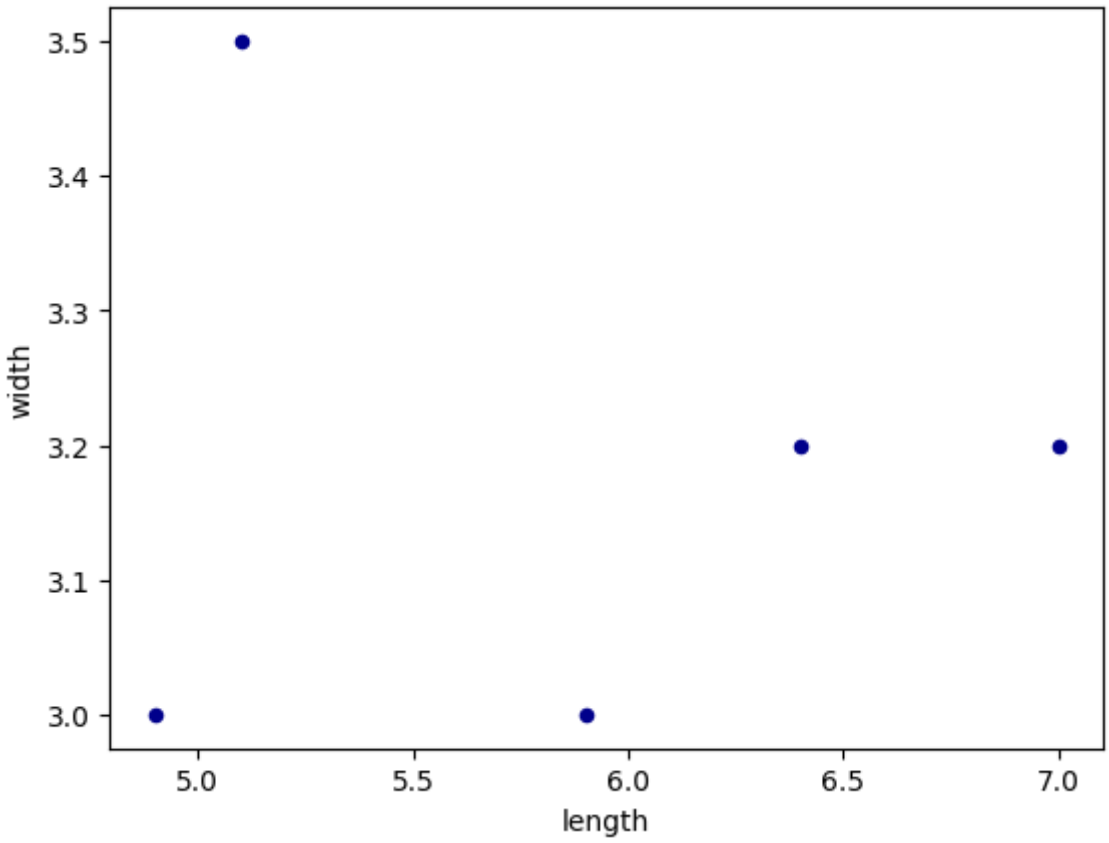
2. Xây dựng Scatterplot:

Hướng dẫn:

Cách 1: Dùng DataFrame của package Pandas

Ví dụ:

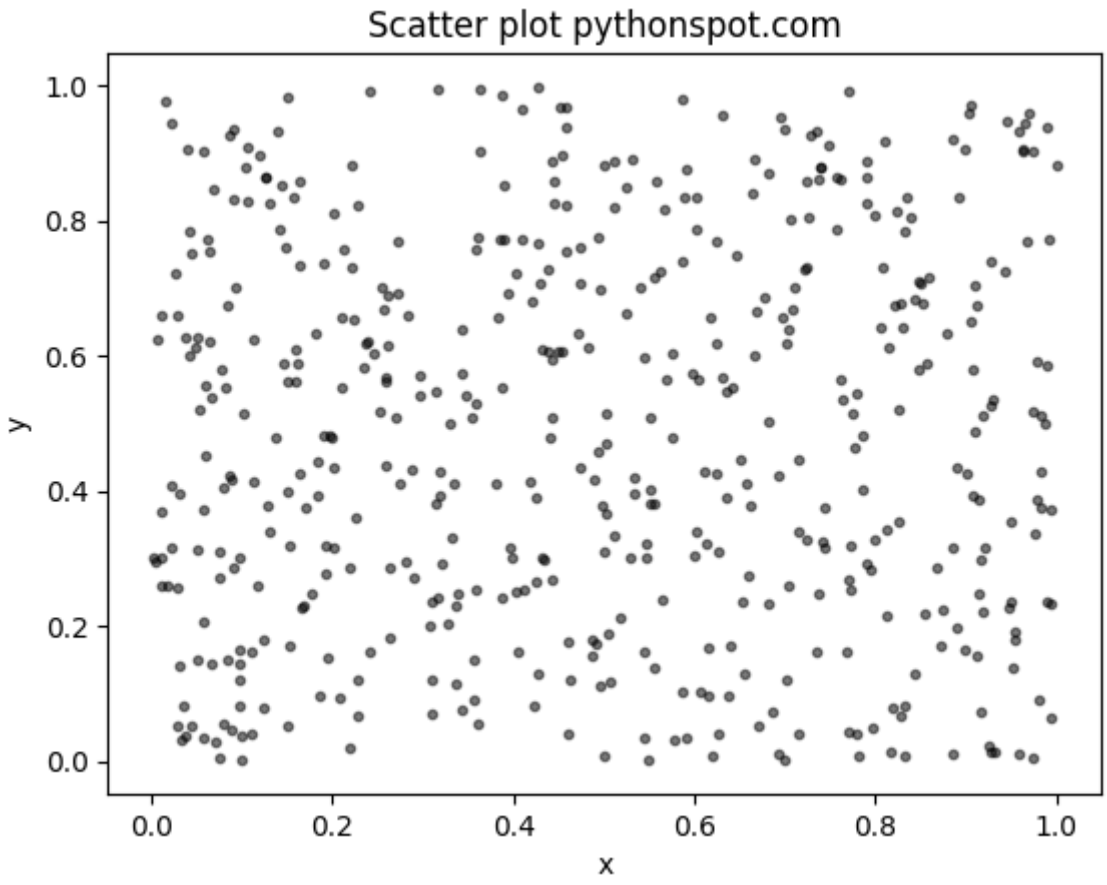
```
import numpy as np
import pandas as pd

# Tạo DataFrame với dữ liệu chiều dài, chiều rộng và loài
df = pd.DataFrame([[5.1, 3.5, 0],
                   [4.9, 3.0, 0],
                   [7.0, 3.2, 1],
                   [6.4, 3.2, 1],
                   [5.9, 3.0, 2]],
                  columns=['length', 'width', 'species'])

ax1 = df.plot.scatter(x='length', y='width', c='DarkBlue')
```



<u>Cách 2:</u> Dùng hàm **matplotlib.pyplot.scatter**

Ví dụ:

```
import numpy as np
import matplotlib.pyplot as plt
#create data
N=500
x = np.random.rand(N)
y = np.random.rand(N)
colors = (0,0,0)
area = np.pi*3
#plot
plt.scatter(x, y, s=area, c=colors, alpha=0.5)
plt.title('Scatter plot pythonspot.com')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



Xây dựng scatter plot cho các bài tập sau:

➢ **Nhựa/CO trong thuốc lá:** trong Dataset 4, biểu diễn **thuộc tính nhựa** trong thuốc lá cỡ king trên trục **X** và sử dụng carbon monoxide **(CO)** trong cùng loại thuốc lá cỡ king trên trục **Y**. Xác định mối quan hệ giữa **nhựa thuốc lá** và **CO** trong thuốc lá cỡ king.

1. KgTar:

- **Mô tả**: Trọng lượng nhựa (tar) trong thuốc lá, thường tính bằng kilogram.
- **Ý nghĩa**: Thể hiện lượng nhựa có trong thuốc lá, một yếu tố quan trọng ảnh hưởng đến sức khỏe.

2. KgNic:

- **Mô tả**: Trọng lượng nicotine trong thuốc lá, thường tính bằng kilogram.
- **Ý nghĩa**: Đo lượng nicotine, một chất gây nghiện, có trong thuốc lá. Nó ảnh hưởng đến cảm giác và mức độ nghiện của người tiêu dùng.

3. KgCO:

- **Mô tả**: Trọng lượng carbon monoxide (CO) trong thuốc lá, thường tính bằng kilogram.
- **Ý nghĩa**: Đo lường lượng CO, một khí độc sinh ra từ quá trình đốt thuốc lá. CO có thể gây ra nhiều vấn đề sức khỏe.

4. MnTar:

- **Mô tả**: Trung bình nhựa (tar) trong thuốc lá, thường tính bằng kilogram.
- **Ý nghĩa**: Thể hiện giá trị trung bình của nhựa trong một loại thuốc lá hoặc mẫu nghiên cứu, giúp đánh giá mức độ nhựa tổng thể.

5. MnNic:

- **Mô tả**: Trung bình nicotine trong thuốc lá, thường tính bằng kilogram.

- **Ý nghĩa**: Cung cấp thông tin về lượng nicotine trung bình có trong các mẫu thuốc lá.

6. MnCO:

- **Mô tả**: Trung bình carbon monoxide (CO) trong thuốc lá, thường tính bằng kilogram.
- **Ý nghĩa**: Thể hiện giá trị trung bình của CO, giúp đánh giá sự an toàn và ảnh hưởng đến sức khỏe.

7. FLTar:

- **Mô tả**: Lượng nhựa (tar) trong thuốc lá ở mức thấp, thường tính bằng kilogram.
- **Ý nghĩa**: Cung cấp thông tin về lượng nhựa trong các loại thuốc lá có nồng độ nhựa thấp hơn.

8. FLNic:

- **Mô tả**: Lượng nicotine trong thuốc lá ở mức thấp, thường tính bằng kilogram.
- **Ý nghĩa**: Thể hiện lượng nicotine trong các loại thuốc lá có nồng độ thấp.

9. FLCO:

- **Mô tả**: Lượng carbon monoxide (CO) trong thuốc lá ở mức thấp, thường tính bằng kilogram.
- **Ý nghĩa**: Cung cấp thông tin về nồng độ CO trong các loại thuốc lá có nồng

➢ **Tiêu thụ năng lượng và nhiệt độ**: trong Dataset 12, sử dụng 22 giá trị trị nhiệt độ trung bình hàng ngày và sử dụng 22 giá trị lượng tiêu thụ năng lượng tương ứng (kWh). (Sử dụng nhiệt độ biểu diễn theo trục X). Dựa trên kết quả, có mối quan hệ giữa nhiệt độ trung bình hàng ngày và lượng năng lượng tiêu thụ hay không?

1. Time Period:

- **Mô tả**: Thời gian mà dữ liệu được ghi nhận, có thể là theo ngày, tuần hoặc tháng.
- **Ý nghĩa**: Giúp xác định thời điểm mà lượng tiêu thụ năng lượng và nhiệt độ được ghi nhận, có thể sử dụng để phân tích theo thời gian.

2. kWh:

- **Mô tả**: Lượng tiêu thụ năng lượng điện, thường tính bằng kilowatt-giờ (kWh).
- **Ý nghĩa**: Đây là chỉ số quan trọng để đánh giá mức tiêu thụ năng lượng của hộ gia đình hoặc cơ sở kinh doanh. Nó ảnh hưởng đến hóa đơn điện và tiêu thụ tài nguyên.

3. Cost:

- **Mô tả**: Chi phí tiêu thụ năng lượng, thường tính bằng đơn vị tiền tệ (như USD, VND, etc.).
- **Ý nghĩa**: Thể hiện tổng chi phí mà người tiêu dùng phải trả cho lượng điện đã tiêu thụ. Nó giúp người tiêu dùng theo dõi và quản lý chi phí năng lượng của mình.

4. Deg Days:

- **Mô tả**: Số ngày độ (degree days), thường được sử dụng để tính toán mức tiêu thụ năng lượng cho hệ thống điều hòa không khí và sưởi ấm.
- **Ý nghĩa**: Số ngày độ giúp dự đoán nhu cầu năng lượng dựa trên sự chênh lệch giữa nhiệt độ bên ngoài và nhiệt độ mục tiêu cho sưởi ấm hoặc làm mát.

5. AvTemp:

- **Mô tả**: Nhiệt độ trung bình hàng ngày, thường tính bằng độ Celsius (°C) hoặc độ Fahrenheit (°F).

- **Ý nghĩa**: Cung cấp thông tin về điều kiện thời tiết trung bình trong một khoảng thời gian nhất định, có thể ảnh hưởng đến nhu cầu tiêu thụ năng lượng cho sưởi ấm hoặc làm mát.

3. Xây dựng Bar char và Pie char:

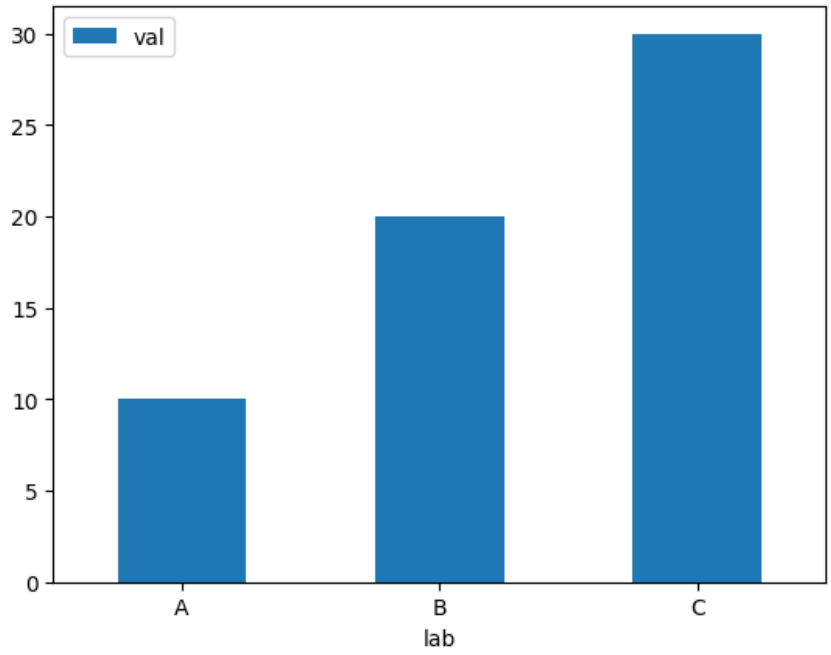
3.1. Bar chart

Hướng dẫn:

Cách 1: Dùng DataFrame của package Pandas

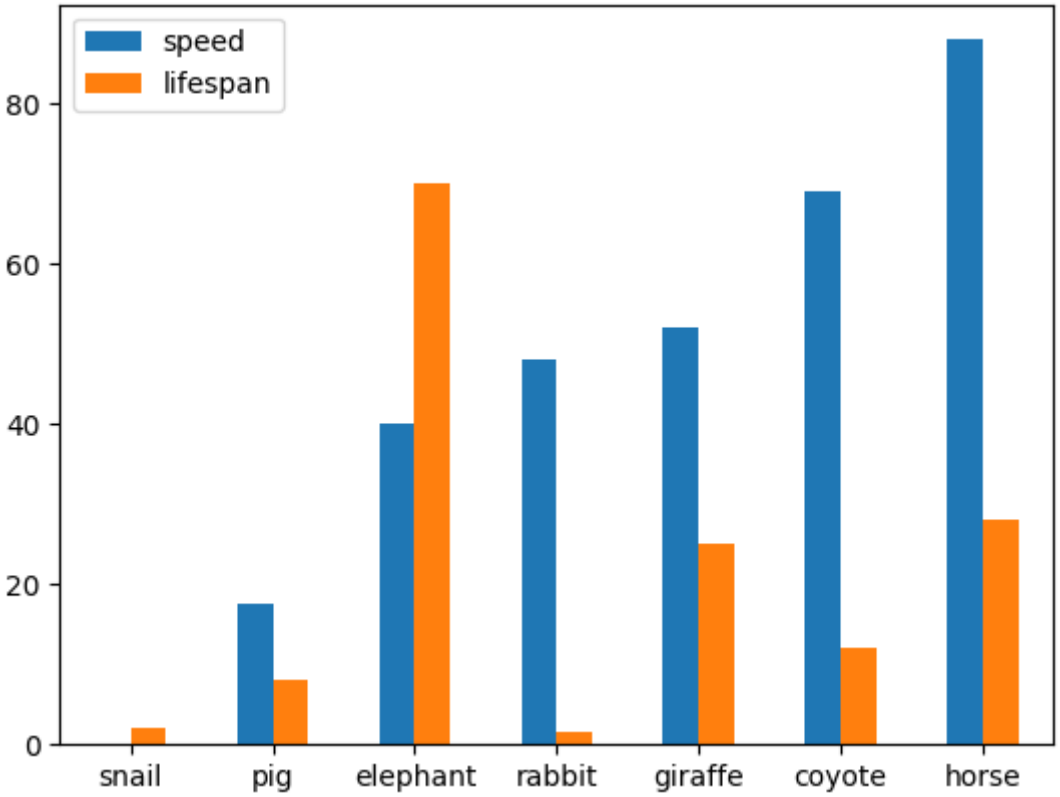
Ví dụ 1:

```
import pandas as pd
df=pd.DataFrame({'lab':['A','B','C'],'val':[10,20,30]})
ax=df.plot.bar(x='lab',y='val',rot=0)
```



Ví dụ 2:

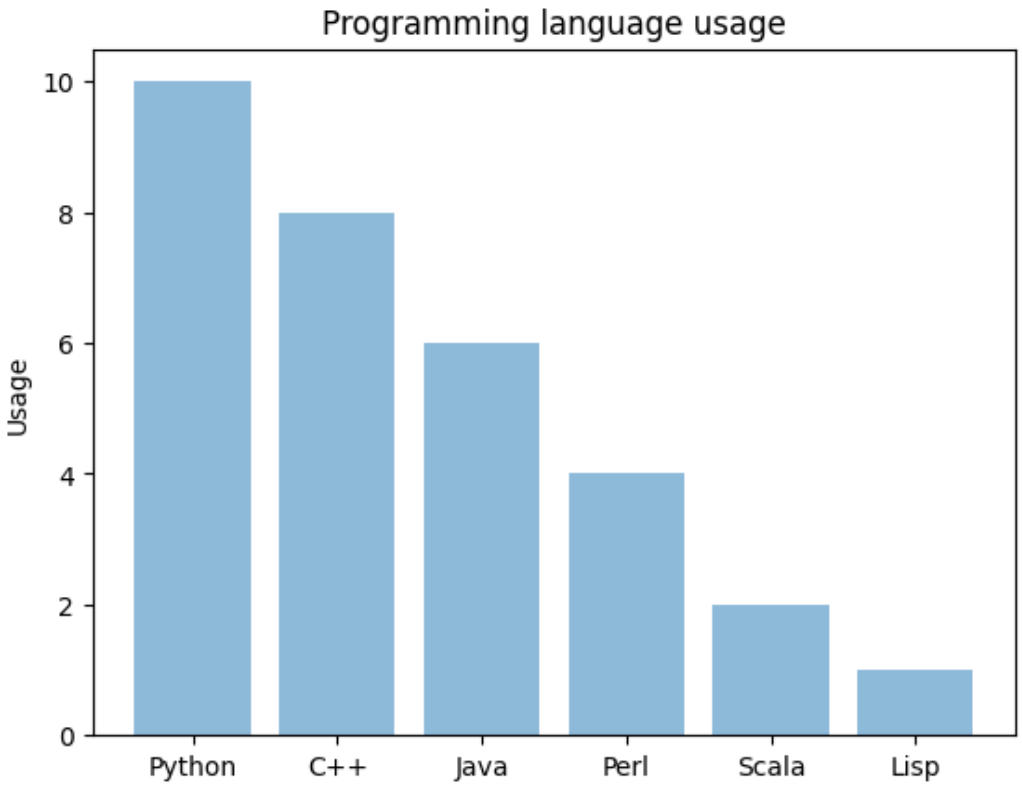
```
speed=[0.1,17.5,40,48,52,69,88]
lifespan=[2,8,70,1.5,25,12,28]
index=['snail','pig','elephant','rabbit','giraffe','coyote',
'horse']
df=pd.DataFrame({'speed':speed,'lifespan':lifespan},index=index)
ax=df.plot.bar(rot=0)
```



Cách 2: **Dùng matplotlib.pyplot.bar**

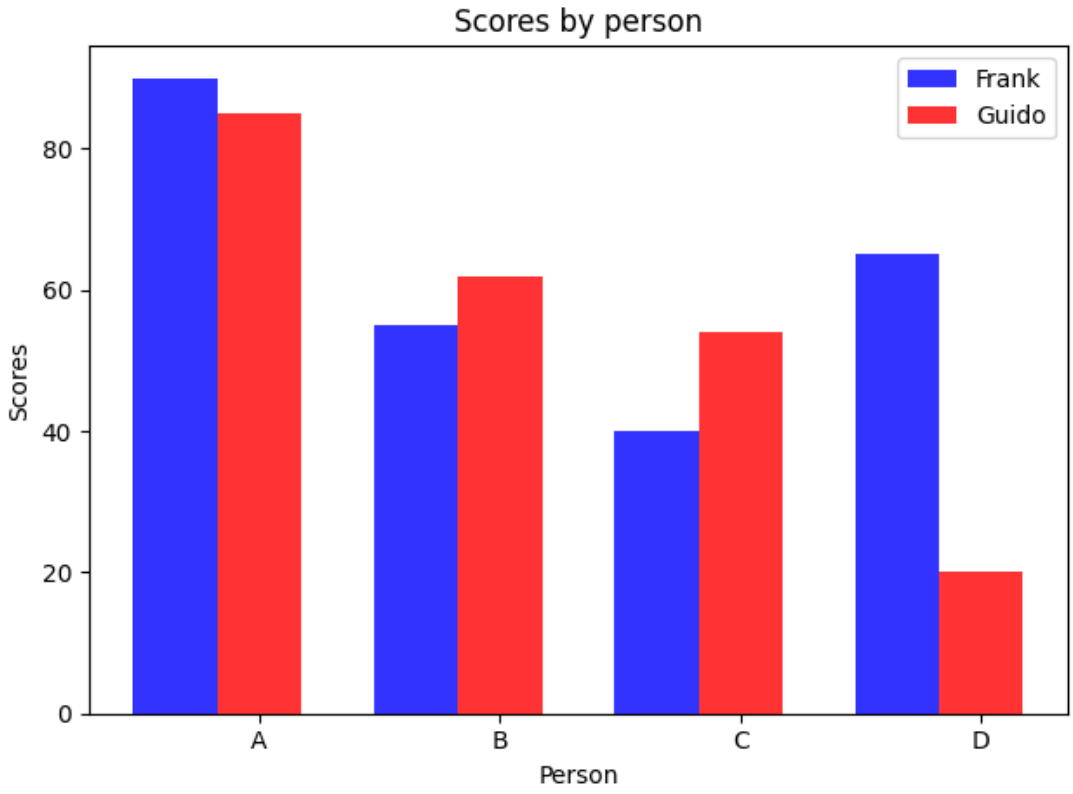
Ví dụ 1:

```
import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt
objects=('Python','C++','Java','Perl','Scala','Lisp')
y_pos=np.arange(len(objects))
performance=[10,8,6,4,2,1]

plt.bar(y_pos,performance,align='center',alpha=0.5)
plt.xticks(y_pos,objects)
plt.ylabel('Usage')
plt.title('Programming language usage')

plt.show()
```


Programming language usage
Usage
0
2
4
6
8
10
Python
C++
Java
Perl
Scala
Lisp

Ví dụ 2:

```
import matplotlib.pyplot as plt
import numpy as np

#data to plot
n_groups=4
means_frank=(90,55,40,65)
means_guido=(85,62,54,20)

#create plot
fig,ax=plt.subplots()
index=np.arange(n_groups)
bar_width=0.35
opacity=0.8

rects1=plt.bar(index,means_frank,bar_width,
               alpha=opacity,color='b',label='Frank')

rects2=plt.bar(index+bar_width,means_guido,bar_width,
                alpha=opacity,color='r',label='Guido')

plt.xlabel('Person')
plt.ylabel('Scores')
plt.title('Scores by person')
plt.xticks(index+bar_width,('A','B','C','D'))
plt.legend()

plt.tight_layout()
plt.show()
```



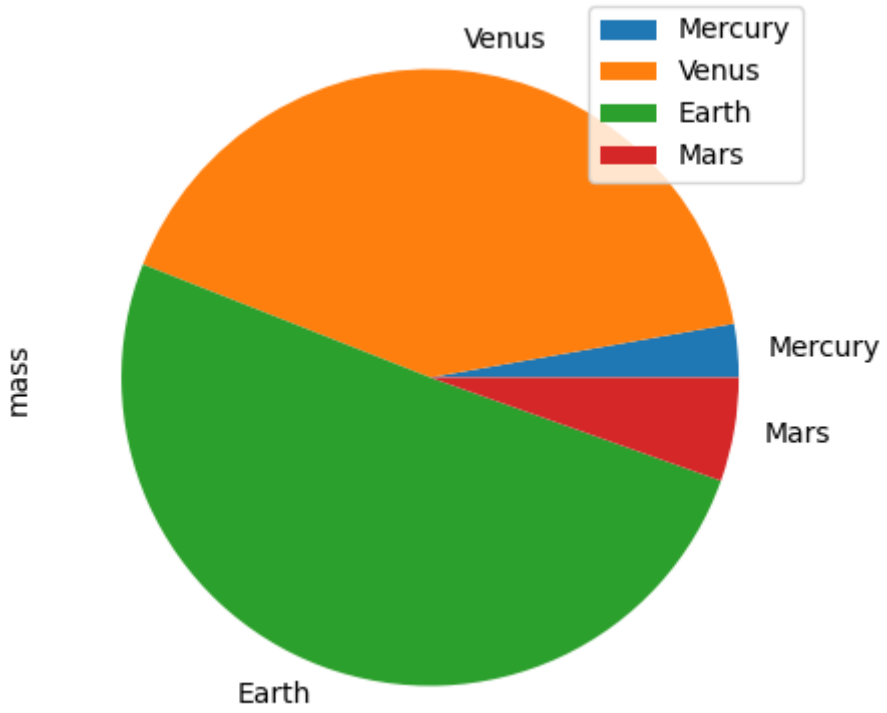
3.2. Pie chart

Hướng dẫn:

Cách 1: Dùng DataFrame của package Pandas

Ví dụ:

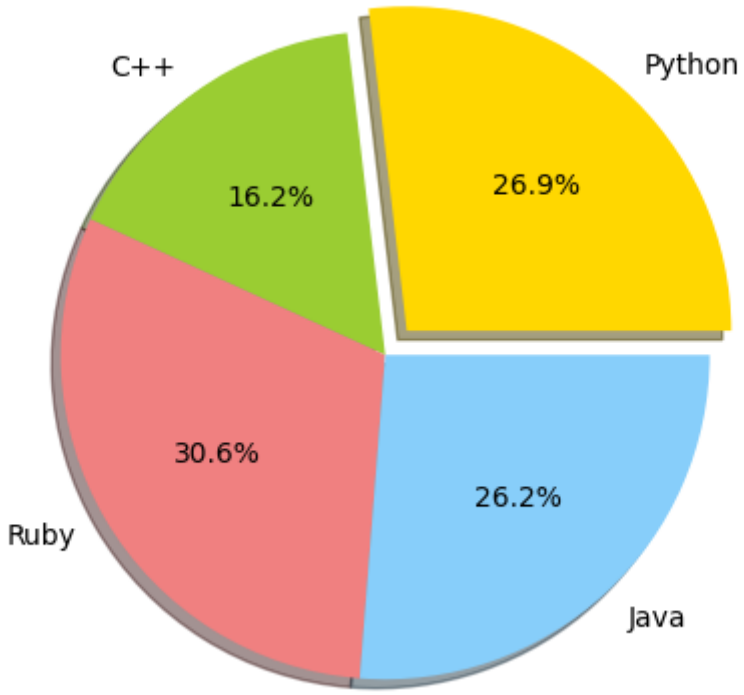
```
import pandas as pd
df=pd.DataFrame({'mass':[0.330,4.87,5.97,0.642],
                 'radius':[2439.7,6051.8,6378.1,3389.5]},
                 index=['Mercury','Venus','Earth','Mars'])

plot=df.plot.pie(y='mass',figsize=(5,5))
```



Cách 2: dùng matplotlib.pyplot.pie

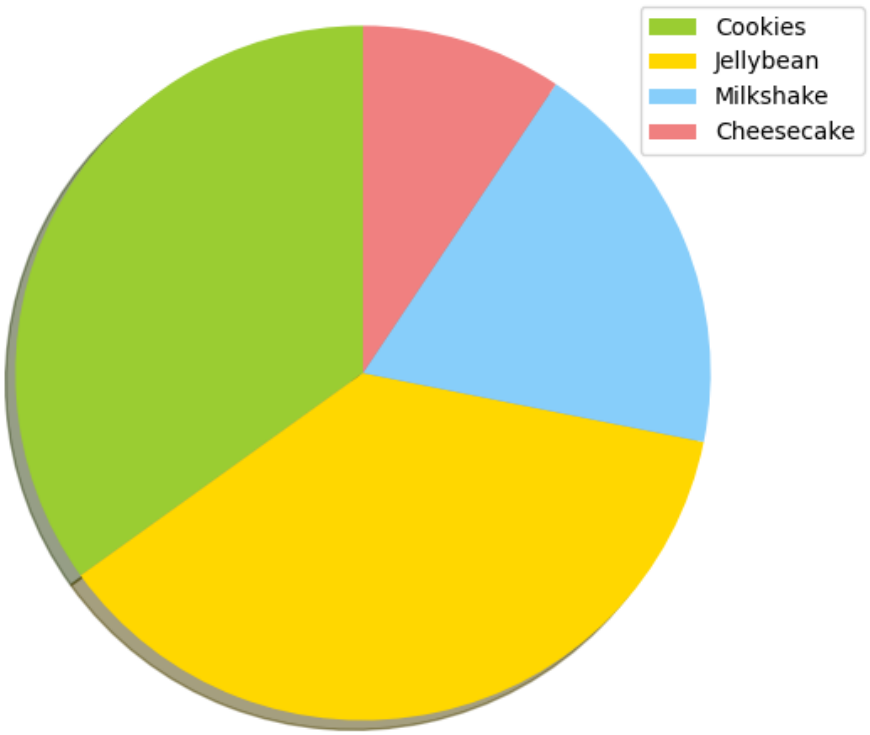
Ví dụ:

```
import matplotlib.pyplot as plt

#data to plot
labels='Python','C++','Ruby','Java'
sizes=[215,130,245,210]
colors=['gold','yellowgreen','lightcoral','lightskyblue']
explode=(0.1,0,0,0)

#plot
plt.pie(sizes,explode=explode,labels=labels,colors=colors,autopct='%1.1f%%',
shadow=True)
plt.axis('equal')
plt.show()
```



Ví dụ 2:

```
import matplotlib.pyplot as plt

lables=['Cookies','Jellybean','Milkshake','Cheesecake']
sizes=[38.4,40.6,20.7,10.3]
colors=['yellowgreen','gold','lightskyblue','lightcoral']
patches,texts=plt.pie(sizes,colors=colors,shadow=True,startangle=90)
plt.legend(patches,lables,loc='best')
plt.axis('equal')
plt.tight_layout()
plt.show()
```



Bài tập 1: Xây dựng bar char và pie char cho dữ liệu ở bảng sau. So sánh 2 biểu đồ trên, biểu đồ nào là hiệu quả hơn trong việc hiển thị thông tin

College	Relative Frequency
Public 2-Year	36.8%
Public 4-Year	40.0%
Private 2-Year	1.6%
Private 4-Year	21.9%

4. Tính các giá trị thống kê: trung bình (mean), trung vị (median), range (min, max), phương sai (varian), độ lệch chuẩn (standard deviation)

Hướng dẫn:

Cách 1: dùng hàm mean(…), median(…), std(…), var(…), max(…), min(…) của DataFrame trong package pandas

Cách 2: dùng hàm mean(...), median(...), std(...), var(...), max(...), min(...) của package numpy

Tính các giá trị thống kê sau: trung bình (mean), trung vị (median):

- **Nhiệt độ cơ thể**: Sử dụng nhiệt độ cơ thể lúc 12:00 AM vào ngày 2 từ Dataset 2. Các kết quả có hỗ trợ hoặc mâu thuẫn với phát biểu "nhiệt độ trung bình của cơ thể là 98,6°F" hay không?
- **Vít máy:** Sử dụng độ dài được liệt kê của các vít máy từ DataSet 19. Các ốc vít được cho là có chiều dài 3/4 in. Kết quả về độ dài quy định có đúng không?
- **Điện áp gia đình:** So sánh mean và median từ 3 tập dữ liệu khác nhau của các mức điện áp đã đo từ Dataset 13.
- **Phim:** Dataset 9. Xét tổng tiền thu được từ hai thể loại phim khác nhau: những phim có xếp hạng R và những phim có xếp hạng PG hoặc PG-13. Các kết quả tính được có hỗ trợ cho phát biểu sau không: "phim có xếp hạng R có tổng tiền thu được lớn hơn vì chúng thu hút khán giả lớn hơn các bộ phim dược xếp hạng PG hoặc PG-13"?

Tính các giá trị thống kê sau: range (min, max), phương sai (varian), độ lệch chuẩn (standard deviation):

- **Nhiệt độ cơ thể:** Sử dụng nhiệt độ cơ thể lúc 12:00 AM vào ngày 2 từ Dataset 2.
- **Vít máy**: Sử dụng độ dài được liệt kê của các vít máy từ DataSet 19.
- **Điện áp gia đình**: So sánh phương sai từ 3 tập dữ liệu khác nhau của các mức điện áp đã đo từ Dataset 13.
- **Phim**: Dataset 9. Xét tổng tiền thu được từ hai thể loại phim khác nhau: những phim có xếp hạng R và những phim có xếp hạng PG hoặc PG-13. Xác định xem hai loại có giống nhau về phương sai không.

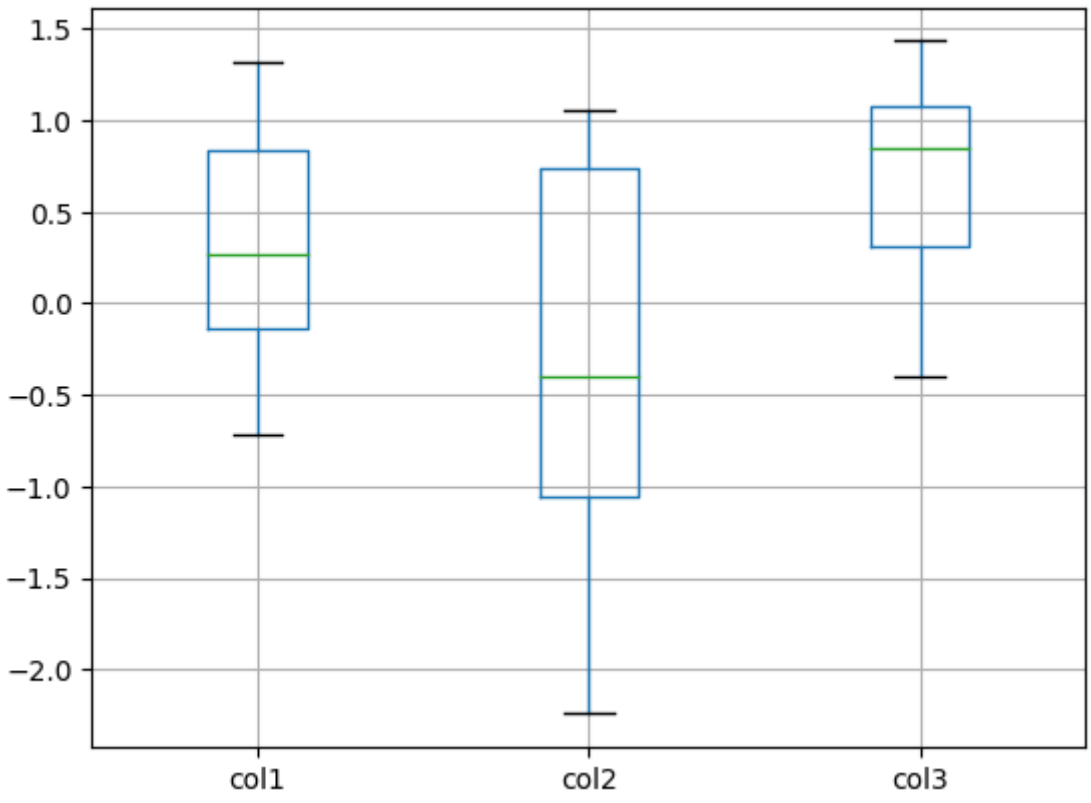
5. Xây dựng box plot

Hướng dẫn:

Cách 1: dùng hàm boxplot trong của DataFrame trong package Pandas.

Ví dụ:

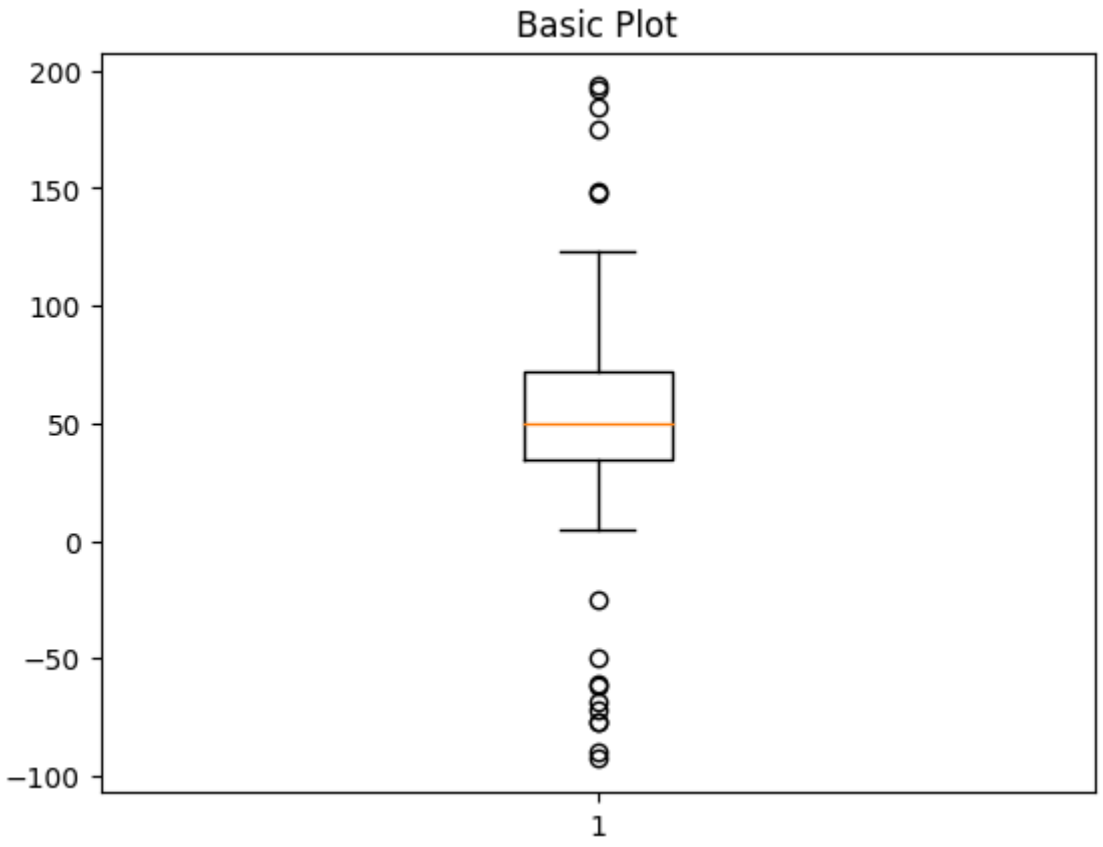
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt  # Đừng quên import matplotlib để vẽ boxplot

# Sửa tên hàm từ radom thành random
np.random.seed(1234)
df = pd.DataFrame(np.random.randn(10, 4),
                  columns=['col1', 'col2', 'col3', 'col4'])

# Vẽ boxplot cho các cột col1, col2, col3
boxplot = df.boxplot(column=['col1', 'col2', 'col3'])

# Hiển thị biểu đồ
plt.show()
```



Cách 2: dùng hàm matplotlib.pyplot.boxplot

Ví dụ:

```python
import numpy as np
import matplotlib.pyplot as plt

#fixing random state for reproducibility
np.random.seed(19680801)

#fake up some data
spread=np.random.rand(50)*100
center=np.ones(25)*50
flier_high=np.random.rand(10)*100+100
flier_low=np.random.rand(10)*-100
data=np.concatenate((spread,center,flier_high,flier_low))
fig1,ax1=plt.subplots()
ax1.set_title('Basic Plot')
ax1.boxplot(data)
```



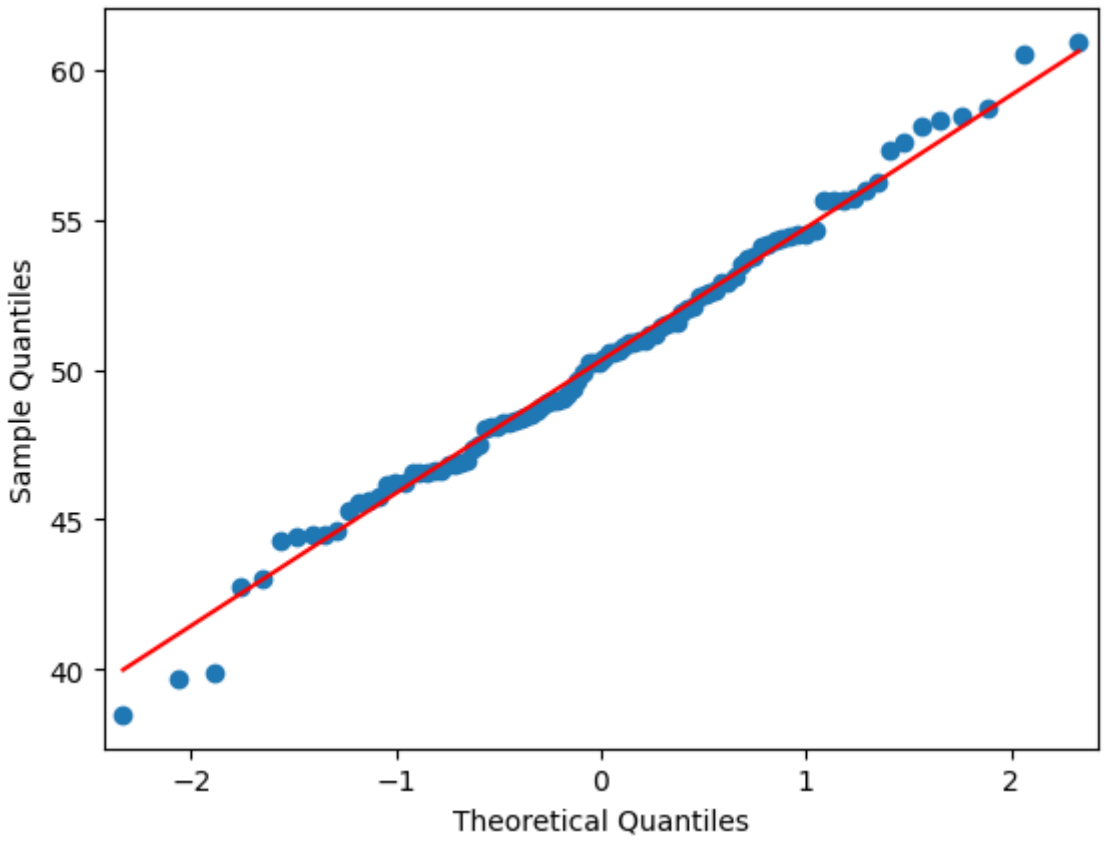
<u>Xây dựng box plot cho các bài tập sau:</u>

- **Trọng lượng của Coke thông thường và Coke ăn kiêng**. Sử dụng cùng một tỷ lệ để xây dựng box plot đối với trọng lượng của Coke thông thường và Coke ăn kiêng từ Dataset 17. Sử dụng box plot để so sánh hai bộ dữ liệu.
- **Trọng lượng của Coke thông thường và Pepsi thông thường**. Sử dụng cùng một tỷ lệ để xây dựng box plot cho trọng lượng của Coke thông thường và Pepsi thông thường từ Dataset 17. Sử dụng box plot để so sánh hai bộ dữ liệu.
- **Trọng lượng của phần tư vị**. Sử dụng cùng một tỷ lệ để xây dựng các box plot cho trọng lượng của các phần tư vị trước năm 1964 và phần tư vị sau năm 1964 từ Dataset 20. Sử dụng box plot để so sánh hai bộ dữ liệu.
- **Số lượng điện áp**. Sử dụng cùng một tỷ lệ để xây dựng các box plot cho lượng điện áp tại nhà và lượng điện áp máy phát từ Dataset 13. Sử dụng box plot để so sánh hai bộ dữ liệu.

6. Kiểm tra dạng chuẩn

Hướng dẫn:

```
#QQ plot
from numpy.random import seed
from numpy.random import randn
from statsmodels.graphics.gofplots import qqplot
from matplotlib import pyplot
#seed the random number generator
seed(1)
#generate univariate observations
data=5*randn(100)+50
#q-q plot
qqplot(data,line='s')
pyplot.show()
```



Vẽ QQ-plot trong các bài tập sau, xác định xem dữ liệu mẫu được lấy từ quần thể có phân phối chuẩn có phân phối chuẩn hay không.

- **Old Faithful:** biểu đồ QQ-plot biểu diễn thời gian (tính bằng giây) phun trào Old Faithful từ Dataset 15.
- **Chiều cao của phụ nữ:** biểu đồ QQ-plot biểu diễn chiều cao của phụ nữ từ dataset 1.
- **Trọng lượng của Coke ăn kiêng**: biểu đồ QQ-plot biểu diễn trọng lượng (tính bằng pound) của Diet Coke từ Dataset 17.