extra-2

October 2, 2023

[ ]:
```
import numpy as np
a = np.
  ↪array([[1,2,3,4,5],[6,7,8,9,10],[11,12,13,14,15],[16,17,18,19,20],[21,22,23,24,25]])
print(a)
print(a[2,1])
print(a[3,4])
print(a[:3,1:2])
print(a[-1])
print(a[-2:])
```

```
[[ 1  2  3  4  5]
 [ 6  7  8  9 10]
 [11 12 13 14 15]
 [16 17 18 19 20]
 [21 22 23 24 25]]
12
20
[[ 2]
 [ 7]
 [12]]
[21 22 23 24 25]
[[16 17 18 19 20]
 [21 22 23 24 25]]
```

[ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[ ]:
```
import numpy as np
a = np.array([[9,8,7],[6,5,4],[3,2,1]])

print(a)
print(a.min(axis=0))
print(a.min(axis=1))
print(a.max(axis=0))
print(a.max(axis=1))
print(a.sum(axis=0))
```

```
print(a.sum(axis=1))
print(a.argmin(axis=0))
print(a.argmax(axis=1))
print(a.mean(axis=0))
print(a.mean(axis=1))
print(np.sort(a,axis=0))
print(np.sort(a,axis=1))
```

```
[[9 8 7]
 [6 5 4]
 [3 2 1]]
[[3 2 1]
 [6 5 4]
 [9 8 7]]
[[7 8 9]
 [4 5 6]
 [1 2 3]]
```

[ ]:
```
import numpy as np
a = np.array([[9,8,7],[6,5,4],[3,2,1]])
b = np.array([[1,2,3],[4,5,6],[7,8,9]])

#print(a+b)
#print(a-b)
#print(a*b)
#print(a/b)
#print(np.power(a,b))
#print(np.remainder(a,b))
print(np.concatenate((a,b),axis=0))
#print(np.concatenate((a,b),axis=1))
```

```
[[9 8 7]
 [6 5 4]
 [3 2 1]
 [1 2 3]
 [4 5 6]
 [7 8 9]]
```

[ ]:
```
import re
n = input()
x = re.findall("[A-Za-z0-9]+@[A-Za-z]+\.(com|in)",n)
#"([A-Za-z0-9]+*[A-Za-z0-9]+@[A-Za-z]+(\.[A-Z|a-z])"
#Gaurav12@gmail.com
print(x)
if x:
    print("done")
else:
```

```
    print("not done")
```

```
Gaurav12@gmail.com
['com']
done
```

[ ]:
```
import re
n = input()
x = re.findall("^[Hh]...[Oo]$",n)
if x and len(n)==5:
    print("Done")
else:
    print("Not Done")
```

```
HELLO
Done
```

[ ]:
```
n = input("Enter String:")
x = input("Enter Word:")
l = list(n.split())
count = 0
for i in l:
    if i == x:
        count+=1
print(count)
```

```
Enter String:xyz
Enter Word:xyz
xyz
1
```

[ ]:
```
import re
n = input()
x = re.findall("^[A-Z].{0}[a-z]+@+.[0-9]$",n)

if x:
    print("done")
else:
    print("not done")
```

```
Ab@12
['Ab@12']
done
```

[ ]:
```
f = open("ABC.txt","R")
x=f.read()
print(x)
```

[ ]:
```
f = open("test.txt","r")
for i in range(5):
        x=f.readline()
        print(x)
```

[ ]:
```
f = open("abc.txt","r")
e = open("xyz.txt","a")
for i in f:
        i = f.readline()
        e.write(i)
z = open("xyz.txt","r")
x = z.read()
print(x)
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
Cell In[49], line 1
----> 1 f = open("abc.txt","r")
      2 e = open("xyz.txt","a")
      3 for i in f:

File ~/anaconda3/lib/python3.10/site-packages/IPython/core/interactiveshell.py:
 ↪282, in _modified_open(file, *args, **kwargs)
    275 if file in {0, 1, 2}:
    276     raise ValueError(
    277         f"IPython won't let you open fd={file} by default "
    278         "as it is likely to crash IPython. If you know what you are␣
 ↪doing, "
    279         "you can use builtins' open."
    280     )
--> 282 return io_open(file, *args, **kwargs)

FileNotFoundError: [Errno 2] No such file or directory: 'abc.txt'
```

[ ]:
```
import numpy as np
a = np.random.randint(0,9,9)
b = a.reshape((3,3))
print(b)
c= np.eye(3)
print(c)
```

```
[[5 4 7]
 [0 6 4]
 [2 0 0]]
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

[ ]:
```
import numpy as np
a = np.random.rand()
print(a)
```

```
0.11553158811724928
```

[ ]:
```
import numpy as np
a = np.
  ↪array([[26,27,28,29,30],[31,32,33,34,35],[36,37,38,39,40],[41,42,43,44,45],[46,47,48,49,50]])
print(a)
print(a[2,1])
print(a[3,4])
print(a[:3,1:2])
print(a[-1])
print(a[-2:])
for i in a:
    for j in i:
        if j % 3 ==0 and j% 5!= 0:
            print(j)
```

```
[[26 27 28 29 30]
 [31 32 33 34 35]
 [36 37 38 39 40]
 [41 42 43 44 45]
 [46 47 48 49 50]]
37
45
[[27]
 [32]
 [37]]
[46 47 48 49 50]
[[41 42 43 44 45]
 [46 47 48 49 50]]
27
33
36
39
42
48
```

[ ]:
```
wap TO CREATE AN WITHCH HOLDS 25 INTEGER VALUES aND pERFORM THE FOLLOWING:
1)Split the array into 10,2,20 and 30 parts.(np.array_split(n,10))
2)Get indices of an arry whose values are even.
3)Get indices of an arry whose values are odd.
4)Get integer numbers less then 15 from the array.
```

[ ]:
```
import pandas as pd
a = [1,2,3,4,5,6,7,8,9,10]
n = pd.Series(a)
print(n[1])
print(n[3])
print(n[5])
print(n[7])
```

```
2
4
6
8
```

[ ]:
```
import pandas as pd
a = ['Computer','civil','Mechanical']
n = pd.Series(a)
print(n[0])
print(n[2])
```

```
Computer
Mechanical
```

[ ]:
```
import pandas as pd
Name = input("Enter Name:")
Surname = input("Enter Surame:")
City = input("Enter City:")
State = input("Enter State:")
Country = input("Enter Country:")
a = [Name,Surname,City,State,Country]
i = ['Name','Surname','City','State','Country']
n = pd.Series(a,i)
print(n)
```

```
Enter Name:a
Enter Surame:b
Enter City:c
Enter State:d
Enter Country:e
Name       a
Surname    b
City       c
State      d
```

```
Country    e
dtype: object
```

[ ]:
```python
import pandas as pd
import numpy as np
subjact = ['Advance Python','Android','UI/UX','IOT']
mark = np.random.randint(70,100,40).reshape((10,4))
roll_no = np.arange(101,111)
n = pd.DataFrame(mark,roll_no,subjact)
print(n)
```

```
     Advance Python  Android  UI/UX  IOT
101              98       73     70   84
102              87       79     83   72
103              92       74     89   73
104              89       85     85   87
105              90       83     87   85
106              71       75     70   83
107              74       94     99   90
108              88       98     87   95
109              70       96     95   98
110              99       85     91   70
```

[ ]:
```python
import pandas as pd
import numpy as np
subject = ['Advance Python','Android','UI/UX','IOT']
mark = np.random.randint(70,100,40).reshape((10,4))
roll_no = np.arange(101,111)
n = pd.DataFrame(mark,roll_no,subject)

print(n.loc[range(101,111,2)])

print(n.loc[range(101,106),subject[0:3]])

n['Total'] = n['Advance Python']+n['Android']+n['UI/UX']+n['IOT']
print(n)

n.drop(109,axis=0,inplace=True)
n.drop(110,axis=0,inplace=True)
print(n)

n.drop(['Total'],axis=1,inplace=True)
print(n)

print(n > 75)

print(n['IOT'] < 75)
```

```python
n.index.name="Roll_no"
print(n)

n.set_index("IOT",inplace=True)
print(n)

n.reset_index(inplace=True)
print(n)
```

```
     Advance Python  Android  UI/UX  IOT
101              75       75     95   74
103              82       94     85   83
105              84       94     86   99
107              96       82     84   96
109              96       81     97   88
     Advance Python  Android  UI/UX
101              75       75     95
102              78       81     71
103              82       94     85
104              73       73     75
105              84       94     86
     Advance Python  Android  UI/UX  IOT  Total
101              75       75     95   74    319
102              78       81     71   83    313
103              82       94     85   83    344
104              73       73     75   87    308
105              84       94     86   99    363
106              76       86     71   92    325
107              96       82     84   96    358
108              88       81     98   72    339
109              96       81     97   88    362
110              80       86     73   79    318
     Advance Python  Android  UI/UX  IOT  Total
101              75       75     95   74    319
102              78       81     71   83    313
103              82       94     85   83    344
104              73       73     75   87    308
105              84       94     86   99    363
106              76       86     71   92    325
107              96       82     84   96    358
108              88       81     98   72    339
     Advance Python  Android  UI/UX  IOT
101              75       75     95   74
102              78       81     71   83
103              82       94     85   83
104              73       73     75   87
```

```
105              84       94     86   99
106              76       86     71   92
107              96       82     84   96
108              88       81     98   72
     Advance Python  Android  UI/UX    IOT
101           False    False   True  False
102            True     True  False   True
103            True     True   True   True
104           False    False  False   True
105            True     True   True   True
106            True     True  False   True
107            True     True   True   True
108            True     True   True  False
101     True
102    False
103    False
104    False
105    False
106    False
107    False
108     True
Name: IOT, dtype: bool
         Advance Python  Android  UI/UX  IOT
Roll_no
101                  75       75     95   74
102                  78       81     71   83
103                  82       94     85   83
104                  73       73     75   87
105                  84       94     86   99
106                  76       86     71   92
107                  96       82     84   96
108                  88       81     98   72
     Advance Python  Android  UI/UX
IOT
74               75       75     95
83               78       81     71
83               82       94     85
87               73       73     75
99               84       94     86
92               76       86     71
96               96       82     84
72               88       81     98
   IOT  Advance Python  Android  UI/UX
0   74              75       75     95
1   83              78       81     71
2   83              82       94     85
3   87              73       73     75
4   99              84       94     86
```

```
5   92              76       86      71
6   96              96       82      84
7   72              88       81      98
```

[ ]:
```
import pandas as pd


n = pd.read_excel("studentdata.xlsx")

print(n)
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-26-431ae81eb478> in <cell line: 3>()
      1 import pandas as pd
      2
----> 3 n = pd.read_excel("studentdata.xlsx")
      4
      5 print(n)

/usr/local/lib/python3.10/dist-packages/pandas/util/_decorators.py in␣
 ↪wrapper(*args, **kwargs)
    209                 else:
    210                     kwargs[new_arg_name] = new_arg_value
--> 211             return func(*args, **kwargs)
    212
    213         return cast(F, wrapper)

/usr/local/lib/python3.10/dist-packages/pandas/util/_decorators.py in␣
 ↪wrapper(*args, **kwargs)
    329                     stacklevel=find_stack_level(),
    330                 )
--> 331             return func(*args, **kwargs)
    332
    333         # error: "Callable[[VarArg(Any), KwArg(Any)], Any]" has no

/usr/local/lib/python3.10/dist-packages/pandas/io/excel/_base.py in␣
 ↪read_excel(io, sheet_name, header, names, index_col, usecols, squeeze, dtype,␣
 ↪engine, converters, true_values, false_values, skiprows, nrows, na_values,␣
 ↪keep_default_na, na_filter, verbose, parse_dates, date_parser, thousands,␣
 ↪decimal, comment, skipfooter, convert_float, mangle_dupe_cols, storage_option
    480     if not isinstance(io, ExcelFile):
    481         should_close = True
--> 482         io = ExcelFile(io, storage_options=storage_options,␣
 ↪engine=engine)
    483     elif engine and engine != io.engine:
    484         raise ValueError(
```

```
/usr/local/lib/python3.10/dist-packages/pandas/io/excel/_base.py in␣
 ↪__init__(self, path_or_buffer, engine, storage_options)
   1650                 ext = "xls"
   1651             else:
-> 1652                 ext = inspect_excel_format(
   1653                     content_or_path=path_or_buffer,␣
 ↪storage_options=storage_options
   1654                 )

/usr/local/lib/python3.10/dist-packages/pandas/io/excel/_base.py in␣
 ↪inspect_excel_format(content_or_path, storage_options)
   1523         content_or_path = BytesIO(content_or_path)
   1524
-> 1525     with get_handle(
   1526         content_or_path, "rb", storage_options=storage_options,␣
 ↪is_text=False
   1527     ) as handle:

/usr/local/lib/python3.10/dist-packages/pandas/io/common.py in␣
 ↪get_handle(path_or_buf, mode, encoding, compression, memory_map, is_text,␣
 ↪errors, storage_options)
    863         else:
    864             # Binary mode
--> 865             handle = open(handle, ioargs.mode)
    866         handles.append(handle)
    867

FileNotFoundError: [Errno 2] No such file or directory: 'studentdata.xlsx'
```

[ ]:

Mounted at ./studentdata.xlsm

[14]:
```
import pandas as pd

path="/content/drive/MyDrive/Python/student_data.xlsx"
df = pd.read_excel(path)

#1
print("SPI mean = ",df['SPI'].mean())
print("SPI sum = ",df['SPI'].sum())
print("SPI min = ",df['SPI'].min())
print("SPI max = ",df['SPI'].max())

print()

#2
```

```
print("University =",df.groupby('University').groups)

print()

#3
print("University & Branch =",df.groupby(['University','Branch']).groups)

print()

#4
for i,j in df.groupby(['University']):
  print(j)
  print()
print()

#5
for i,j in df.groupby(['University']):
  print(i)
  print()
print()

#6
for i,j in df.groupby(['University']):
  print(i)
  print(j)
  print()
print()

#7
for i,j in df.groupby(['University']):
  print()
  print("Mean")
  print(j.mean())
  print()
  print("Sum")
  print(j.sum())
  print()
  print("Max")
  print(j.max())
  print()
  print("Min")
  print(j.min())
  print()
  print()
print()

#8
```

```
print("Mean Value of University & Branch")
for i,j in df.groupby(['University','Branch']):
  print(j.mean())
  print()
print()

#9
print("Mean Value of University & Branch & SPI")
for i,j in df.groupby(['University','Branch','SPI']):
  print(j.mean())
  print()
print()
```

```
SPI mean =  8.2
SPI sum =  41.0
SPI min =  7.0
SPI max =  9.0

University = {'DU': [0, 1], 'MU': [2, 3], 'RKU': [5]}

University & Branch = {('DU', 'CE'): [0], ('DU', 'EE'): [1], ('MU', 'ME'): [2,
3], ('RKU', 'CE'): [5], (nan, nan): [4]}

    Enrollment Name University Branch  Semester  SPI
0  200540001.0  abc         DU     CE       4.0  8.0
1  200540002.0  def         DU     EE       4.0  9.0

    Enrollment Name University Branch  Semester  SPI
2  200540003.0  ghi         MU     ME       5.0  7.0
3  200540004.0  jkl         MU     ME       5.0  8.0

    Enrollment Name University Branch  Semester  SPI
5  200540006.0  pqr        RKU     CE       5.0  9.0


DU

MU

RKU


DU
    Enrollment Name University Branch  Semester  SPI
0  200540001.0  abc         DU     CE       4.0  8.0
1  200540002.0  def         DU     EE       4.0  9.0
```

```
MU
    Enrollment Name University Branch  Semester  SPI
2  200540003.0  ghi         MU     ME       5.0  7.0
3  200540004.0  jkl         MU     ME       5.0  8.0

RKU
    Enrollment Name University Branch  Semester  SPI
5  200540006.0  pqr        RKU     CE       5.0  9.0



Mean
Enrollment    200540001.5
Semester              4.0
SPI                   8.5
dtype: float64

Sum
Enrollment    401080003.0
Name               abcdef
University           DUDU
Branch               CEEE
Semester              8.0
SPI                  17.0
dtype: object

Max
Enrollment    200540002.0
Name                  def
University             DU
Branch                 EE
Semester              4.0
SPI                   9.0
dtype: object

Min
Enrollment    200540001.0
Name                  abc
University             DU
Branch                 CE
Semester              4.0
SPI                   8.0
dtype: object



Mean
Enrollment    200540003.5
```

```
Semester              5.0
SPI                   7.5
dtype: float64

Sum
Enrollment    401080007.0
Name                ghijkl
University            MUMU
Branch                MEME
Semester              10.0
SPI                   15.0
dtype: object

Max
Enrollment    200540004.0
Name                  jkl
University             MU
Branch                 ME
Semester              5.0
SPI                   8.0
dtype: object

Min
Enrollment    200540003.0
Name                  ghi
University             MU
Branch                 ME
Semester              5.0
SPI                   7.0
dtype: object



Mean
Enrollment    200540006.0
Semester              5.0
SPI                   9.0
dtype: float64

Sum
Enrollment    200540006.0
Name                  pqr
University            RKU
Branch                 CE
Semester              5.0
SPI                   9.0
dtype: object
```

```
Max
Enrollment    200540006.0
Name                  pqr
University            RKU
Branch                 CE
Semester              5.0
SPI                   9.0
dtype: object

Min
Enrollment    200540006.0
Name                  pqr
University            RKU
Branch                 CE
Semester              5.0
SPI                   9.0
dtype: object



Mean Value of University & Branch
Enrollment    200540001.0
Semester              4.0
SPI                   8.0
dtype: float64

Enrollment    200540002.0
Semester              4.0
SPI                   9.0
dtype: float64

Enrollment    200540003.5
Semester              5.0
SPI                   7.5
dtype: float64

Enrollment    200540006.0
Semester              5.0
SPI                   9.0
dtype: float64


Mean Value of University & Branch & SPI
Enrollment    200540001.0
Semester              4.0
SPI                   8.0
dtype: float64
```

```
Enrollment    200540002.0
Semester              4.0
SPI                   9.0
dtype: float64

Enrollment    200540003.0
Semester              5.0
SPI                   7.0
dtype: float64

Enrollment    200540004.0
Semester              5.0
SPI                   8.0
dtype: float64

Enrollment    200540006.0
Semester              5.0
SPI                   9.0
dtype: float64
```

```
<ipython-input-14-a4cff3b03eee>:25: FutureWarning: In a future version of
pandas, a length 1 tuple will be returned when iterating over a groupby with a
grouper equal to a list of length 1. Don't supply a list with a single grouper
to avoid this warning.
  for i,j in df.groupby(['University']):
<ipython-input-14-a4cff3b03eee>:31: FutureWarning: In a future version of
pandas, a length 1 tuple will be returned when iterating over a groupby with a
grouper equal to a list of length 1. Don't supply a list with a single grouper
to avoid this warning.
  for i,j in df.groupby(['University']):
<ipython-input-14-a4cff3b03eee>:37: FutureWarning: In a future version of
pandas, a length 1 tuple will be returned when iterating over a groupby with a
grouper equal to a list of length 1. Don't supply a list with a single grouper
to avoid this warning.
  for i,j in df.groupby(['University']):
<ipython-input-14-a4cff3b03eee>:44: FutureWarning: In a future version of
pandas, a length 1 tuple will be returned when iterating over a groupby with a
grouper equal to a list of length 1. Don't supply a list with a single grouper
to avoid this warning.
  for i,j in df.groupby(['University']):
<ipython-input-14-a4cff3b03eee>:47: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(j.mean())
```

```
<ipython-input-14-a4cff3b03eee>:47: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(j.mean())
<ipython-input-14-a4cff3b03eee>:47: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(j.mean())
<ipython-input-14-a4cff3b03eee>:64: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(j.mean())
<ipython-input-14-a4cff3b03eee>:64: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(j.mean())
<ipython-input-14-a4cff3b03eee>:64: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(j.mean())
<ipython-input-14-a4cff3b03eee>:64: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(j.mean())
<ipython-input-14-a4cff3b03eee>:71: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(j.mean())
<ipython-input-14-a4cff3b03eee>:71: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(j.mean())
```

```
<ipython-input-14-a4cff3b03eee>:71: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(j.mean())
<ipython-input-14-a4cff3b03eee>:71: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(j.mean())
<ipython-input-14-a4cff3b03eee>:71: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(j.mean())
```

[ ]:

```python
import pandas as pd

CSV1="/content/drive/MyDrive/Python/dcx_marks.csv"
CSV2="/content/drive/MyDrive/Python/dcy_marks.csv"
CSV3="/content/drive/MyDrive/Python/dcz_marks.csv"

x = pd.read_csv(CSV1)
y = pd.read_csv(CSV2)
z = pd.read_csv(CSV3)

df = pd.concat([x,y,z], axis=0)
print(df)
```

```
   Roll No  Advance Python  Android  UI/UX  IOT
0      101              71       61     51   41
1      102              72       62     52   42
2      103              73       63     53   43
3      104              74       64     54   44
4      105              75       65     55   45
5      106              71       61     51   41
6      107              72       62     52   42
7      108              73       63     53   43
8      109              74       64     54   44
9      110              75       65     55   45
0      201              71       61     51   41
1      202              72       62     52   42
2      203              73       63     53   43
3      204              74       64     54   44
4      205              75       65     55   45
```

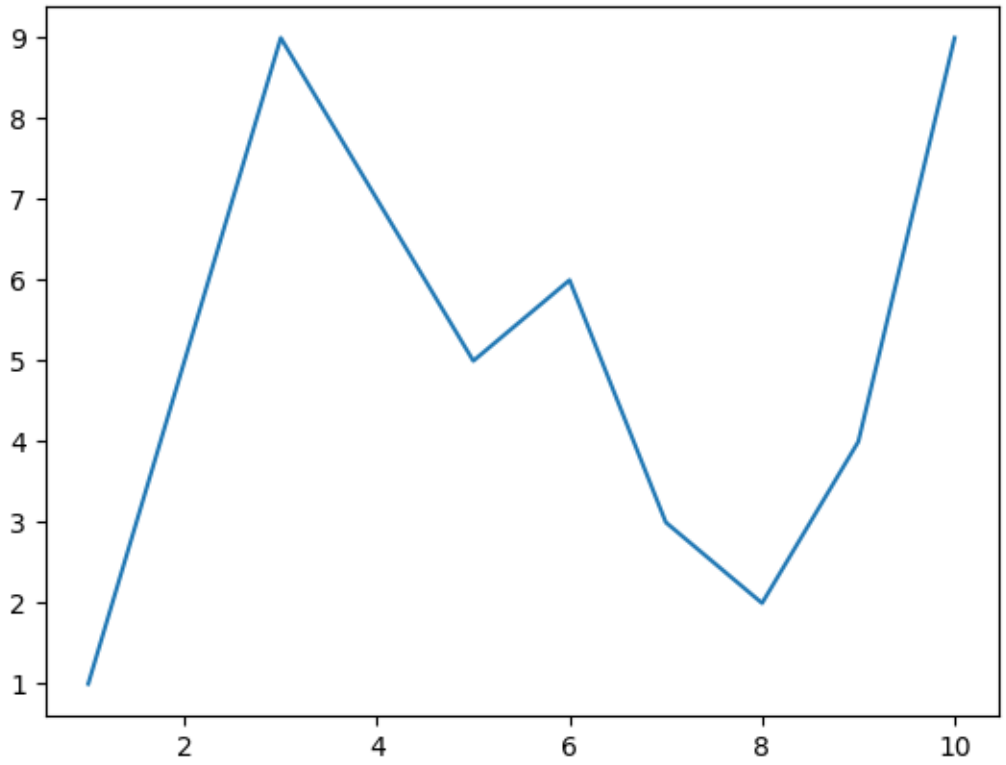
```
5      206              71       61      51    41
6      207              72       62      52    42
7      208              73       63      53    43
8      209              74       64      54    44
9      210              75       65      55    45
0      301              71       61      51    41
1      302              72       62      52    42
2      303              73       63      53    43
3      304              74       64      54    44
4      305              75       65      55    45
5      306              71       61      51    41
6      307              72       62      52    42
7      308              73       63      53    43
8      309              74       64      54    44
9      310              75       65      55    45
```

[ ]:
```
import pandas as pd

CSV1="/content/drive/MyDrive/Python/dcx_marks.csv"
CSV2="/content/drive/MyDrive/Python/dcy_marks.csv"
CSV3="/content/drive/MyDrive/Python/dcz_marks.csv"

x = pd.read_csv(CSV1)
y = pd.read_csv(CSV2)
z = pd.read_csv(CSV3)

df = pd.concat([x,y,z], axis=1)
print(df)
```

```
   Roll No  Advance Python  Android  UI/UX  IOT  Roll No  Advance Python  \
0      101              71       61      51   41      201              71
1      102              72       62      52   42      202              72
2      103              73       63      53   43      203              73
3      104              74       64      54   44      204              74
4      105              75       65      55   45      205              75
5      106              71       61      51   41      206              71
6      107              72       62      52   42      207              72
7      108              73       63      53   43      208              73
8      109              74       64      54   44      209              74
9      110              75       65      55   45      210              75

   Android  UI/UX  IOT  Roll No  Advance Python  Android  UI/UX  IOT
0       61     51   41      301              71       61     51   41
1       62     52   42      302              72       62     52   42
2       63     53   43      303              73       63     53   43
3       64     54   44      304              74       64     54   44
4       65     55   45      305              75       65     55   45
5       61     51   41      306              71       61     51   41
```

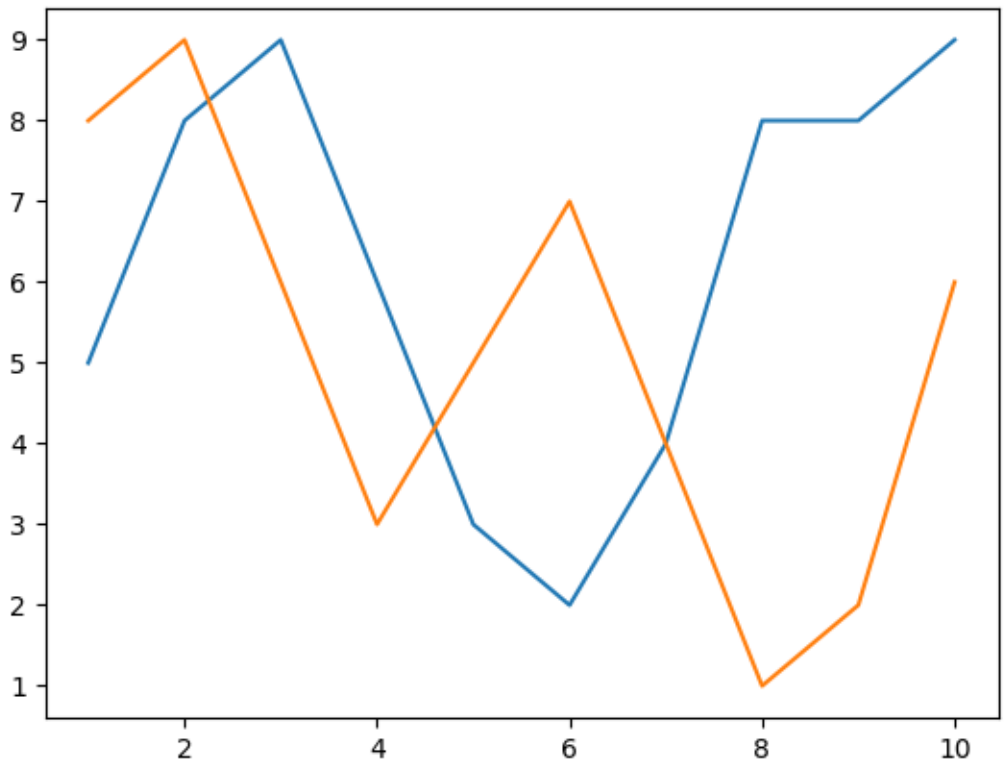
```
6       62      52   42      307             72      62      52   42
7       63      53   43      308             73      63      53   43
8       64      54   44      309             74      64      54   44
9       65      55   45      310             75      65      55   45
```

[ ]:
```
#1
import matplotlib.pyplot as plt
x = range(1,11)
y = [1,5,9,7,5,6,3,2,4,9]
plt.plot(x,y)
plt.show()
```



[2]:
```
import matplotlib.pyplot as plt
x = [1,2,3,4,5,6,7,8,9,10]
value1 = [5,8,9,6,3,2,4,8,8,9]
value2 = [8,9,6,3,5,7,4,1,2,6]

#1
plt.plot(x,value1)
plt.plot(x,value2)

#2
```

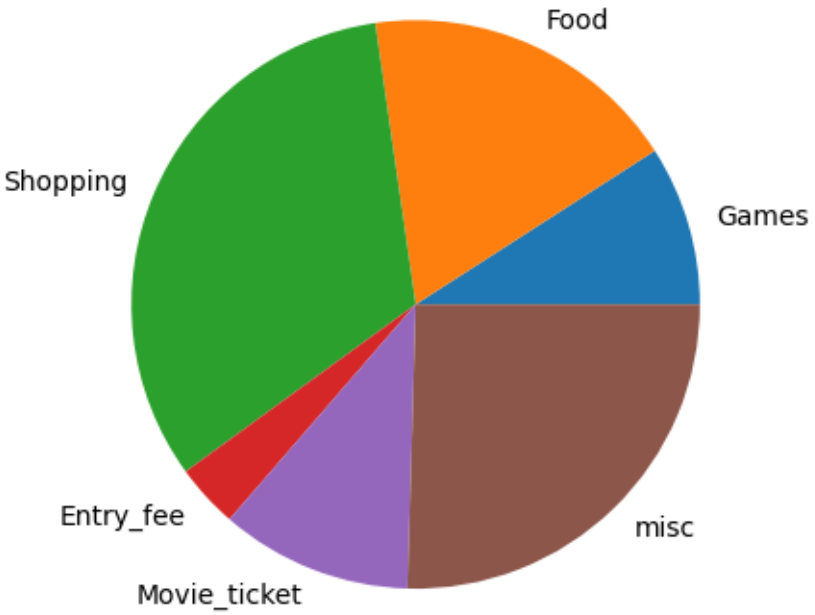
```
plt.plot(x,value1,ls="--")
plt.plot(x,value2,ls=":")

#3
plt.plot(x,value1,ls="-.",c="pink")
plt.plot(x,value2,c="green")

#4
plt.plot(x,value1,lw=2)
plt.plot(x,value2,lw=5)

#5
plt.plot(x,value1,marker=">")
plt.plot(x,value2,marker="o")

#6
plt.xlabel("x")
plt.ylabel("Y")
plt.annotate(xy=[3,7],text='Highest')
plt.legend(['X','Y'])

plt.show()
```



[2]:
```
from google.colab import drive
drive.mount('/content/drive')
```

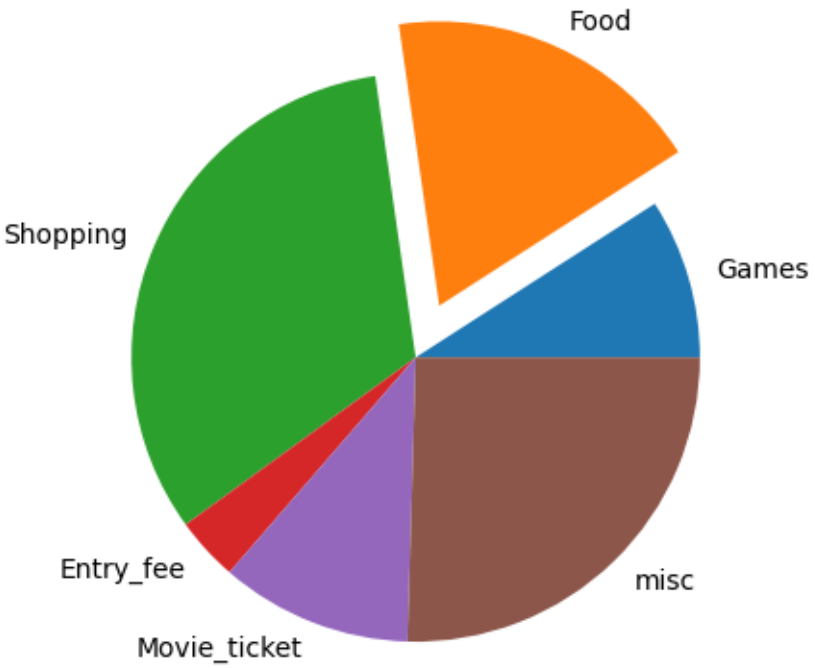
```
Mounted at /content/drive
```

[ ]:
```
import matplotlib.pyplot as plt
x = ([100,200,360,40,120,280])
y = ['Games','Food','Shopping','Entry_fee','Movie_ticket','misc']

plt.pie(x, labels = y)
plt.show()
```



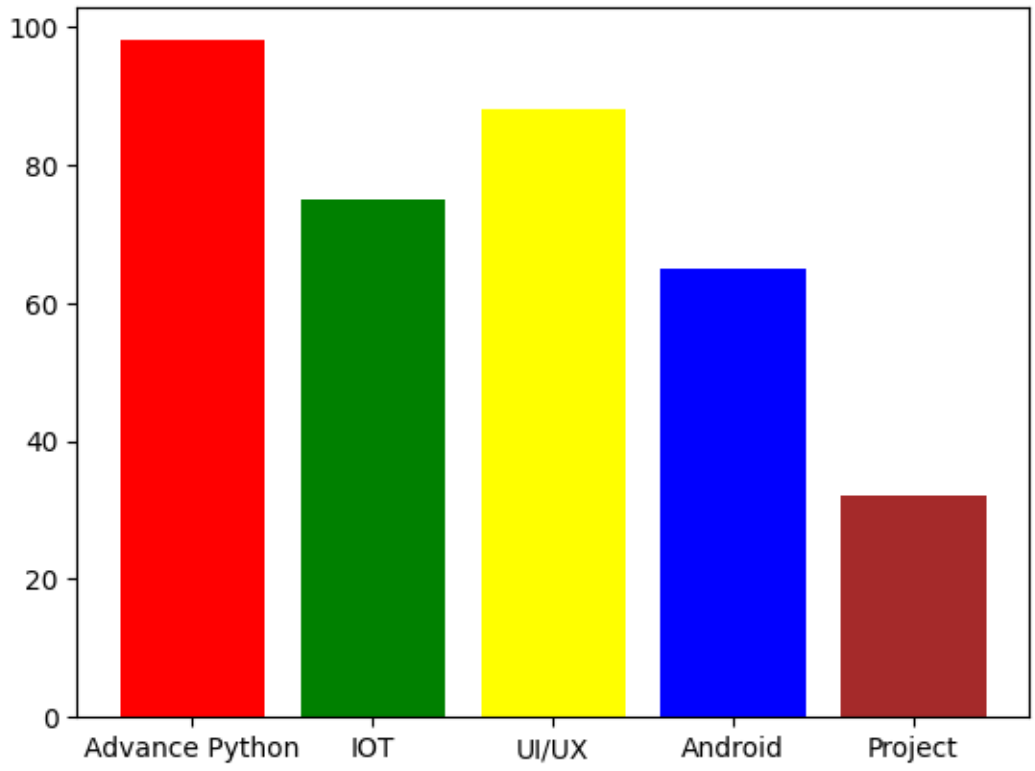
[4]:
```
import matplotlib.pyplot as plt
x = ([100,200,360,40,120,280])
y = ['Games','Food','Shopping','Entry_fee','Movie_ticket','misc']

#1
explode = [0,0.2,0,0,0,0]
plt.pie(x, labels = y, explode = explode)

#2
c = ['red','black','green','yellow','blue','brown']
plt.pie(x, labels = y, colors = c)
```

```
#3
plt.pie(x, labels = y, autopct='%1.1f%%')

#4.1
plt.pie(x, labels = y, autopct='%1.3f%%')

#4.2
plt.pie(x, labels = y, autopct='%1.2f%%')

#4.3
plt.pie(x, labels = y, autopct='%1f%%')

#5
explode = [0,0.2,0,0,0,0]
plt.pie(x, labels = y, explode = explode, shadow = True)

plt.show()
```



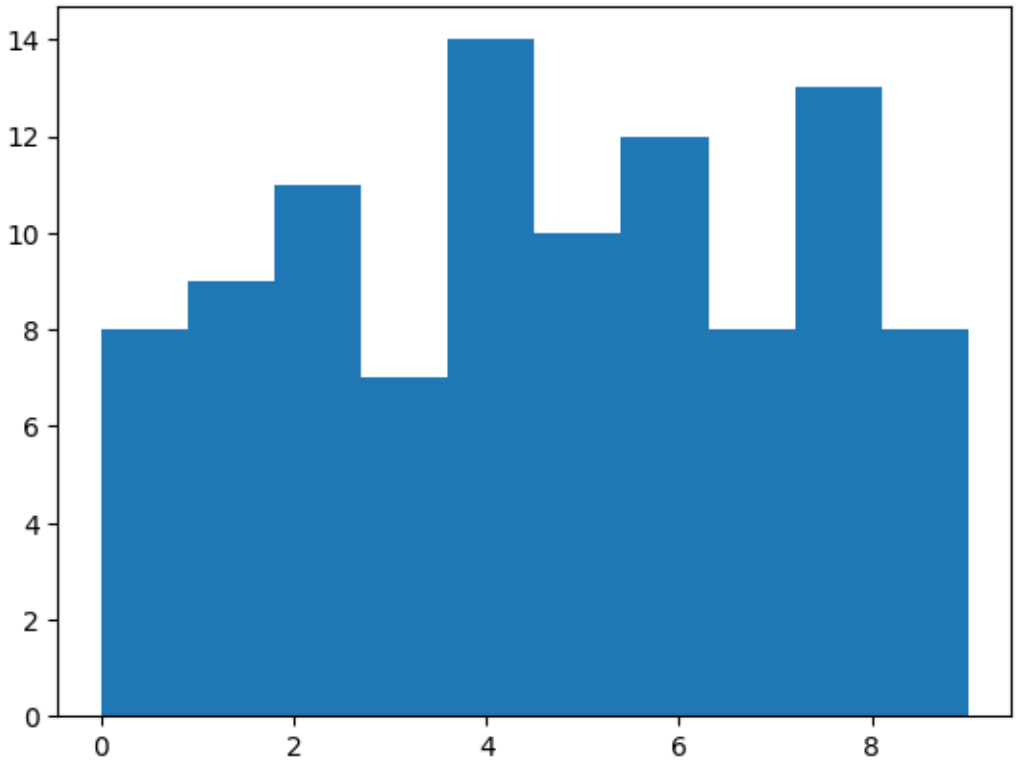
[ ]:
```
import matplotlib.pyplot as plt
y = [98,75,88,65,32]
x = ["Advance Python",'IOT','UI/UX','Android','Project']
```

```
#1
c = ['red','green','yellow','blue','brown']
plt.bar(x,y, color=c)

#2
plt.title("Result")

#3
plt.bar(x,y, color=c, width=0.5)

#4
w = [0.2,0.4,0.6,0.8,1]
plt.bar(x,y, color=c, width=w)

#5
plt.barh(x,y,linewidth=1)

plt.show()
```

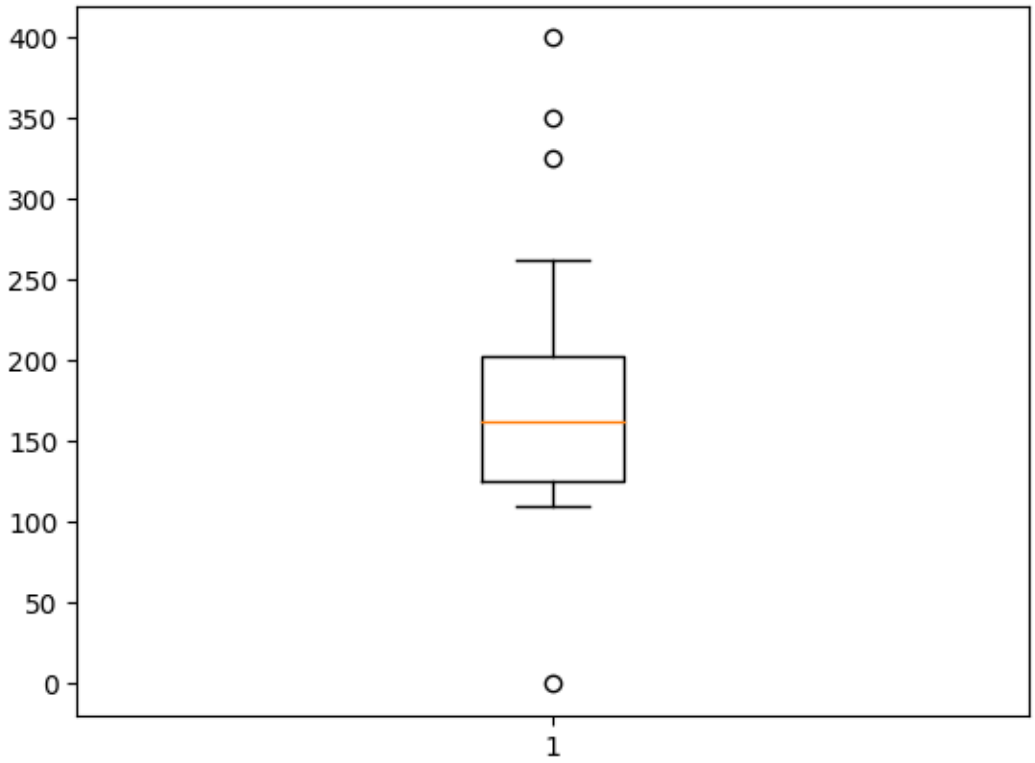


[ ]:
```
import matplotlib.pyplot as plt
import numpy as np
x = np.random.randint(0,10,100)
```

```
plt.hist(x)
plt.show()
```

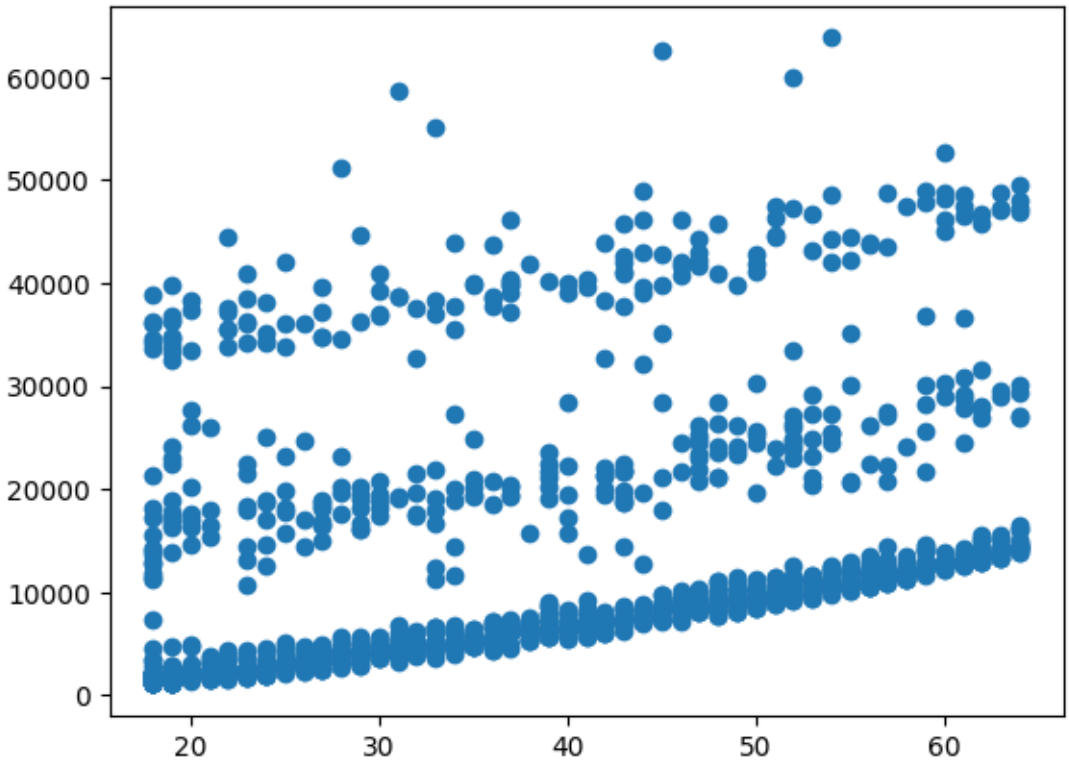


[ ]:
```
import matplotlib.pyplot as plt
x = [120, 110, 130, 125, 112, 0, 142, 162, 203, 350, 325, 200, 153, 250, 120,␣
  ↪153, 168, 198, 185, 400, 262]
plt.boxplot(x)
plt.show()
```



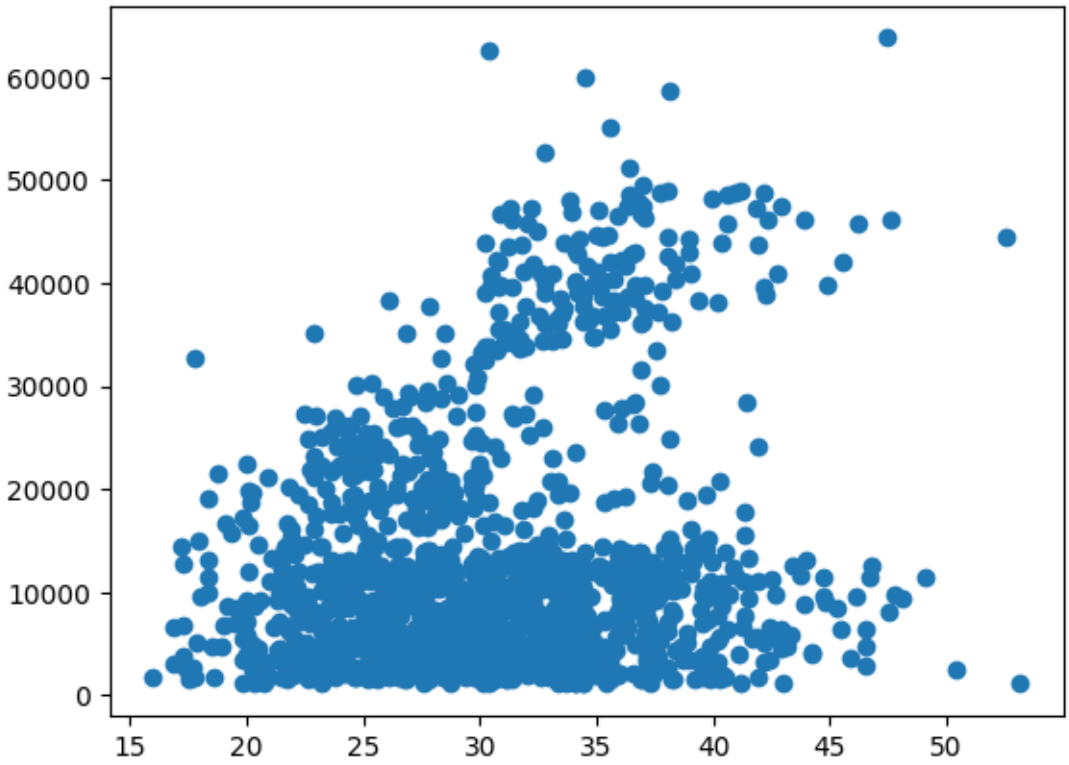
[5]:
```
import matplotlib.pyplot as plt
import pandas as pd

CSV="/content/drive/MyDrive/Python/insurance.csv"

x = pd.read_csv(CSV)

plt.scatter(x['age'],x['charges'])
```

[5]: <matplotlib.collections.PathCollection at 0x7d5cfcb089a0>



[6]:
```
import matplotlib.pyplot as plt
import pandas as pd

CSV="/content/drive/MyDrive/Python/insurance.csv"

x = pd.read_csv(CSV)

plt.scatter(x['bmi'],x['charges'])
```

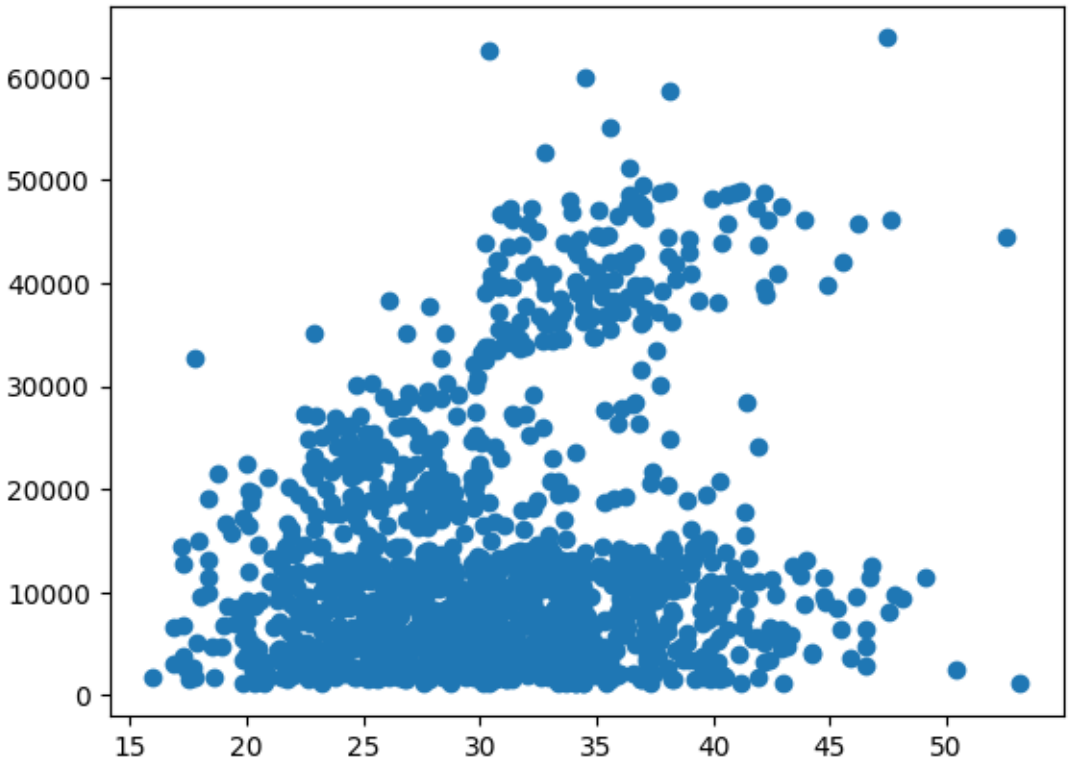
[6]: <matplotlib.collections.PathCollection at 0x7d5cfaa31fc0>



[21]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

CSV="/content/drive/MyDrive/Python/insurance.csv"

x = pd.read_csv(CSV)
y = np.array(x.groupby(['smoker']))
plt.scatter(x['bmi'],x['charges'])
```

```
<ipython-input-21-5f6f7760146a>:8: FutureWarning: In a future version of pandas,
a length 1 tuple will be returned when iterating over a groupby with a grouper
equal to a list of length 1. Don't supply a list with a single grouper to avoid
this warning.
  y = np.array(x.groupby(['smoker']))
<ipython-input-21-5f6f7760146a>:8: VisibleDeprecationWarning: Creating an
ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-
tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant
to do this, you must specify 'dtype=object' when creating the ndarray.
  y = np.array(x.groupby(['smoker']))
```

[21]: `<matplotlib.collections.PathCollection at 0x7d5cfa56df30>`


60000
50000
40000
30000
20000
10000
0
15
20
25
30
35
40
45
50

[ ]: