

Ex no 4 Text File and JSON

Aim :

To open, read and write the text files and basic JSON operation.

Description:

1. Open a text file in write mode.
2. Write the content and close the file.
3. Open the same text file in read mode and read the contents.
4. Read the contents through the read function readline() function.
5. Add some content to the already created file.
6. Display the file content through the read function and close the file.
7. Use basic JSON loads and dumps functions.
8. Data Cleaning and Preparation
9. Data Wrangling
10. Data Visualization
11. Time series

#1

```
a=open("textfile.txt", "w")
a.close()
```

#2

```
a=open("textfile.txt", "w")
a.write("python programming for machine learning.")
a.close()
```

#3

```
a=open("textfile.txt", "r")
c=a.read()
print(c)
a.close()
```

#4

```
a=open("textfile.txt", "r")
line=a.readline()
print(line)
a.close()
```

#5

```
a = open("textfile.txt", "a")
a.write("python programming for machine learning.PPML is a laboratory course.")
a.close()
```

#6

```
a = open("textfile.txt", "r")
contents = a.read()
print(contents)
a.close()
```

o/p:

python programming for machine learning.PPML is a laboratory course.

#7

```
import json
data = {
    "name": "John Doe",
    "age": 30,
    "city": "New York"
}

json_string = json.dumps(data)
print(json_string)
python_object = json.loads(json_string)
print(python_object)
```

```
o/p:
{"name": "John Doe", "age": 30, "city": "New York"}
{'name': 'John Doe', 'age': 30, 'city': 'New York'}
```

#8 data cleaning and preparation

```
import pandas as pd
import numpy as np
df=pd.DataFrame(np.random.randn(5,3),index=['a','c','e','f','h'],columns=['one','two',
'three'])
print(df)
```

o/p:

	one	two	three
a	0.151651	-0.585318	0.477398
c	-1.126654	0.809996	0.215782
e	-1.536315	0.284351	-0.208342
f	-0.126283	-1.630998	1.898356
h	1.749104	0.440270	-0.441704

a) Re indexing

```
df=df.reindex(['a','b','c','d','e','f','g','h'])
print(df)
```

	one	two	three
a	0.151651	-0.585318	0.477398
b	NaN	NaN	NaN
c	-1.126654	0.809996	0.215782
d	NaN	NaN	NaN
e	-1.536315	0.284351	-0.208342
f	-0.126283	-1.630998	1.898356
g	NaN	NaN	NaN
h	1.749104	0.440270	-0.441704

b) Dropna

```
df1=df.dropna()
print(df1)
```

	one	two	three
a	0.151651	-0.585318	0.477398
c	-1.126654	0.809996	0.215782
e	-1.536315	0.284351	-0.208342
f	-0.126283	-1.630998	1.898356
h	1.749104	0.440270	-0.441704

c) Fillna

```
df2=df1
print(df1.fillna(0))
```

	one	two	three
a	0.151651	-0.585318	0.477398
c	-1.126654	0.809996	0.215782
e	-1.536315	0.284351	-0.208342
f	-0.126283	-1.630998	1.898356
h	1.749104	0.440270	-0.441704

d)

```
df3=df2
print(df2.fillna(method='pad'))
```

	one	two	three
a	0.151651	-0.585318	0.477398
c	-1.126654	0.809996	0.215782
e	-1.536315	0.284351	-0.208342
f	-0.126283	-1.630998	1.898356
h	1.749104	0.440270	-0.441704

e)

```
df4=df3
print(df3.fillna(method='bfill'))
```

	one	two	three
a	0.151651	-0.585318	0.477398
c	-1.126654	0.809996	0.215782
e	-1.536315	0.284351	-0.208342
f	-0.126283	-1.630998	1.898356
h	1.749104	0.440270	-0.441704

e) Is null

```
print(df['two'].isnull())
```

```
a    False
b     True
c    False
d     True
e    False
f    False
g     True
h    False
Name: two, dtype: bool
```

f)

```
print(df['three'].dropna())
```

```
a    0.477398
c    0.215782
e   -0.208342
f    1.898356
h   -0.441704
Name: three, dtype: float64
```

#9 data wrangling

```
import numpy as np
import pandas as pd
df=pd.DataFrame([[ 'dog',67],[ 'cat',58],[ 'horse',80],[ 'horse',78],[ 'deer',90],[ 'deer', '96']],index=[1,2,3,4,5,6],columns=[ 'animal', 'speed'])
print(df)
```

	animal	speed
1	dog	67
2	cat	58
3	horse	80
4	horse	78
5	deer	90
6	deer	96

a) Mean

```
a=df.groupby(['animal']).mean()
print(a)
```

	speed
animal	
cat	58.0
deer	92.5
dog	67.0
horse	81.0

b) Sum

```
b=df.groupby(['animal']).sum()
print(b)
```

	speed
animal	
cat	58
deer	185
dog	67
horse	162

c) Count

```
c=df.groupby(['animal']).count()
print(c)
```

	speed
animal	
cat	1
deer	2
dog	1
horse	2

d) First()

```
d=df.groupby(['animal']).first()
print(d)
```

	speed
animal	
cat	58
deer	90
dog	67
horse	86

e) Last()

```
e=df.groupby(['animal']).last()
print(e)
```

	speed
animal	
cat	58
deer	95
dog	67
horse	76

f) Size()

```
f=df.groupby(['animal']).size()
print(f)
```

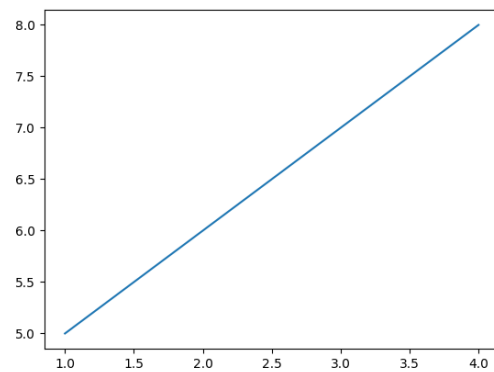
animal	
cat	1
deer	2
dog	1
horse	2

dtype: int64

#10 data visualisation

a) Line graph

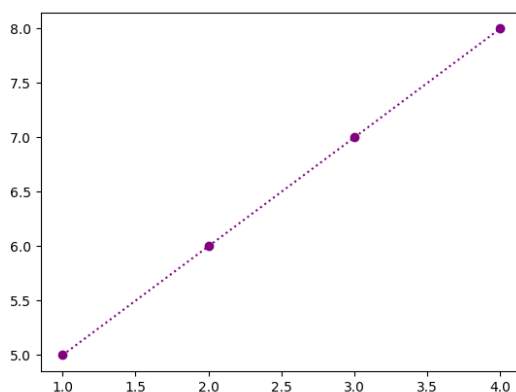
```
import numpy as np
import matplotlib.pyplot as plt
a=np.array([1,2,3,4])
b=np.array([5,6,7,8])
plt.plot(a,b)
plt.show()
```



b)

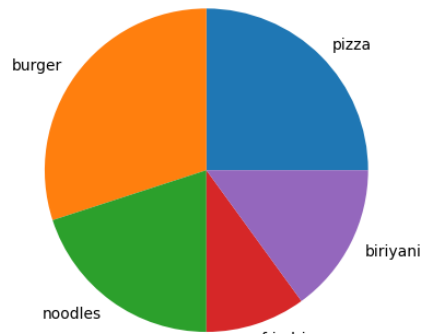
Line graph

```
import numpy as np
import matplotlib.pyplot as plt
a=np.array([1,2,3,4])
b=np.array([5,6,7,8])
plt.plot(a,b,marker='o',linestyle='dotted',color='purple')
plt.show()
```



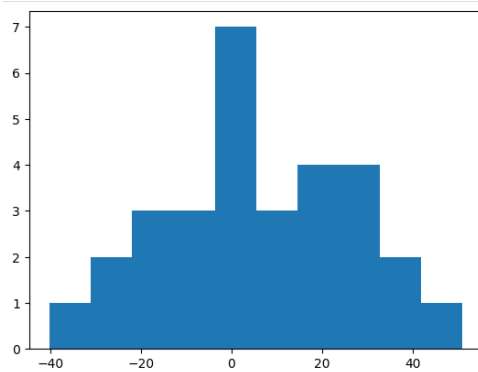
c) Pie chart

```
import numpy as np
import matplotlib.pyplot as plt
a=['pizza','burger','noodles','friedrice','biryani']
b=[25,30,20,10,15]
plt.pie(b,labels=a)
plt.show()
```



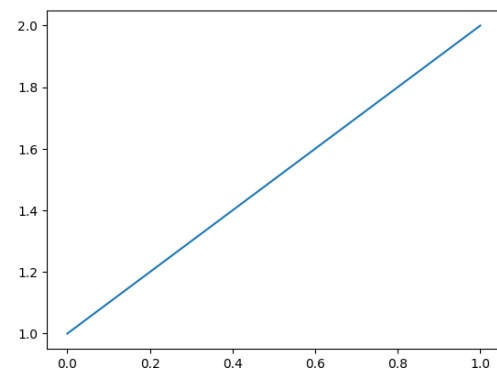
d) Histogram

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data=np.random.normal(10,20,30)
plt.hist(data)
plt.show
```



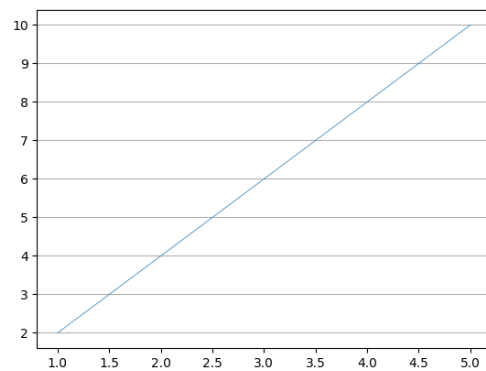
e)

```
import numpy as np
import matplotlib.pyplot as plt
a=np.array([[1],[2]])
plt.plot(a)
plt.show()
```



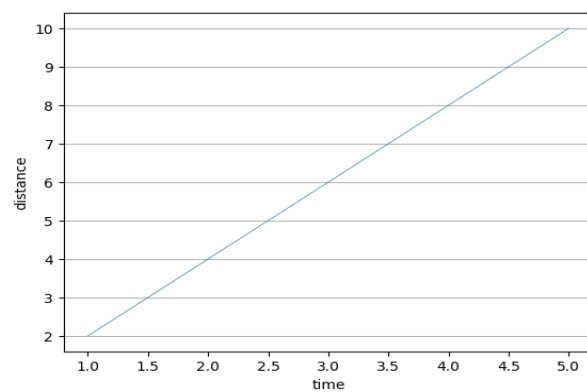
f)

```
import numpy as np
import matplotlib.pyplot as plt
a=np.array([1,2,3,4,5])
b=np.array([2,4,6,8,10])
plt.plot(a,b,linewidth=0.5)
plt.grid(axis='y')
plt.show()
```



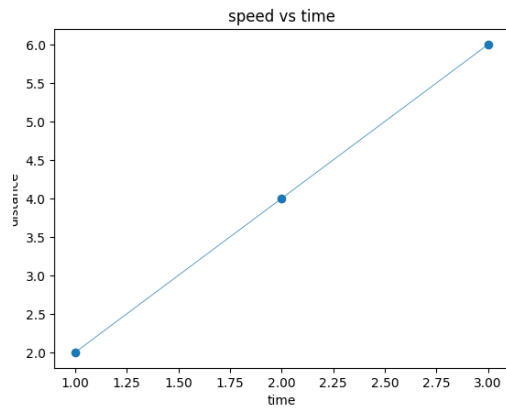
g)

```
import numpy as np
import matplotlib.pyplot as plt
a=np.array([1,2,3,4,5])
b=np.array([2,4,6,8,10])
plt.plot(a,b,linewidth=0.5)
plt.grid(axis='y')
plt.xlabel('time')
plt.ylabel('distance')
plt.show()
```



h)

```
import numpy as np
import matplotlib.pyplot as plt
a=np.array([1,2,3])
b=np.array([2,4,6])
plt.plot(a,b,linewidth=0.5)
plt.xlabel('time')
plt.ylabel('distance')
plt.title('speed vs time')
plt.scatter(a,b)
plt.show()
```



10) Date and time


```
In [1]: 1 import datetime as d
        2 r=d.datetime.now()
        3 print(r)
```

2024-08-21 10:40:11.497834

```
In [4]: 1 import datetime as d
        2 r=d.datetime.now()
        3 tomo=r+d.timedelta(days=1)
        4 print(tomo)
```

2024-08-22 10:42:12.136479

```
In [5]: 1 yest=r-d.timedelta(days=1)
        2 print(yest)
```

2024-08-20 10:42:12.136479

```
In [6]: 1 import datetime as d
        2 r1=d.datetime(2020,6,8,23,10,25,404040)
        3 print(r1)
```

2020-06-08 23:10:25.404040

```
In [8]: 1 print(r1.replace(day=7,month=12,year=2003))
```

2003-12-07 23:10:25.404040

```
In [13]: 1 from datetime import date
        2 print(date(2005,7,11).ctime())
```

Mon Jul 11 00:00:00 2005

```
In [14]: 1 print(r.strftime("%Y"))
```

2024

```
In [15]: 1 print(r.strftime("%y"))
```

24

```
In [16]: 1 print(r.strftime("%M"))
```

M

```
In [17]: 1 print(r.strftime("%p")) #am or pm
```

AM

```
In [19]: 1 print(r.strftime("%Y"))#full year
        2 print(r.strftime("%y"))#last 2 digits of year
```

2024

24

```
In [22]: 1 print(r.strftime("%b"))#to display month(short form)
          2 print(r.strftime("%B"))#to display month(full form)
```

Aug
August

```
In [23]: 1 print(r.strftime("%H"))
```

10

```
In [ ]: 1
```