220801185

EX. NO :06

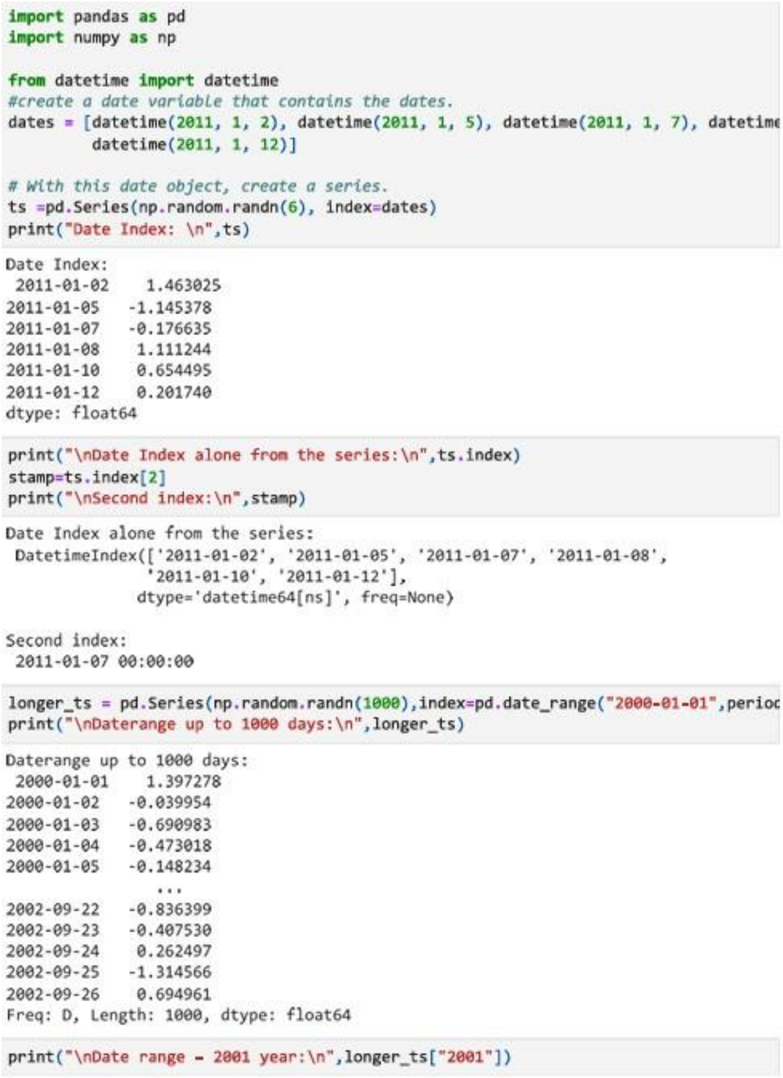
DATE : 06/04/2024

TIME SERIES

AIM :

To implement and check Time Series in Python

PROGRAM :



```
import pandas as pd
import numpy as np

from datetime import datetime
#create a date variable that contains the dates.
dates = [datetime(2011, 1, 2), datetime(2011, 1, 5), datetime(2011, 1, 7), datetime
         datetime(2011, 1, 12)]

# With this date object, create a series.
ts =pd.Series(np.random.randn(6), index=dates)
print("Date Index: \n",ts)
```

```
Date Index:
 2011-01-02    1.463025
2011-01-05   -1.145378
2011-01-07   -0.176635
2011-01-08    1.111244
2011-01-10    0.654495
2011-01-12    0.201740
dtype: float64
```

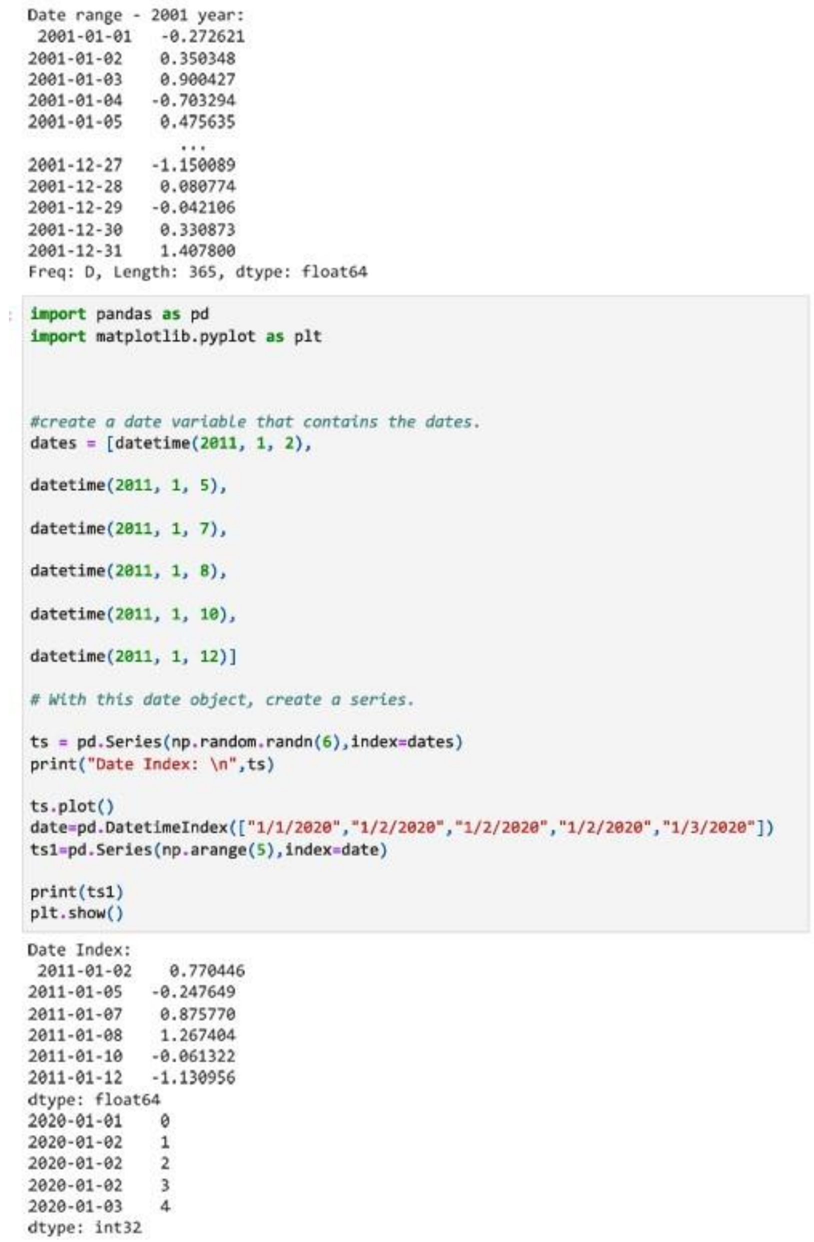
```
print("\nDate Index alone from the series:\n",ts.index)
stamp=ts.index[2]
print("\nSecond index:\n",stamp)
```

```
Date Index alone from the series:
 DatetimeIndex(['2011-01-02', '2011-01-05', '2011-01-07', '2011-01-08',
               '2011-01-10', '2011-01-12'],
              dtype='datetime64[ns]', freq=None)

Second index:
 2011-01-07 00:00:00
```

```
longer_ts = pd.Series(np.random.randn(1000),index=pd.date_range("2000-01-01",perioc
print("\nDaterange up to 1000 days:\n",longer_ts)
```

```
Daterange up to 1000 days:
 2000-01-01    1.397278
2000-01-02   -0.039954
2000-01-03   -0.690983
2000-01-04   -0.473018
2000-01-05   -0.148234
                ...
2002-09-22   -0.836399
2002-09-23   -0.407530
2002-09-24    0.262497
2002-09-25   -1.314566
2002-09-26    0.694961
Freq: D, Length: 1000, dtype: float64
```

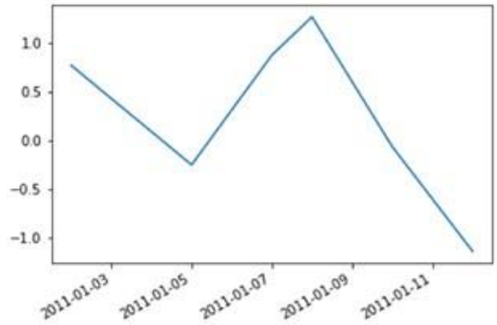
```
print("\nDate range - 2001 year:\n",longer_ts["2001"])
```

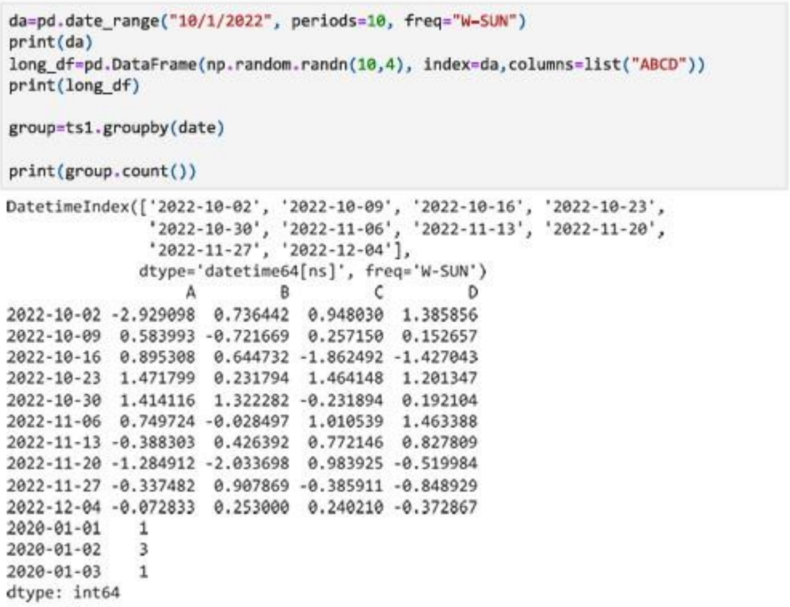


```
Date range - 2001 year:
 2001-01-01   -0.272621
2001-01-02    0.350348
2001-01-03    0.900427
2001-01-04   -0.703294
2001-01-05    0.475635
                 ...
2001-12-27   -1.150089
2001-12-28    0.080774
2001-12-29   -0.042106
2001-12-30    0.330873
2001-12-31    1.407800
Freq: D, Length: 365, dtype: float64
```

```
import pandas as pd
import matplotlib.pyplot as plt



#create a date variable that contains the dates.
dates = [datetime(2011, 1, 2),

datetime(2011, 1, 5),

datetime(2011, 1, 7),

datetime(2011, 1, 8),

datetime(2011, 1, 10),

datetime(2011, 1, 12)]

# With this date object, create a series.

ts = pd.Series(np.random.randn(6),index=dates)
print("Date Index: \n",ts)

ts.plot()
date=pd.DatetimeIndex(["1/1/2020","1/2/2020","1/2/2020","1/2/2020","1/3/2020"])
ts1=pd.Series(np.arange(5),index=date)

print(ts1)
plt.show()
```

```
Date Index:
 2011-01-02    0.770446
2011-01-05   -0.247649
2011-01-07    0.875770
2011-01-08    1.267404
2011-01-10   -0.061322
2011-01-12   -1.130956
dtype: float64
2020-01-01    0
2020-01-02    1
2020-01-02    2
2020-01-02    3
2020-01-03    4
dtype: int32
```





```
da=pd.date_range("10/1/2022", periods=10, freq="W-SUN")
print(da)
long_df=pd.DataFrame(np.random.randn(10,4), index=da,columns=list("ABCD"))
print(long_df)

group=ts1.groupby(date)

print(group.count())
```

```
DatetimeIndex(['2022-10-02', '2022-10-09', '2022-10-16', '2022-10-23',
               '2022-10-30', '2022-11-06', '2022-11-13', '2022-11-20',
               '2022-11-27', '2022-12-04'],
              dtype='datetime64[ns]', freq='W-SUN')
                   A         B         C         D
2022-10-02 -2.929098  0.736442  0.948030  1.385856
2022-10-09  0.583993 -0.721669  0.257150  0.152657
2022-10-16  0.895308  0.644732 -1.862492 -1.427043
2022-10-23  1.471799  0.231794  1.464148  1.201347
2022-10-30  1.414116  1.322282 -0.231894  0.192104
2022-11-06  0.749724 -0.028497  1.010539  1.463388
2022-11-13 -0.388303  0.426392  0.772146  0.827809
2022-11-20 -1.284912 -2.033698  0.983925 -0.519984
2022-11-27 -0.337482  0.907869 -0.385911 -0.848929
2022-12-04 -0.072833  0.253000  0.240210 -0.372867
2020-01-01    1
2020-01-02    3
2020-01-03    1
dtype: int64
```

Result:

Hence the Time Series is implemented.