(ii) Add title "Delhi AQI for last ten days". (1)

(iii) Set label for x-axis "Date" and y-axis "AQI". (1)

(iv) Show grids in the background. (1)

(v) Set marker as '*'. (1)

[This question paper contains 16 printed pages.]

Your Roll No.....

Sr. No. of Question Paper : 4432 **G**

Unique Paper Code : 32347507

Name of the Paper : Data Analysis and Visualization

Name of the Course : **B.Sc. (Hons.) Computer Science**

Semester : V

Duration : 3 Hours Maximum Marks : 75

Instructions for Candidates

1. Write your Roll No. on the top immediately on receipt of this question paper.
2. Question No. **1** is compulsory.
3. Attempt any **four** questions out of **Q.2** to **Q.7**.
4. Parts of a question must be answered together.

1. (a) Provide code to create a time-series with two index labels- 2011/9/01 and 2011/9/02. Assign random values. (2)

(b) What will be the output of the following codes?

(i) (2)

```
import numpy as np
arr = np.array([[1,2,3,4,5],[6,7,8,9,10]])
print(arr[1,-1], arr[-1:])
```

(ii) (2)

```
List = [str[::-1] for str in ('happy','go','lucky')]
print(List)
```

(c) Reshape the following array to dimension (2,6)

[[3,4,5,6], [7,8,9,10], [11,12,13,14]] (2)

(d) Python is a strongly "typed" language. Comment. (2)

(ii) Downsample data to 30s. (2)

7. (a) Create a DataFrame of 8 rows and 8 columns containing random integers in the range of 1 to 10. Compute the correlation of each row with the preceding row. (2)

(b) Consider the following table that lists the last week Delhi's AQI.

AQI	Date
67	2/10/2022
79	3/10/2022
80	4/10/2022
90	5/10/2022
99	6/10/2022
110	7/10/2022
112	8/10/2022
140	9/10/2022
165	10/10/2022
178	11/10/2022

(i) Plot a line graph showing AQI (Air Quality Index) against date with line colour as red, line width as "4pixels" and dashed line style. (4)

(b) Identify the need to resample Timeseries data. (2)

(c) Consider following dataset.

Datetime	value1	value2	value3
2020-01.01 00:00:00	2	92	56
2020-01.01 00:01:00	9	78	80
2020-01.01 00:02:00	69	83	43
2020-01.01 00:03:00	47	62	45
2020-01.01 00:04:00	47	90	13
...	...	...	...
2020-02.27 23:56:00	73	81	35
2020-02.27 23:57:00	20	66	58
2020-02.27 23:58:00	42	16	48
2020-02.27 23:59:00	32	40	19
2020-02.28 00:00:00	37	63	95
83521 rows x 3 columns			

(i) Resample for 10min with sum function for *value1*, mean for *value2* and max for *value 3*. (3)

(e) Give the output of the following code and identify the role of *is_unique* attribute in the code. (2)

```
import pandas as pd
series = pd.Series([4,5,1,2,3,3,4,5,6])
print(series)
print("Is Unique: ",series.is_unique)
```

(f) Differentiate between mutable and immutable objects. (2)

(g) Write a program to create the given dataframe. (3)

	id	value
0	1	a
1	1	a
2	2	b
3	3	None
4	3	a
5	4	a
6	4	None
7	4	b

Further, split it into groups and count unique values of *'value'* column.

(h) Provide the output of the following code : (3)

```
from datetime import datetime, date, time
dt = datetime(2011, 10, 29, 20, 30, 21)
dt2 = datetime(2011, 11, 15, 22, 30)
delta = dt2 - dt

print(delta)
print(type(delta))
print(dt.replace(minute=0, second=0))
```

(i) Consider the given dataframe *df* containing data of students admitted in the college. (3)

Id	Name	Age	Section	City	Gender	Marks
S0	Anit	10	A	Gurgaon	M	60
S1	Alka	22	B	Delhi	F	80
S2	Sid	13	C	Mumbai	M	60
S3	Ruhi	21	B	Delhi	F	55
S4	Nehu	12	B	Mumbai	F	60
S5	Geet	11	A	Delhi	F	56
S6	Om	17	A	Mumbai	M	45

(c) Write the code to split the given dataset into groups based on *customer_id* and create a list of order date *ord_date* for each group. (4)

	ord_no	purch_amt	ord_date	customer_id
0	70009.0	890.00	2012-09-11	3004.0
1	70002.0	270.65	2012-09-10	3001.0
2	70007.0	65.26	2012-09-11	3001.0
3	70008.0	78.00	2012-09-10	3002.0
4	70006.0	948.50	2012-09-17	3002.0
5	70005.0	2400.60	2012-07-27	3001.0
6	70004.0	5760.00	2012-09-10	3001.0
7	70010.0	1983.43	2012-10-10	3004.0
8	70003.0	2480.40	2012-10-10	3003.0
9	70012.0	250.45	2012-06-27	3002.0
10	70034.0	75.29	2012-08-17	3001.0
11	70022.0	56.90	2012-06-27	3003.0

6. (a) Create a Timeseries Dataframe with date range 01-02-2022 to 30-02-2022 with 1 min frequency interval. The dataframe has two columns populated with random values. (3)

5. (a) Give output of the following code. Justify your answer. (2)

```
var=(1, 2, (3,4))
   var[1]='geet'
   print(var)
```

(b) Write the code to merge the two given datasets using key1, key2. (4)

data1:

	key1	key2	P	Q
0	K0	K0	P0	Q0
1	K0	K1	P1	Q1
2	K1	K0	P2	Q2
3	K2	K1	P3	Q3

data2:

	key1	key2	R	S
0	K0	K0	R0	S0
1	K1	K0	R1	S1
2	K1	K0	R2	S2
3	K2	K0	R3	S3

Set the first column *'Id'* as the row index of the given dataframe *df.* Create a pivot table of df to display the total number of admissions as per *'Section'* and *'Gender'.*

(j) (i) Provide the output of the following code : (4)

```
df = pd.DataFrame({
'a':np.arange(1,7),
'b':np.arange(7,13),
'c':np.arange(12,18),
'd':np.arange(17,23),
'e':np.arange(23,29),
'f':np.arange(29,35)},
columns=['a', 'b', 'c', 'd', 'e','f'],
index=['Svaksh', 'Sarah', 'Svaraj', 'Rivika',
'Rahul','Geet'])}
print(df)
df.iloc[2:4,[1,2]]=np.NaN
print(df)

mapping = {'a': 'red', 'b': 'red', 'c': 'blue', 'd':
'blue', 'e': 'red', 'f': 'orange'}
```

(ii) Using the above dataframe, group *df by* mapping and find the sum.

(k) Consider the following dataset to perform the following operations : (4)

	Age	Section	City	Gender	Favourite_ color
0	10	A	Gurgaon	M	red
1	22	B	Delhi	F	NaN
2	13	C	Mumbai	F	yellow
3	21	B	Delhi	M	NaN
4	12	B	Mumbai	M	black
5	11	A	Delhi	M	green
6	17	A	Mumbai	F	red

(i) Find all the raws Where Age is greater than or equal to 12 and the *Gender* is male.

(ii) If *Age* is greater than 20, then use the loc function to update *Section* with "S" and *City* with Pune.

(iii) Select rows 1 to 2 with columns 2 to 3

(ii) Group the data on the column *customer_Id* and create a list of order date *ord_date* for each group. (2)

(iii) Group on the columns *customer_id, salesman_id* and then sort sum of *purch_amt* within the groups. (2)

(b) (i) Write a generator function to print Fibonacci numbers. (4)

(ii) What is the output of the following code :

```
def simpleGeneratorFunc():
    yield 1
    yield 2
x = simpleGeneratorFunc()

print(next(x))
print(next(x))
```

4. (a) Consider following dataframe.

ord_no	purch_amt	ord_date	customer_id	salesman_id
70009.0	890.00	2012-09-11	3004.0	5001
70002.0	270.65	2012-09-10	3001.0	5006
70007.0	65.26	2012-09-11	3001.0	5005
70008.0	78.00	2012-09-10	3002.0	5003
70006.0	948.50	2012-09-17	3002.0	5002
70005.0	2400.60	2012-07-27	3001.0	5001
70004.0	5760.00	2012-09-10	3001.0	5003
70010.0	1983.43	2012-10-10	3004.0	5006
70003.0	2480.40	2012-10-10	3003.0	5005
70012.0	250.45	2012-06-27	3002.0	5002
70034.0	75.29	2012-08-17	3001.0	5004
70022.0	56.90	2012-06-27	3003.0	5005

With respect to the above dataframe, write the code for the following :

(i) Group the data on the column *ord_date* and calculate the total purchase amount *purch_amt* year wise and month wise.

(2)

using iloc.

(L) What is the output of the following code : (4)

```
import pandas as pd
fruits=['apple','orange','apple','apple']*2
N=len(fruits)


print(N)
df=pd.DataFrame({'fruit':fruits,'basket_ID':
np.arange(N),'count' :
np.random.randint(3,15,size=N),'weight':
np.random.uniform(0,4, size=N)})
print(df)
Fruit_cat=df['fruit'].astype('category')
print(Fruit_cat)
  print(df.dtypes)
```

2. (a) Differentiate between :

(i) qcut and cut methods

(ii) Pandas.merge and pandas.concat (2)

(b) Consider the following numeric grades (out of 4). Formulate bins for the given grades as per the following condition : (3)

Below 2.5	Very bad
Between 2.5 to 3	Bad
Between 3 to 3.25	Average
Between 3.25 to 3.5	Good
Between 3.5 to 3.75	Very good
Between 3.75 to 4	Excellent

(c) Given the following dataframe, provide the output for the following commands : (5)

	ord_no	purch_amt	ord_date	customer_id
0	NaN	NaN	NaN	NaN
1	NaN	270.65	2012-09-10	3001.0
2	70002.0	65.26	NaN	3001.0
3	NaN	NaN	NaN	NaN
4	NaN	948.50	2012-09-10	3002.0
5	70005.0	2400.60	2012-07-27	3001.0
6	NaN	5760.00	2012-09-10	3001.0
7	70010.0	1983.43	2012-10-10	3004.0
8	70003.0	2480.40	2012-10-10	3003.0
9	70012.0	250.45	2012-06-27	3002.0
10	NaN	75.29	2012-08-17	3001.0
11	NaN	NaN	NaN	NaN

(i) df.dropna(thresh=2)

(ii) df.dropna(how='all')

(iii) df.dropna(how='all, axis=1)

(iv) df.isnull()

(v) df.isnull().values.any()

3. (a) Write the code to read each row of a given csv file. Skip the header of the file while reading. Also print the number of rows and the field names. (6)

(b) (i) Differentiate between ffill and bfill. (4)

(ii) Provide the output of the given code :

```
import pandas as pd
obj3 = pd.Series(['blue', 'purple', 'yellow'],
                 index=[0, 2, 4])
print(obj3.reindex(range(6), method='ffill'))
print(obj3.reindex(range(6), method='bfill'))
```