CBSE Class 12th - Data handling using Pandas

2 Marks - Most Important Questions

[1] What do you mean by python libraries? Illustrate your answer with an example.

Ans.: A collection of built in modules that allows to peform many tasks with optimized program codes. Each library in python contains a large number of modules that one can imort and uses.

Numpy, Pandas and Martplotlb are three well established python librries for scientific and analytical use. These libraries also used to manipulate, trasnform and visualize data easily and efficiently.

[2] NumPy stands for what? What is used for?

Ans.: NumPy stands for Numerical Python.
It is a python pakacge used for numerical data analysis and scientific computing.

It uses multidimensional array object and have tools and functions working with arrays.

[3] What do you mean by Pandas? What are three important data structures of pandas?

Ans.: Pandas Stand for Panel Data System. It is a high-level data manipulation tool used for data analysis. Pandas library contains a rich set of functions for data analysis work.

Three important data structure of pandas are as follows:

1] Series
2] DataFrame
3] Panel

[4] What do you mean by data structure?

Ans.: A data structure is a special format of data that allows to organize, process, retrieve and store data in systematic manner. It make it easy for users to access and work with data and presents them in human readable form.

[5] What is Matplolib library? Explain in short.

Ans.: The Matplotlib library in Python is used for plotting graphs and visualisation. Using Matplotlib, with just a few lines of code we can generate publication quality plots, histograms, bar charts, scatterplots, etc. It is also built on Numpy, and is designed to work well with Numpy and Pandas.

[6] Differentiate between NumPy and Pandas.

Ans.:

	NumPy	Pandas
Developer	Travis Oliphant	Wes McKinney
Requires	Homogenous Data	Heterogeneous data
Applications	Useful in calculations	Useful in data processing applications
Data format	Used where data is available in Tabular format	Used for numeric-based data manipulations
Dataset	handles small dataset	handles large dataset
Tools	NumPy works on Arrays	Pandas works on Series and DataFrame
Speed	Fast in processing compared to Pandas	Slow in processing compared to NumPy

[7] Samira wants to install python pandas on her laptop. She is not aware of the prerequisite for the same and commands to install it. Help her to understand her requirements and complete her task.

Ans.: She needs to download and install python 3.5.3 and above version.
To install python pandas following command is used:
pip install pandas

To get the version information of pandas use this command in python:
pd.show_versions()
Consider pandas is imported as import pandas as pd

[8] Krish wants to store an individual score of players 34,56,78,23 with appropriate data labels. Which Pandas data structure is used for this purpose? Write a short code to store the values with data labels as names of players.

Ans.: Kirsh can use series data structure.
Code:

```
import pandas as pd
s=pd.Series([34,56,78,23],index=['Virat','Rohit','Surya','Rahul'])
```

[9] Lax is confused to understand what is series and how it is helpful in data analysis? Illustrate the answer in short and help him.

Ans.: A series is a one-dimensional array containing a sequence of values of any data type having numeric data labels starting with 0 by default. The associated labels with each value is known as index. These index can be assigned with other values as well. It represents the values in single column.

[10] Shiv wants to know various ways to create a series. Suggest any four ways to create a series for him.

Ans.: The following ways are as follows for Shiv:
1) From scalar values
2) From NumPy arrays
3) From dictionaries
4) From lists/tuples/sets

[11] Identify the error from below given code, rectify errors and rewrite the correct code:

```
import panda as pd

s=pd.series([99,999,9999],index=1,2,3)

print s
```

Ans.:

```
import pandas as pd

s=pd.Series([99,999,9999],index=[1,2,3])

print(s)
```

[12] Adnan has been given a task to create an empty series. But he is confused about what it is and how to create it. Suggest to him the way to create it and understand what is empty series?

Ans. An empty series is a series object created using python pandas without any value. To create an empty series Adnan has to write the following code:
import pandas as pd
s=pd.Series()

[13] Krushn has given the task to create an empty series with default data type float. Write code for the same and display series.

Ans.:
import pandas as pd
s=pd.Series(dtype=float)
print(s)

[14] What are the various ways of accessing series elements?

Ans. There are three ways to access pandas series elements. They are:
1) loc: It is a label based indexing.
2) Labels List: List of labels given inside loc.
3) Slicing : It extracts subset of series elements as start, stop and step value
4) iloc: Ineteger based indexing by position
5) direct indexing with []: Specifying the index value in square brackets

[15] What do you mean by slicing? How to access series elements using slicing? Explain in short.

Ans.: Slicing refers to extract part of series by sepcifying the start and end parameters. The positional indices are used to slice. The value at the endindex will be exculded.
For example:
s[1:4] will return 3 elements with the positional index 1,2, and 3.

[16] Consider the following code and fill in the gaps to get the given output:

import _____ as pd

s=pd.Series(_____,_____)

print(___)

1	3

2	3
3	3
4	3

Ans.:
```
import pandas as pd
s=pd.Series(3,index=[1,2,3,4])
print(s)
```

[17] Consider the following series and write code to get the given output:

Shikhar	23
Virat	78
Rohit	53
Rahul	40
Hardik	37
Surya	85

Output 1:

Shikhar	23
Rohit	53
Hardik	37

Output 2:

Virat	78

Rahul	53
Surya	85

Ans.:

print(s[::2])

print(s[1::2])

[18] Consider the above data frame and write output for the following code:

print(s[-3:])

print(s[:-3])

Ans.:

Output 1:

Rahul	40
Hardik	37
Surya	85

Output 2:

Surya	23
Virat	78
Rohit	53

[19] Consider the above data frame and write code for the following:

i) Display the records of batsmen who scored more than 50.

ii) Display the runs of Rohit.

Ans.:

i) print(s[s>50])

ii) print(s['Rohit'])

[20] What will be the output of the following:

```
import pandas as pd
s=pd.Series([[34,56,78],[4,3,2]])
print(s)
```

Ans.:

```
0 [34, 56, 78]
1 [4, 3, 2]
dtype: object
```

[21] Consider a given Series , M1. Write a program in Python Pandas to create the series.

	Marks
PT1	23
WT2	19
PT2	58
WT2	21

Ans.:

```
import pandas as pd
M1=pd.Series([23,19,58,21],index=['PT1','WT1','PT2','WT2'])
print(M1)
```

[22] Given two series S1 and S2

S1	
P	23
Q	14

R	19
S	24
S2	
P	17
Q	17
T	17
U	17

Find the output for the following python pandas statements.

a. S1[: 2]*100

b. S1 * S2

Ans.:

a)

Steps:

- S1[:2] returns the values at 0 and 1 index – 23 and 14.
- Now its multiplies by 100.
- So the final values are 2300 and 1400

Output:

P 2300

Q 1400

dtype: int64

b) It performs multiplication with matched values and the rest will be shown NaN.

Output:

P 230.0

Q 140.0

R NaN

S NaN

T NaN

U NaN

dtype: float64

[23] Carefully observe the following code:

```
import pandas as pd
L=[[1,'Adnan',68],[2,'Krish',56],[3,'Krishn',55],[4,'Laksh',45],[5,'Shiv',52]]
df=pd.DataFrame (L, columns=['Rno','Name','Marks'])
print(df)
```

i. Write the statement to print shape of dataframe df

ii. Write statement to print the index and column names of dataframe df

Ans:

i. df.shape

ii. df.index, df.columns

[24] Write a program to create a series objects using a dictionary that stores the number of cities in district of your state.

Note: Assume some districts like Ahmedabad, Baroda, and Bharuch have 44,32,12 cities respectively and the pandas library has been imported as pd_district_cities.

Ans.:

```
import pandas as pd_district_cities
dis={'Ahmedabad':44,'Baroda':32,'Bharuch':12}
s=pd_district_cities.Series(dis)
```

[25] Write a program to create a series object using a dictionary that stores the number of students in each section of class XII.

Note: Assume four sections as ‘XII A’, ‘XII B’, ‘ XII C’ and ‘XII D’ with 40, 38, 40, 26 students respectively.

Ans.:

```
import pandas as pd
d={'XII A':40,'XII B':38,'XII C':40,'XII D':26}
s=pd.Series(d)
```

[26] Consider the commands below:

```
import pandas as pd
lst=[10,20]
ds=pd.Series([10,20])
```

Here lst is a list and ds is a series. Both have same values 10 and 20. What will be the output of the following commands? Justify your answer.

a. print (lst * 2)

b. print (ds * 2)

Ans.:

a) [10,20,10,20]

b) 0 20

1 40

[27] What will be the output of the following code:

```
import pandas as pd
mydata=pd.Series( ['Delhi', 'Mumbai', 'Kolkata', 'Ahmedabad'] )
print(mydata < 'Baroda' )
```

Ans.:

0 False

1 False

2 False

3 True

dtype: bool

[28] Carefully observe the following code:

```
import pandas as pd
xiic = {'Arnavee':[34,59], 'Samira':[27,38], 'Ayush':[37,55]}
df = pd.DataFrame(xiic,index=['PT1','PT2'])
print(df)
```

What will be the output of following:

i) print(df.dtypes)

ii) print(df.axes)

Ans.:

i)

```
Arnavee    int64
Samira     int64
Ayush      int64
dtype: object
```

ii) [Index(['PT1', 'PT2'], dtype='object'),

Index(['Arnavee', 'Samira', 'Ayush'],

dtype='object')]

[29]Consider the following DataFrame, DF

	Rollno	Name	Class	Section	CGPA
St1	1	Aman	IX	E	8.7
St2	2	Preeti	X	F	8.9
St3	3	Kartikey	IX	D	9.2
St4	4	Lakshay	X	A	9.4

Write commands to :

i. Write a statement to print the dimensions of given dataframe

ii. Write a statement to print the number of elements of given dataframe

Ans.:

i) DF.ndim

ii) DF.size or DF.shape[0]*DF.shape[1]

[30] Carefully observe the following code:

```
import pandas as pd
L=[['XIIC01','Nishtha',65,67],['XIIC02','Rohit',56,54],['XIID04','Laksh',37,43]
,['XIID05','Krishn',45,48]]
df=pd.DataFrame (L, columns=['ID','Name','PB1','PB2'])
print(df)
```

i. Write a statement to display the average of Preboard 1 and Preboard 2 marks

ii. Write a statement to display the difference between Preboard 1 and Preboard 2 marks

Ans.:

i. (df.PB1+df.PB2)/2 or (df['PB1']+df['PB2'])/2

ii. df.PB1-df.PB2 or df['PB1']-df['PB2']