

Pandas

Multiple Choice Questions

1 Which two data structures are commonly used in Pandas?

- A. Array and Tuple B. List and Dictionary
- C. Series and DataFrame D. Stack and Queue

Answer: C. Series and DataFrame

2 What is a Series in Pandas?

- A. A two-dimensional labeled array
- B. A one-dimensional array with labeled data
- C. A type of function in Pandas
- D. A spreadsheet file format

Answer: B. A one-dimensional array with labeled data

3 In a Pandas Series, what is the default type of index assigned to values?

- A. Alphabetic labels
- B. Custom-defined names
- C. Numeric labels starting from one
- D. Numeric labels starting from zero

Answer: D. Numeric labels starting from zero

4. Which of the following best describes the index in a Pandas Series?

- A. A row number in a table
- B. A name given to the Series
- C. A data label associated with each value in the Series
- D. The length of the Series

Answer: C. A data label associated with each value in the Series

5. In Pandas library of Python, a one-dimensional array containing a sequence of values of any datatype is known as :

- A. DataFrame B. Histogram C. Series D. Panel

Answer: C. Series

6 Which of the following command is used to display the first three elements of a series, 'S' ?

- A. S.head() B. S.head(4) C. S.Head() D. S.head(3)

Answer: D. S.head(3)

7 Which of the following command is used to display the last three elements of a series, 'S' ?

- A. S.Tail B. S.tail() C. S.tail(3) D. S.Tail(3)

Answer: C. S.tail(3)

8 Which attribute can be used to name the series?

- A. name B. index C. Name D. seriesname

Answer: A. name

9 Which attribute returns a list of values in the series object?

- A. value B. values C. elements D. ditems

Answer: B . values

10 Which attribute checks the existence of NaN value in a Series?

- A. len() B. count() C. values() D. hasnans()

Answer: D.hasnans()

11 Which of the following can be used to create a Pandas Series?

A. List B. Dictionary C. Scalar value D. All of the above

Answer: D.All of the above

11 Which of the following can be used to create a Pandas Series?

A. List B. Dictionary C. Scalar value D. All of the above

Answer: D.All of the above

12 What will be the output of the following code?

```
import pandas as pd
s = pd.Series([10, 20, 30], index=['a', 'b', 'c'])
print(s['b'])
```

A.10 B. 20 C. b D. Error

Answer: B. 20

13 What does the dtype property of a Series return?

A. Number of elements
B. Data type of the Series elements
C. Memory size
D. None of the above

Answer: B. Data type of the Series elements

14 Which of the following statement will import pandas library?

A. import panndas as py
B. import pandas as pd
C. import panda as pd
D. Import pandas as pd

Answer: B. import pandas as pd

Short Answer Type Questions

1 Find the output of the following Python code

```
import pandas as pd
S1=pd.Series([ 1,2,2,7,5], index=['a', 'b', 'c' , 'd' , 'e'])
print(S1)
```

pg. 18

```
print(S1.head(2))
```

Answer:

a 1

b 2

c 2

d 7

e 5

dtype: int64

1

b 2

dtype: int64

2 “Series has immutable index” . Explain?

Answer:

Once a series is created, the size or structure of index cannot be changed in place whereas the entire index can be replaced. One can not add more elements to a series or remove elements from the original series.

3 Give the output of the following code:

```
import pandas as pd
```

```
# Dictionary with names as keys and marks as values
marks_dict = { 'Alice': 85, 'Bob': 78, 'Charlie': 92, 'David': 76, 'Eva': 89 }
# Creating a Series from the dictionary
marks_series = pd.Series(marks_dict)
# Display the Series
print("Student Marks:\n")
print(marks_series)
```

Answer:

Student Marks:

Alice 85

Bob 78

Charlie 92

David 76

Eva 89

dtype: int64

4 Consider the following series, 'Student' , and answer the questions that follow

Alice 85

Bob 78

Charlie 92

David 76

Eva 89

dtype: int64

a) Display the values of this series

b) Display the indexes

c) Find how many elements are there in this series

d) Display the mark of 'Alice'

Answers:

a) **print(Student.values)**

b) **print(Student.index)**

c) **print(Student.size)**

d) **print(Student['Alice'])**

5 Consider the following series, 'Student' , and answer the questions that follow

Alice 85

Bob 78

Charlie 92

David 76

Eva 89

dtype: int64

a) Increase the marks of all the students by 5

b) Display the details if mark is greater than 80

c) Display the marks of Alice, Eva

d) Modify the marks of 'Alice'. New mark is 95

Answers:

a) **Student = Student+5**

b) **print(Student[Student>80])**

c) **print(Student[['Alice', 'Eva']])**

d) **Student['Alice'] = 95**

6 Write the output of the following code:

```
import pandas as pd
S1=pd.Series([10,15,16,12,18,19], index=['A', 'B', 'C', 'D', 'E', 'F'])
print(S1[0:3])
print("_____")
print(S1['A':'B'])
print("_____")
print(S1[0:5:2])
print("_____")
print(S1['A':'F':2])
```

Answers:

A 10

B 15

C 16

dtype: int64

A 10

B 15

dtype: int64

A 10

C 16

E 18

dtype: int64

A 10

C 16

E 18

dtype: int64

7 What do the keys and values in the dictionary represent in a Series?

Answer:

Keys become the **index labels** of the Series.

Values become the **data** (the elements) of the Series.

8 What is a Pandas Series?

Answer:

A Pandas Series is a one-dimensional labelled array capable of holding data of any type (integers, strings, floats, Python objects). It has both data and an index to label each element. Once created we can't add or remove elements in original Series.

9 Can the data type in a Series be mixed?

Answer:

A Series holds elements of a single data type. If mixed types are provided, Pandas will pick a common data type (usually object/string) for all elements.

Example :

```
import pandas as pd
s = pd.Series([10, "hello", 3.14, True])
print(s)
```

print("Data type:", s.dtype) **How do we check for missing values in the Series?**

Answer:

Use the isnull() method, which returns a Boolean Series indicating where values

are missing. Also 'hasnans' attribute can be used which will return True or False depending on the use of NaN values in the series.

11 How can you sort a Series by its index or values?

Answer:

Index: <seriesobject>.sort_index()

Values: <seriesobject>.sort_values()

12 What are the key features of a Pandas Series? How is it different from a NumPy array and a Python list?

Answer:

Key Features of a Pandas Series:

- One-dimensional labeled array.
- Supports different data types.
- Allows custom indexing.
- Supports vectorized operations.
- Handles missing data gracefully using NaN.

Comparison of List, ndarray and Series

13 Give the output of the following:

```
import pandas as pd
```

```
S1=pd.Series([10,15,16,12,18,19], index=['A', 'B', 'C', 'D', 'E', 'F'])
```

```
S2=pd.Series([2,3,2,4,5,3], index=['A', 'B', 'C', 'D', 'F', 'G'])
```

```
print(S1+S2)
```

Answer:

A 12.0

B 18.0

C 18.0

D 16.0

E NaN

F 24.0

G NaN

dtype: float64

Assertion and Reasoning based Questions

A: Assertion, R: Reason

Choose the correct option:

- A. Both A and R are true, and R is the correct explanation of A.
- B. Both A and R are true, but R is NOT the correct explanation of A.
- C. A is true, but R is false.
- D. A is false, but R is true.

1 **Assertion (A):** A Pandas Series is fundamentally a two-dimensional labelled data structure.

Reason (R): It contains both a row and index

Answer: D. A is false, R is true

2 **Assertion (A):** When creating a Pandas Series from a NumPy array, the length of the index labels passed must match the size of the array to avoid a ValueError.

Reason (R): If the lengths do not match, Pandas automatically fills the missing values with NaN to prevent errors

Answer: C. A is true, R is false.

3 **Assertion (A):** The empty attribute of a Pandas Series returns True if the Series has no data values.

Reason (R): This attribute is useful for quickly determining if a Series object has been initialized with any elements

Answer: A. A is true, R is a correct explanation for A

Explanation: The **empty** attribute prints True if the Series is empty and False otherwise. This attribute indeed serves to check whether a Series contains any values, making the reason a valid explanation for the assertion's purpose.

4 Assertion (A): In Pandas Series, positional indexing for slicing excludes the value at the end index position.

Reason (R): This slicing behaviour is identical to how slicing with labeled indexes works, where the end label is also excluded.

Answer: C. A is true, R is false.

5 Assertion (A): Mathematical operations between two Pandas Series automatically align data based on their labels.

Reason (R): If a label is not found in one of the Series during an operation, the result for that label will be marked as NaN

Answer: A. A is true, R is a correct explanation for A

6 Assertion (A): The head(n) method of a Pandas Series returns the last n members of the Series.

Reason (R): By default, if n is not specified, head() returns the first 5 members

Answer: D. A is false, R is true

DATAFARME

Multiple Choice Questions With Answers

1 What is the primary characteristic of a Pandas DataFrame?

- A) A one-dimensional array containing a sequence of values
- B) A collection of built-in Python modules for various actions
- C) A two-dimensional labeled data structure, similar to a spreadsheet, with both row and column indices
- D) A package used for numerical data analysis that works with homogeneous multidimensional arrays

Answer: C

2 How do Pandas DataFrames handle data types compared to NumPy arrays?

- A) Both NumPy arrays and Pandas DataFrames require homogeneous data types
- B) A NumPy array can have different data types, while a Pandas DataFrame requires homogeneous data
- C) A Pandas DataFrame can contain different data types (e.g., float, int, string, datetime) in its columns, whereas a NumPy array requires homogeneous data
- D) Pandas DataFrames only store numerical data, while NumPy arrays can store any data type.

Answer: C

What will be the output of **pd.DataFrame([1, 2, 3])**?

- A) A DataFrame with one column and three rows

- B) A DataFrame with three columns and one row
- C) An error
- D) A Series

Answer: A) A DataFrame with one column and three rows

3 Which parameter is used to set custom row labels when creating a DataFrame?

- A) columns
- B) index
- C) labels
- D) rows

Answer: B) index

4 You have the following dictionary: data = {'A': [1, 2], 'B': [3, 4]}. How many rows will the resulting DataFrame have?

- A) 1
- B) 2
- C) 3
- D) 4

Answer: B) 2

5. What is the default value of the index parameter when creating a DataFrame?

- A) None
- B) Range Index starting from 0
- C) Empty list
- D) Depends on the data

Answer: B) Range Index starting from 0

6 What happens if the lists in a dictionary used to create a DataFrame are of different lengths?

- A) It fills missing values with zeros
- B) It fills missing values with NaN
- C) It raises a ValueError
- D) It aligns the shorter list to the left

Answer: C) It raises a ValueError

7 Which of the following can NOT be used to create a DataFrame?

- A) Dictionary of lists
- B) List of tuples
- C) Single scalar value
- D) NumPy array

Answer: C) Single scalar value

8 To set custom column names in a DataFrame created from a list of lists, which parameter is used?

- A) headers
- B) columns
- C) colnames
- D) labels

Answer: B) columns

9. Give the output of the following code:

```
import pandas as pd
```

```
df = pd.DataFrame({'A': [1, 2], 'B': [3, 4]})  
print(df.shape)
```

- A) (2, 2)
- B) (2, 1)
- C) (1, 2)
- D) (4,)

Answer: A) (2, 2)

10. What will be the output?

```
import pandas as pd  
data = {'Name': ['John', 'Alice'], 'Age': [25, 30]}  
df = pd.DataFrame(data)  
print(df['Name'][1])
```

- A) John
- B) Alice
- C) 25
- D) Error

Answer: B) Alice

11. What is the type of df in the following code?

```
import pandas as pd  
df = pd.DataFrame([{'a': 1, 'b': 2}, {'a': 3, 'b': 4}])
```

- A) Series
- B) DataFrame
- C) Dictionary
- D) List

Answer: B) DataFrame

12 Choose the correct output:

```
import pandas as pd  
df = pd.DataFrame({'x': [1, 2], 'y': [3, 4]}, index=['a', 'b'])  
print(df.index.tolist())
```

- A) ['x', 'y']
- B) ['a', 'b']
- C) [0, 1]
- D) ['1', '2']

Answer: B) ['a', 'b']

13 What does this code return?

```
import pandas as pd  
df = pd.DataFrame({'A': [10], 'B': [20]})  
print(df.values)
```

- A) [[10, 20]]
- B) [10, 20]
- C) (10, 20)
- D) [[10][20]]

Answer: A) [[10, 20]]

Assertion and Reasoning based Question

Directions

In the following questions, a statement of Assertion (A) is followed by a statement of Reason (R). Mark the correct choice as:

(A) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).

(B) Both Assertion (A) and Reason (R) are true, but Reason (R) is not the correct explanation of Assertion (A).

(C) Assertion (A) is true, but Reason (R) is false.

(D) Assertion (A) is false, but Reason (R) is true.

1 Assertion (A): A DataFrame is a two-dimensional labeled array. Its columns can have heterogeneous types, i.e., they can hold varying types of data.

Reason (R): We need a DataFrame with a Boolean index to use Boolean indexing.

Answer: (C) Assertion (A) is true, but Reason (R) is false.

Explanation:

Assertion (A): True. A DataFrame is indeed a two-dimensional, labeled data structure in pandas. Its columns can have different data types, such as integers, strings, floats, etc. This makes DataFrames highly versatile for handling structured data.

Reason (R): False. Boolean indexing does not require the DataFrame to have a Boolean index. Instead, it is a technique for filtering data using Boolean conditions (i.e., True/False values) applied to the DataFrame's rows or columns.

2 Assertion (A): Iteration is a general term for taking each item of something one after another.

Reason (R): `iter tuples()` returns the iterator yielding each index value along with a series containing the data in each row.

Answer: (C) Assertion (A) is true, but Reason (R) is false

3 Assertion (A): Indexing can also be known as subselection.

Reason (R): Pandas DataFrame `.loc` attribute accesses groups of rows and columns by label(s) or a Boolean array in the given DataFrame.

Answer: (A) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).

4 Assertion (A): To delete a column from a pandas DataFrame, `drop()` method is used. **Reason (R):** Columns are deleted by dropping columns with index labels.

Answer: (C) Assertion (A) is true, but Reason (R) is false.

Explanation:

Assertion (A): True. The `drop()` method is indeed used to delete a column from a pandas DataFrame. It allows the removal of one or more columns by specifying their names.

Reason (R): False. Columns are not deleted by using **index labels**; instead, they are deleted by specifying **column labels** (i.e., the names of the columns). The `drop()` method with `axis=1` is used to indicate that a column is being dropped.

5. Assertion (A): Rows can also be selected by passing integer location.

Reason (R): Integer location can be passed to the `.iloc[]` method.

Answer: (A) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).

Explanation

Assertion (A): True. Rows can indeed be selected by passing integer locations, which refers to their positional index in the DataFrame.

Reason (R): True. The `.iloc[]` method is specifically designed for positional indexing, allowing you to select rows and columns using integer-based location indices.

6 Assertion (A): The `head()` function returns the first n rows from the object based on position.

Reason (R): n is the selected number of rows whose default value is 3.

Answer: (C) Assertion (A) is true, but Reason (R) is false.

Explanation:

Assertion (A): True. The `head()` function in pandas does indeed return the first n rows of a DataFrame or Series based on their position. **Reason (R):** False. The default value of n for the `head()` function is 5, not 3.

Assertion (A): A list of dictionaries can be passed to form a DataFrame.

Reason (R): Keys of the dictionary are taken as row names by default.

Answer: (C) Assertion (A) is true, but Reason (R) is false.

Explanation:

7. Assertion (A): True. A list of dictionaries can indeed be passed to pandas to form a DataFrame. Each dictionary in the list represents a row in the DataFrame, and the keys of the dictionary become column names.

Reason (R): False. Keys of the dictionary are not taken as row names by default. Instead, they are used as column names, while the row indices are assigned automatically (starting from 0) unless explicitly specified.

8. Assertion (A): Indexing can also be known as subselection.

Reason (R): Pandas DataFrame `.loc` attribute accesses groups of rows and columns by label(s) or a Boolean array in the given DataFrame.

Answer: (A) Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A).

MATPLOTLIB

Multiple Choice Questions

Q1. Which Python library is commonly used for data visualization?

a) NumPy b) Pandas c) matplotlib d) SciPy

Answer: c) matplotlib

Q2. What does the plot() function in matplotlib do?

a) Reads a file b) Draws a line chart c) Displays data in a table d) Opens a window

Answer: b) Draws a line chart

Q3. Which function is used to display the plot on screen?

a) show() b) display() c) render() d) open()

Answer: a) show()

Q4. Which parameter is used to label a line in plot() for legends?

a) text b) label c) name d) legend Answer:

b) label

Q5. What is the purpose of legend() function in matplotlib?

a) To save the plot b) To show grid lines c) To display the labels of plotted data d) To format the plot

Answer: c) To display the labels of plotted data

Assertion and Reasoning Based Questions Mark the correct choice as:

A. Both Assertion (A) and Reason (R) are true, and Reason (R) is the correct explanation of Assertion (A)

B. Both Assertion (A) and Reason (R) are true, but Reason (R) is not the correct explanation of Assertion (A)

C. Assertion (A) is true, but Reason (R) is false

D. Assertion (A) is false, but Reason (R) is true

Q. 1 Assertion (A): Bar charts are suitable for comparing different categories of discrete data.
Reason (R): Bar charts display data using bins, and the bars touch each other without gaps.

Answer : C

Q. 2 Assertion (A): Line charts are best used to display data that changes over time.

Reason (R): Line charts help in observing trends and patterns in time-series data.

Answer : A

Q.3 Assertion (A): In a histogram, bars must have gaps between them.

Reason (R): Histogram is used for plotting frequency of discrete data.

Answer : D

Exercise Q. 1 Write a Python code using matplotlib to:

1. Plot a bar chart for sales data ['Q1', 'Q2', 'Q3', 'Q4'] with corresponding values [150, 200, 180, 220].
2. Set bar colour to cyan
3. Label axes as "Quarter" and "Sales (in ₹)".
4. Include a title "Quarterly Sales" and display values on top of each bar

. Answer :

```
import matplotlib.pyplot as plt
quarters = ['Q1', 'Q2', 'Q3', 'Q4']
sales = [150, 200, 180, 220]

plt.bar(quarters, sales, color='cyan')      # Custom bar color
plt.xlabel('Quarter')
plt.ylabel('Sales (in ₹)')
plt.title('Quarterly Sales')
```

Q. 2 Radhika has to fill in the blanks in the given Python program that generates a line plot as shown below. The given line plot represents the sales in lakhs over five months as given in the table



```
import _____ as plt  #statement 1
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May']
sales = [50, 60, 45, 90, 67]
```

```
plt._____(months, sales, marker='o', color='blue', linestyle='-', label='Monthly Sales')  
#statement 2
```

```
plt.xlabel('_____')
```

```
plt.ylabel('Sales') plt.title('_____')
```

```
plt.grid(True)
```

```
plt.legend()
```

```
plt.show()
```

4 Write the missing statement according to given specification

1. write the suitable code to import the required module in the blank space in the line mark as statement 1

2. fill in the blank in statement 2 with a suitable python function name to create a line plot.

3. referred to the graph shown and fill in the blank in statement 3 to display the appropriate label for x axis.

4. refer to the graph shown and fill in the blank in statement 4 to display the suitable chart title.
Answer :

```
import matplotlib.pyplot as plt #statement 1
```

```
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May'] sales = [50, 60, 45, 90, 67]
```

```
plt.plot(months, sales, marker='o', color='blue', linestyle='-', label='Monthly Sales') #statement 2
```

```
plt.xlabel('Month') #statement 3
```

```
plt.ylabel('Sales')
```

```
plt.title('Sales Over Months')
```

```
plt.grid(True)
```

```
plt.legend()
```

```
plt.show()
```

note: Along with these questions prepare practice paper of Series ,Dataframe and Matplotlib.