Pandas Questions

1. How do you read a CSV file and display the first 10 rows?
2. How do you find the number of rows and columns in a DataFrame?
3. Given a DataFramedf, how do you display only the rows where age > 25?
4. Replace all missing values in column salary with the **mean salary**.
5. How do you rename the column emp_id to employee_id?
6. How to drop duplicate rows in a DataFrame?
7. How do you convert a column of string dates "2023-05-01" into datetime format?
8. How to group a DataFrame by department and find the total salary for each?
9. How do you sort a DataFrame by date in descending order?
10. Create a new column total_marks by adding three columns: math, science, and english.
11. How do you filter rows where column score is between 40 and 80?
12. Display the top 3 rows with the highest values in column sales.
13. Find the average, min, and max of the price column using a single function.
14. How to set a column (e.g., date) as the index of a DataFrame?
15. Convert a categorical column gender (with values M/F) into numeric using mapping (M → 1, F → 0).

NumPy Questions

16. Create a 1D NumPy array from 0 to 9.
17. Create a 3x3 NumPy array filled with random integers between 1 and 100.
18. How do you find the mean and standard deviation of a NumPy array?
19. Given an array, replace all values greater than 50 with 50.
20. Create a NumPy array and reshape it from 1D to 2D (e.g., 12 elements → 3x4).
21. Find the number of even numbers in a NumPy array.
22. How do you flatten a 2D NumPy array to 1D?
23. Create an array of 10 random floats between 0 and 1.
24. Multiply two NumPy arrays element-wise.
25. Given a NumPy array, how do you find the index of the maximum value?

```
import pandas as pd

import numpy as np


data = {

    'emp_id': [101, 102, 103, 104, 105, 106, 107, 108, 109, 110],

    'age': [22, 35, 28, 24, 21, 29, 26, 27, 34, 30],

    'salary': [50000, None, 60000, 55000, 52000, 58000, None, 61000, 57000, 54000],

    'department': ['HR', 'IT', 'HR', 'Finance', 'HR', 'IT', 'Finance', 'IT', 'HR', 'Finance'],

    'date': ['2022-05-01', '2023-04-15', '2023-03-10', '2024-02-01', '2023-01-20', '2023-01-10', '2023-01-05',
'2022-12-30', '2022-12-15', '2022-12-01'],

    'score': [45, 78, 82, 39, 66, 59, 41, 90, 77, 52],

    'sales': [300, 450, 500, 400, 350, 390, 420, 460, 480, 370],

    'math': [80, 70, 90, 85, 75, 88, 92, 67, 78, 86],

    'science': [75, 65, 95, 88, 70, 85, 90, 60, 77, 89],

    'english': [70, 80, 85, 90, 88, 76, 84, 79, 83, 81],

    'gender': ['M', 'F', 'F', 'M', 'M', 'F', 'F', 'M', 'F', 'M'],

}

df_sample = pd.DataFrame(data)

df_sample.to_csv('sample.csv', index=False)


# 1. Read a CSV file and display the first 10 rows

df = pd.read_csv('sample.csv')

print("1. First 10 rows:\n", df.head(10), "\n")


# 2. Find number of rows and columns

print("2. Shape (rows, columns):", df.shape, "\n")


# 3. Rows where age > 25

print("3. Rows where age > 25:\n", df[df['age'] > 25], "\n")
```

```
# 4. Replace missing salary values with mean

df_copy = df.copy()

df_copy['salary'] = df_copy['salary'].fillna(df_copy['salary'].mean())

print("4. Salary after filling missing values (using df.copy safely):\n", df_copy['salary'], "\n")


# 5. Rename 'emp_id' to 'employee_id'

df.rename(columns={'emp_id': 'employee_id'}, inplace=True)

print("5. Renamed column:\n", df.columns, "\n")


# 6. Drop duplicate rows

df.drop_duplicates(inplace=True)

print("6. After dropping duplicates:\n", df, "\n")


# 7. Convert 'date' column to datetime format

df['date'] = pd.to_datetime(df['date'])

print("7. Date column dtype:", df['date'].dtypes)

print(df[['employee_id', 'date']],"\n")


# 8. Group by department and find total salary

print("8. Total salary by department:\n", df.groupby('department')['salary'].sum(), "\n")


# 9. Sort by date in descending order

df_sorted = df.sort_values(by='date', ascending=False)

print("9. Sorted by date (descending):\n", df_sorted[['employee_id', 'date']])


# 10. Create new column 'total_marks' (math + science + english)

df['total_marks'] = df['math'] + df['science'] + df['english']

print("10. Added 'total_marks':\n", df[['math', 'science', 'english', 'total_marks']], "\n")
```

```
# 11. Filter rows where score is between 40 and 80
print("11. Score between 40 and 80:\n", df[df['score'].between(40, 80)], "\n")

# 12. Top 3 rows with highest sales
print("12. Top 3 by sales:\n", df.nlargest(3, 'sales'), "\n")

# 13. Avg, min, max of 'salary'
print("13. Salary stats:\n", df['salary'].agg(['mean', 'min', 'max']), "\n")

# 14. Set 'date' as index
df.set_index('date', inplace=True)
print("14. After setting date as index:\n", df, "\n")

# 15. Convert gender M/F to 1/0 using mapping
df['gender'] = df['gender'].map({'M': 1, 'F': 0})
print("15. Gender mapped to numeric:\n", df['gender'], "\n")

# 16. Create a 1D NumPy array from 0 to 9
arr1 = np.arange(10)
print("16. 1D array from 0 to 9:\n", arr1, "\n")

# 17. Create a 3x3 NumPy array filled with random integers between 1 and 100
arr2 = np.random.randint(1, 101, (3, 3))
print("17. 3x3 array with random integers (1–100):\n", arr2, "\n")

# 18. Find the mean and standard deviation of a NumPy array
arr_new = np.array([10, 20, 30, 40, 50])
mean_val = np.mean(arr_new)
std_val = np.std(arr_new)
```

```
print("18. Mean and standard deviation of new array [10, 20, 30, 40, 50]:")
print("Mean =", mean_val, ", Std Dev =", std_val, "\n")

# 19. Replace all values > 50 with 50
arr_new = np.array([12, 67, 34, 89, 45, 22, 51])
arr_new[arr_new > 50] = 50
print("19. New array after replacing values > 50 with 50:\n", arr_new, "\n")

# 20. Reshape 1D array (12 elements) to 3x4
arr3 = np.arange(1,13).reshape(3, 4)
print("20. Reshaped 1D array to 3x4 matrix:\n", arr3, "\n")

# 21. Find number of even numbers in a NumPy array
arr4 = np.array([1, 2, 3, 4, 5, 6, 8])
even_count = np.sum(arr4 % 2 == 0)
print("21. Number of even elements in array:\n", even_count, "\n")

# 22. Flatten a 2D NumPy array to 1D
arr_new = np.array([[5, 10, 15], [20, 25, 30]])
arr_flat = arr_new.flatten()
print("22. Flattened new 2D array to 1D:")
print("Original 2D array:\n", arr_new)
print("Flattened 1D array:\n", arr_flat, "\n")

# 23. Create an array of 10 random floats between 0 and 1
arr5 = np.random.rand(10)
print("23. Array of 10 random floats (0 to 1):\n", arr5, "\n")
```

```
# 24. Multiply two NumPy arrays element-wise
a = np.array([[1, 2], [3, 4]])
b = np.array([[5, 6], [7, 8]])
print("24. Element-wise multiplication of two arrays:\n", a * b, "\n")

# 25. Find the index of the maximum value in an array
arr6 = np.array([10, 20, 35, 50, 45])
max_index = np.argmax(arr6)
print("25. Index of maximum value in array:\n", max_index)
```

OUTPUT :

```
1. First 10 rows:
   emp_id  age   salary department        date  score  sales  math  science  english gender
0     101   22  50000.0         HR  2022-05-01     45    300    80       75       70      M
1     102   35      NaN         IT  2023-04-15     78    450    70       65       80      F
2     103   28  60000.0         HR  2023-03-10     82    500    90       95       85      F
3     104   24  55000.0    Finance  2024-02-01     39    400    85       88       90      M
4     105   21  52000.0         HR  2023-01-20     66    350    75       70       88      M
5     106   29  58000.0         IT  2023-01-10     59    390    88       85       76      F
6     107   26      NaN    Finance  2023-01-05     41    420    92       90       84      F
7     108   27  61000.0         IT  2022-12-30     90    460    67       60       79      M
8     109   34  57000.0         HR  2022-12-15     77    480    78       77       83      F
9     110   30  54000.0    Finance  2022-12-01     52    370    86       89       81      M

2. Shape (rows, columns): (10, 11)

3. Rows where age > 25:
   emp_id  age   salary department        date  score  sales  math  science  english gender
1     102   35      NaN         IT  2023-04-15     78    450    70       65       80      F
```

```
2    103   28  60000.0        HR  2023-03-10    82    500    90      95      85     F
5    106   29  58000.0        IT  2023-01-10    59    390    88      85      76     F
6    107   26      NaN   Finance  2023-01-05    41    420    92      90      84     F
7    108   27  61000.0        IT  2022-12-30    90    460    67      60      79     M
8    109   34  57000.0        HR  2022-12-15    77    480    78      77      83     F
9    110   30  54000.0   Finance  2022-12-01    52    370    86      89      81     M
```

```
4. Salary after filling missing values (using df.copy safely):
 0    50000.0
1    55875.0
2    60000.0
3    55000.0
4    52000.0
5    58000.0
6    55875.0
7    61000.0
8    57000.0
9    54000.0
Name: salary, dtype: float64
```

```
5. Renamed column:
 Index(['employee_id', 'age', 'salary', 'department', 'date', 'score', 'sales', 'math', 'science', 'english', 'gender'],
dtype='object')
```

```
6. After dropping duplicates:
   employee_id  age   salary department        date  score  sales  math  science  english gender
0          101   22  50000.0        HR  2022-05-01    45    300    80      75      70     M
1          102   35      NaN        IT  2023-04-15    78    450    70      65      80     F
2          103   28  60000.0        HR  2023-03-10    82    500    90      95      85     F
```

```
3         104   24  55000.0   Finance  2024-02-01    39    400    85     88     90    M
4         105   21  52000.0        HR  2023-01-20    66    350    75     70     88    M
5         106   29  58000.0        IT  2023-01-10    59    390    88     85     76    F
6         107   26      NaN   Finance  2023-01-05    41    420    92     90     84    F
7         108   27  61000.0        IT  2022-12-30    90    460    67     60     79    M
8         109   34  57000.0        HR  2022-12-15    77    480    78     77     83    F
9         110   30  54000.0   Finance  2022-12-01    52    370    86     89     81    M

7. Date column dtype: datetime64[ns]
   employee_id       date
0          101 2022-05-01
1          102 2023-04-15
2          103 2023-03-10
3          104 2024-02-01
4          105 2023-01-20
5          106 2023-01-10
6          107 2023-01-05
7          108 2022-12-30
8          109 2022-12-15
9          110 2022-12-01

8. Total salary by department:
 department
Finance    109000.0
HR         219000.0
IT         119000.0
Name: salary, dtype: float64

9. Sorted by date (descending):
   employee_id       date
3          104 2024-02-01
1          102 2023-04-15
2          103 2023-03-10
4          105 2023-01-20
5          106 2023-01-10
6          107 2023-01-05
7          108 2022-12-30
8          109 2022-12-15
9          110 2022-12-01
0          101 2022-05-01
```

```
10. Added 'total_marks':
   math  science  english  total_marks
0    80       75       70          225
1    70       65       80          215
2    90       95       85          270
3    85       88       90          263
4    75       70       88          233
5    88       85       76          249
6    92       90       84          266
7    67       60       79          206
8    78       77       83          238
9    86       89       81          256
```

```
11. Score between 40 and 80:
   employee_id  age   salary department       date  score  sales  math  science  english gender  total_marks
0          101   22  50000.0         HR 2022-05-01     45    300    80       75       70      M          225
```

```
1          102   35     NaN       IT 2023-04-15     78    450    70      65      80      F         215
4          105   21  52000.0       HR 2023-01-20     66    350    75      70      88      M         233
5          106   29  58000.0       IT 2023-01-10     59    390    88      85      76      F         249
6          107   26     NaN  Finance 2023-01-05     41    420    92      90      84      F         266
8          109   34  57000.0       HR 2022-12-15     77    480    78      77      83      F         238
9          110   30  54000.0  Finance 2022-12-01     52    370    86      89      81      M         256

12. Top 3 by sales:
   employee_id  age   salary department       date  score  sales  math  science  english gender  total_marks
2          103   28  60000.0       HR 2023-03-10     82    500    90      95      85      F         270
8          109   34  57000.0       HR 2022-12-15     77    480    78      77      83      F         238
7          108   27  61000.0       IT 2022-12-30     90    460    67      60      79      M         206

13. Salary stats:
 mean    55875.0
min     50000.0
max     61000.0
Name: salary, dtype: float64

14. After setting date as index:
            employee_id  age   salary department  score  sales  math  science  english gender  total_marks
date
2022-05-01          101   22  50000.0       HR     45    300    80      75      70      M         225
2023-04-15          102   35      NaN       IT     78    450    70      65      80      F         215
2023-03-10          103   28  60000.0       HR     82    500    90      95      85      F         270
2024-02-01          104   24  55000.0  Finance     39    400    85      88      90      M         263
2023-01-20          105   21  52000.0       HR     66    350    75      70      88      M         233
2023-01-10          106   29  58000.0       IT     59    390    88      85      76      F         249
2023-01-05          107   26      NaN  Finance     41    420    92      90      84      F         266
```

```
2022-12-30      108  27  61000.0      IT    90   460   67     60     79    M        206
2022-12-15      109  34  57000.0      HR    77   480   78     77     83    F        238
2022-12-01      110  30  54000.0   Finance   52   370   86     89     81    M        256
```

15. Gender mapped to numeric:

```
 date
2022-05-01    1
2023-04-15    0
2023-03-10    0
2024-02-01    1
2023-01-20    1
2023-01-10    0
2023-01-05    0
2022-12-30    1
2022-12-15    0
2022-12-01    1
Name: gender, dtype: int64
```

16. 1D array from 0 to 9:

```
 [0 1 2 3 4 5 6 7 8 9]
```

17. 3x3 array with random integers (1–100):

```
 [[34 50 20]
 [74 33 79]
 [88 32 41]]
```

18. Mean and standard deviation of new array [10, 20, 30, 40, 50]:

```
Mean = 30.0 , Std Dev = 14.142135623730951
```

19. New array after replacing values > 50 with 50:

```
 [12 50 34 50 45 22 50]
```

20. Reshaped 1D array to 3x4 matrix:

```
 [[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

21. Number of even elements in array:

```
 4
```

22. Flattened new 2D array to 1D:

Original 2D array:

```
 [[ 5 10 15]
 [20 25 30]]
```

Flattened 1D array:

```
 [ 5 10 15 20 25 30]
```

23. Array of 10 random floats (0 to 1):

```
 [0.93037794 0.55051444 0.2439346  0.02182121 0.73412895 0.92327884
 0.15525785 0.18036578 0.71322449 0.47615968]
```

24. Element-wise multiplication of two arrays:

```
 [[ 5 12]
 [21 32]]
```

25. Index of maximum value in array:

```
 3
```