

Q1. List any five functions of the pandas library with execution.

Sure, here are five common functions in the pandas library with their executions:

1.read\_csv(): Used to read data from a CSV file into a DataFrame.

```
In [ ]: import pandas as pd
df = pd.read_csv('data.csv')
```

2.head(): Displays the first n rows of a DataFrame (default n=5).

```
In [ ]: df.head(10) # Display the first 10 rows of the DataFrame
```

3.groupby(): Groups data based on one or more columns and allows you to perform aggregate functions.

```
In [ ]: grouped_data = df.groupby('Category')['Sales'].sum()
```

4.drop(): Removes specified columns or rows from a DataFrame.

```
In [ ]: df = df.drop(columns=['UnwantedColumn'])
```

5.fillna(): Fills missing values in the DataFrame with a specified value or method.

```
In [ ]: df['Age'].fillna(0, inplace=True) # Fill missing Age values with 0
```

Q2. Given a Pandas DataFrame df with columns 'A', 'B', and 'C', write a Python function to re-index the DataFrame with a new index that starts from 1 and increments by 2 for each row.

You can use the set\_index function to re-index the DataFrame:

```
In [21]: def reindex_dataframe(df):
new_index = range(1, len(df) * 2, 2)
df = df.set_index(pd.Index(new_index))
return df
```

Q3. You have a Pandas DataFrame df with a column named 'Values'. Write a Python function that iterates over the DataFrame and calculates the sum of the first three values in the 'Values' column. The function should print the sum to the console. For example, if the 'Values' column of df contains the values [10, 20, 30, 40, 50], your function should calculate and print the sum of the first three values, which is 60.

```
In [19]: def calculate_sum_of_first_three_values(df):
first_three_values = df['Values'].head(3)
sum_of_first_three = first_three_values.sum()
print("Sum of the first three values:", sum_of_first_three)
```

Q4. Given a Pandas DataFrame df with a column 'Text', write a Python function to create a new column 'Word\_Count' that contains the number of words in each row of the 'Text'

column.

```
In [6]: def count_words_in_text_column(df):  
        df['Word_Count'] = df['Text'].apply(lambda x: len(x.split()))  
        return df
```

Q5. How are DataFrame.size() and DataFrame.shape() different?

DataFrame.size: Returns the total number of elements in the DataFrame, which is equal to the product of the number of rows and the number of columns.

DataFrame.shape: Returns a tuple representing the dimensions of the DataFrame, i.e., the number of rows and the number of columns.

So, DataFrame.size gives you the total number of elements, whereas DataFrame.shape provides the dimensions of the DataFrame.

Q6. Which function of pandas do we use to read an excel file?

You can use the read\_excel() function in pandas to read data from an Excel file into a DataFrame. Here's an example:

```
In [ ]: import pandas as pd  
df = pd.read_excel('data.xlsx')
```

Q7. You have a Pandas DataFrame df that contains a column named 'Email' that contains email addresses in the format 'username@domain.com'. Write a Python function that creates a new column 'Username' in df that contains only the username part of each email address. The username is the part of the email address that appears before the '@' symbol. For example, if the email address is 'john.doe@example.com', the 'Username' column should contain 'john.doe'. Your function should extract the username from each email address and store it in the new 'Username' column.

```
In [ ]:
```

Q8. You have a Pandas DataFrame df with columns 'A', 'B', and 'C'. Write a Python function that selects all rows where the value in column 'A' is greater than 5 and the value in column 'B' is less than 10. The function should return a new DataFrame that contains only the selected rows. For example, if df contains the following values:

A	B	C
0	3	5
1	1	8
2	7	2
2	6	9
4	3	2
3	5	4
9	1	2

Assignment

Data Science Masters

Your function should select the following rows: A B C 1 8 2 7 4 9 1 2 The function should return a new DataFrame that contains only the selected rows.

```
In [8]: def extract_username_from_email(df):
df['Username'] = df['Email'].apply(lambda x: x.split('@')[0])
return df
```

Q9. Given a Pandas DataFrame df with a column 'Values', write a Python function to calculate the mean, median, and standard deviation of the values in the 'Values' column.

```
In [12]: def calculate_stats(df):
mean = df['Values'].mean()
median = df['Values'].median()
std_dev = df['Values'].std()
return mean, median, std_dev
```

Q10. Given a Pandas DataFrame df with a column 'Sales' and a column 'Date', write a Python function to create a new column 'MovingAverage' that contains the moving average of the sales for the past 7 days for each row in the DataFrame. The moving average should be calculated using a window of size 7 and should include the current day.

```
In [11]: def calculate_moving_average(df):
df['MovingAverage'] = df['Sales'].rolling(window=7).mean()
return df
```

Q11. You have a Pandas DataFrame df with a column 'Date'. Write a Python function that creates a new column 'Weekday' in the DataFrame. The 'Weekday' column should contain the weekday name (e.g. Monday, Tuesday) corresponding to each date in the 'Date' column. For example, if df contains the following values: Date 0 2023-01-01 1 2023-01-02 2 2023-01-03 3 2023-01-04 4 2023-01-05 Your function should create the following DataFrame:

```
Date Weekday 0 2023-01-01 Sunday 1 2023-01-02 Monday 2 2023-01-03 Tuesday 3
2023-01-04 Wednesday 4 2023-01-05 Thursday
```

The function should return the modified DataFrame.

Note: Create your assignment in Jupyter notebook and upload it in GitHub & share that github repository link through your dashboard. Make sure the repository is public.

Assignment

Data Science Masters

```
In [20]: def add_weekday_column(df):
df['Date'] = pd.to_datetime(df['Date'])
df['Weekday'] = df['Date'].dt.strftime('%A')
return df
```

Q12. Given a Pandas DataFrame df with a column 'Date' that contains timestamps, write a Python function to select all rows where the date is between '2023-01-01' and '2023-01-31'.

```
In [14]: def select_rows_between_dates(df):
df['Date'] = pd.to_datetime(df['Date'])
```

```
selected_rows = df[(df['Date'] >= '2023-01-01') & (df['Date'] <= '2023-01-31')]  
return selected_rows
```

Q13. To use the basic functions of pandas, what is the first and foremost necessary library that needs to be imported?

The first and foremost library that needs to be imported to use basic functions of pandas is, of course, pandas itself. You import it using the following statement:

```
In [ ]: import pandas as pd
```

This statement allows you to access all the functionality provided by the pandas library using the alias pd.