

Answer 1

List Any 5 functions of the pandas library with the execution.

Let's call a data first

```
In [2]: import pandas as pd  
df = pd.read_csv("police.csv")
```

head() Function

```
In [3]: df.head() #it present the first 5 data inputs to us
```

```
Out[3]:
```

	stop_date	stop_time	county_name	driver_gender	driver_age_raw	driver_age	driver_ra
0	2005-01-02	01:55	NaN	M	1985.0	20.0	Wh
1	2005-01-18	08:15	NaN	M	1965.0	40.0	Wh
2	2005-01-23	23:15	NaN	M	1972.0	33.0	Wh
3	2005-02-20	17:15	NaN	M	1986.0	19.0	Wh
4	2005-03-14	10:00	NaN	F	1984.0	21.0	Wh

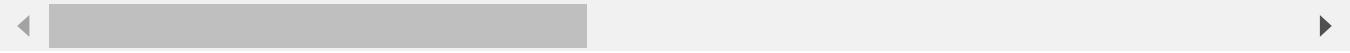


tail() Function

```
In [4]: df.tail() #It presents the last 5 data inputs to us
```

Out[4]:

	stop_date	stop_time	county_name	driver_gender	driver_age_raw	driver_age	driv
91736	2015-12-31	20:27	NaN	M	1986.0	29.0	
91737	2015-12-31	20:35	NaN	F	1982.0	33.0	
91738	2015-12-31	20:45	NaN	M	1992.0	23.0	
91739	2015-12-31	21:42	NaN	M	1993.0	22.0	
91740	2015-12-31	22:46	NaN	M	1959.0	56.0	H



dtypes Function

In [5]: `df.dtypes #it shows us the data type of all the columns`

Out[5]:

stop_date	object
stop_time	object
county_name	float64
driver_gender	object
driver_age_raw	float64
driver_age	float64
driver_race	object
violation_raw	object
violation	object
search_conducted	bool
search_type	object
stop_outcome	object
is_arrested	object
stop_duration	object
drugs_related_stop	bool
dtype:	object

describe() function

In [6]: `df.describe() # It gives us an specific analysis of the data`

Out[6]:

	county_name	driver_age_raw	driver_age
count	0.0	86414.000000	86120.000000
mean	NaN	1970.491228	34.011333
std	NaN	110.914909	12.738564
min	NaN	0.000000	15.000000
25%	NaN	1967.000000	23.000000
50%	NaN	1980.000000	31.000000
75%	NaN	1987.000000	43.000000
max	NaN	8801.000000	99.000000

nunique() Function

In [7]: `df.nunique() # It tells us the number of unique values a column holds`

Out[7]:

stop_date	3768
stop_time	1436
county_name	0
driver_gender	2
driver_age_raw	97
driver_age	78
driver_race	5
violation_raw	12
violation	6
search_conducted	2
search_type	24
stop_outcome	6
is_arrested	2
stop_duration	5
drugs_related_stop	2
dtype: int64	

isnull() Function

In [8]: `df.isnull() # It shows us if the value is null or non-null (for null = True & for`

Out[8]:

	stop_date	stop_time	county_name	driver_gender	driver_age_raw	driver_age	driv
0	False	False	True	False	False	False	False
1	False	False	True	False	False	False	False
2	False	False	True	False	False	False	False
3	False	False	True	False	False	False	False
4	False	False	True	False	False	False	False
...
91736	False	False	True	False	False	False	False
91737	False	False	True	False	False	False	False
91738	False	False	True	False	False	False	False
91739	False	False	True	False	False	False	False
91740	False	False	True	False	False	False	False

91741 rows × 15 columns



Answer 2

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.

In [11]:

```
import pandas as pd

data = {"A": [1,2,3,4,5],
        "B": [4,5,6,7,1],
        "C": [7,8,9,4,3]}

df = pd.DataFrame(data)
```

In [12]:

```
df
```

```
Out[12]:   A  B  C
0    1  4  7
1    2  5  8
2    3  6  9
3    4  7  4
4    5  1  3
```

```
In [13]: new_index = [0,2,4]
```

```
In [14]: df.reindex(new_index)
```

```
Out[14]:   A  B  C
0    1  4  7
2    3  6  9
4    5  1  3
```

Answer 3

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.

```
In [16]: import pandas as pd
data = {"Values" : [1,2,3,4,5,6,7,8,9]}
df = pd.DataFrame(data)
```

```
In [17]: df
```

Out[17]:

	Values
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9

In [18]:

```
total = df["Values"][0:3].sum()
total
```

Out[18]:

Answer 4

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 [16]:

```
import pandas as pd
```

In [17]:

```
data = {"Text" : ["hare krishna", "Nitai gaur is the only truth", "gaurhari"]}
df = pd.DataFrame(data)
```

In [18]:

```
df
```

Out[18]:

	Text
0	hare krishna
1	Nitai gaur is the only truth
2	gaurhari

In [19]:

```
df["Word_Count"] = df["Text"].str.count(" ") + 1
```

In [20]:

```
df
```

Out[20]:

	Text	Word_Count
0	hare krishna	2
1	Nitai gaur is the only truth	6
2	gaurhari	1

Answer 5

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

```
In [21]: data1 = {"a": [1, 2, 3],
              "b": [2, 3, 4],
              "c": [3, 4, 5]}

df = pd.DataFrame(data1)
df
```

```
Out[21]:   a  b  c
0  1  2  3
1  2  3  4
2  3  4  5
```

```
In [22]: df.shape
```

```
Out[22]: (3, 3)
```

```
In [23]: df.size
```

```
Out[23]: 9
```

From the

Dataframe.size tells the number of elements in the dataframe.

Dataframe.shape tells the rows and columns of our dataframe.

Answer 6

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

Answer : pd.read_excel function helps us to read an excel file.

Answer 7

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.

```
In [24]: data = {"Email" : ["harshit@gmail.com", "radha.krishna@gmail.com", "gaurnitai@gmail.com"]}

df1 = pd.DataFrame(data)
```

```
In [25]: df1
```

```
Out[25]:
```

	Email
0	harshit@gmail.com
1	radha.krishna@gmail.com
2	gaurnitai@gmail.com
3	haribol@gmail.com

```
In [26]: df1["Username"] = df1["Email"].str.removesuffix("@gmail.com")
```

```
In [27]: df1
```

```
Out[27]:
```

	Email	Username
0	harshit@gmail.com	harshit
1	radha.krishna@gmail.com	radha.krishna
2	gaurnitai@gmail.com	gaurnitai
3	haribol@gmail.com	haribol

Answer 8

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.

```
In [23]: import pandas as pd

data1 = {"A" : [3,8,6,2,9],
         "B" : [5,2,9,3,1],
         "C" : [1,7,4,5,2]}
df = pd.DataFrame(data1)
```

```
In [24]: df
```

```
Out[24]:   A  B  C
```

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

```
In [25]: try :
    print(df[(df["A"] >5) & (df["B"] <10)])
except Exception as e:
    print("error", e)
```

	A	B	C
1	8	2	7
2	6	9	4
4	9	1	2

```
In [27]: df4 = df[(df["A"] >5) & (df["B"] <10)]
df4
```

```
Out[27]:   A  B  C
```

	A	B	C
1	8	2	7
2	6	9	4
4	9	1	2

Answer 9

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 [31]: data = {"Values" : [1,2,3,4,6,5]}

df = pd.DataFrame(data)
```

```
In [32]: df
```

```
Out[32]:
```

	Values
0	1
1	2
2	3
3	4
4	6
5	5

```
In [33]: df["Values"].mean()
```

```
Out[33]: 3.5
```

```
In [34]: df["Values"].median()
```

```
Out[34]: 3.5
```

```
In [35]: df["Values"].std()
```

```
Out[35]: 1.8708286933869707
```

Answer 10

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 [19]: data = {"Sales" : [1,2,3,4,5,6,7,8,9,11,22,33,44,55,77,66,88],
              "Date" : [2,3,4,5,6,7,8,9,0,33,44,55,66,77,88,99,11]}

df = pd.DataFrame(data)
```

```
In [20]: df
```

Out[20]:

	Sales	Date
0	1	2
1	2	3
2	3	4
3	4	5
4	5	6
5	6	7
6	7	8
7	8	9
8	9	0
9	11	33
10	22	44
11	33	55
12	44	66
13	55	77
14	77	88
15	66	99
16	88	11

In [21]: `df["MovingAverage"] = df["Sales"].rolling(7).mean()`In [22]: `df`

Out[22]:

	Sales	Date	MovingAverage
0	1	2	NaN
1	2	3	NaN
2	3	4	NaN
3	4	5	NaN
4	5	6	NaN
5	6	7	NaN
6	7	8	4.000000
7	8	9	5.000000
8	9	0	6.000000
9	11	33	7.142857
10	22	44	9.714286
11	33	55	13.714286
12	44	66	19.142857
13	55	77	26.000000
14	77	88	35.857143
15	66	99	44.000000
16	88	11	55.000000

Answer 11

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.

```
In [28]: data = {"Date": ["2023-01-01", "2023-01-02", "2023-01-03", "2023-01-04", "2023-01-05"],  
           "day": ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday"]}  
df = pd.DataFrame(data)  
df
```

Out[28]:

	Date
0	2023-01-01
1	2023-01-02
2	2023-01-03
3	2023-01-04
4	2023-01-05

In [29]: df["Weekday"] = day

In [30]: df

Out[30]:

	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

Answer 12

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 [31]: data = {"Date" : ["2023-01-01", "2023-01-02", "2023-01-03", "2023-01-04", "2023-01-05"]}
df = pd.DataFrame(data)
df

Out[31]:

Date

0	2023-01-01
1	2023-01-02
2	2023-01-03
3	2023-01-04
4	2023-01-34
5	2023-01-25
6	2023-01-30
7	2023-01-31

In [39]:

`btw = df[df["Date"].between("2023-01-02", "2023-01-30")]`

In [40]:

`btw`

Out[40]:

Date

1	2023-01-02
2	2023-01-03
3	2023-01-04
5	2023-01-25
6	2023-01-30

Answer 13

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

To use the basic functions of pandas, The first and foremost necessary library that need to be imported is "pandas" itself.

In [44]:

`import pandas as pd`