Assignment

Name: : Chukka Chanakya Devendra

Roll Number: : AP21110011577

Submission Date: : 07-08-2024

Question

1. Write a Python program to create a DataFrame containing the following columns: `StudentID`, `Name`, `Sub1`, `Sub2`, `Sub3`, `Sub4`, `Sub5`, `Sub6`. The DataFrame should have at least 20 rows of data.

   Hint: Use the pandas library to create DataFrame, and numpy for generate scores

1. Check if there are any missing values in the DataFrame. If there are, fill the missing values in numerical columns with the mean of the respective columns.
2. Write a Python program to check whether the student passed or failed (<35) in the courses
   Input: Student ID, Sub1, Sub 2, Sub 3…. Sub N.
   Output: Student is passed/failed according to given data.
3. Filter out students who have an AverageScore less than 60 and display the resulting DataFrame.
4. Write a Pandas program to convert a dictionary to a Pandas series.

```
Sample Series:
Original dictionary:
{'a': 10, 'b': 20, 'c': 40, 'd': 50, 'e': 70}

Converted series:
a    10
b    20
c    40
d    50
e    70
dtype: int64
```

In [ ]:
```python
import pandas as pd
import numpy as np
import names_generator as ng
```

Question 1

In [ ]:
```python
# Create the DataFrame with 20 rows of data
data = {
    'StudentID': [f'S{str(i).zfill(4)}' for i in range(1, 21)],
    'Name': [ng.generate_name() for i in range(1, 21)],
    'Sub1': np.random.randint(30, 100, size=20),
    'Sub2': np.random.randint(30, 100, size=20),
    'Sub3': np.random.randint(30, 100, size=20),
    'Sub4': np.random.randint(30, 100, size=20),
    'Sub5': np.random.randint(30, 100, size=20),
    'Sub6': np.random.randint(30, 100, size=20)
}

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

Out[ ]:

	StudentID	Name	Sub1	Sub2	Sub3	Sub4	Sub5	Sub6
0	S0001	cool_allen	48	60	84	43	42	53
1	S0002	gifted_bell	37	85	62	60	88	49
2	S0003	dreamy_murdock	89	58	84	57	39	68
3	S0004	relaxed_chaplygin	33	38	54	45	72	94
4	S0005	frosty_bhabha	34	80	77	81	65	56
5	S0006	affectionate_margulis	90	86	70	85	89	40
6	S0007	nervous_burnell	69	67	34	54	32	44
7	S0008	hardcore_snyder	52	94	66	65	83	31
8	S0009	amazing_jepsen	55	59	69	76	30	90
9	S0010	suspicious_leakey	70	72	98	65	68	65
10	S0011	objective_villani	78	70	95	76	78	56
11	S0012	sweet_ellis	56	37	40	60	88	57
12	S0013	elated_gould	56	99	78	71	74	50
13	S0014	fervent_mccarthy	71	65	98	75	89	34
14	S0015	relaxed_darwin	30	82	94	60	31	97
15	S0016	serene_bouman	69	81	58	91	64	48
16	S0017	nice_euclid	38	60	46	83	68	56
17	S0018	fervent_herschel	68	47	59	54	96	41
18	S0019	upbeat_joliot	46	84	79	39	72	33
19	S0020	competent_mirzakhani	53	87	36	46	59	77

Question 1

In [ ]:
```
print("Original DataFrame with missing values:")
print(df)
print("\nChecking for missing values...")
print(df.isnull().sum())

df.fillna(df.mean(), inplace=True)
print("\nDataFrame after filling missing values:")
print(df)
```

```
Original DataFrame with missing values:
   StudentID                   Name  Sub1  Sub2  Sub3  Sub4  Sub5  Sub6
0      S0001             cool_allen    48    60    84    43    42    53
1      S0002            gifted_bell    37    85    62    60    88    49
2      S0003         dreamy_murdock    89    58    84    57    39    68
3      S0004      relaxed_chaplygin    33    38    54    45    72    94
4      S0005          frosty_bhabha    34    80    77    81    65    56
5      S0006  affectionate_margulis    90    86    70    85    89    40
6      S0007        nervous_burnell    69    67    34    54    32    44
7      S0008        hardcore_snyder    52    94    66    65    83    31
8      S0009         amazing_jepsen    55    59    69    76    30    90
9      S0010      suspicious_leakey    70    72    98    65    68    65
10     S0011      objective_villani    78    70    95    76    78    56
11     S0012            sweet_ellis    56    37    40    60    88    57
12     S0013           elated_gould    56    99    78    71    74    50
13     S0014       fervent_mccarthy    71    65    98    75    89    34
14     S0015         relaxed_darwin    30    82    94    60    31    97
15     S0016          serene_bouman    69    81    58    91    64    48
16     S0017            nice_euclid    38    60    46    83    68    56
17     S0018       fervent_herschel    68    47    59    54    96    41
18     S0019          upbeat_joliot    46    84    79    39    72    33
19     S0020   competent_mirzakhani    53    87    36    46    59    77

Checking for missing values...
StudentID    0
Name         0
Sub1         0
Sub2         0
Sub3         0
Sub4         0
Sub5         0
Sub6         0
dtype: int64

DataFrame after filling missing values:
   StudentID                   Name  Sub1  Sub2  Sub3  Sub4  Sub5  Sub6
0      S0001             cool_allen    48    60    84    43    42    53
1      S0002            gifted_bell    37    85    62    60    88    49
2      S0003         dreamy_murdock    89    58    84    57    39    68
3      S0004      relaxed_chaplygin    33    38    54    45    72    94
4      S0005          frosty_bhabha    34    80    77    81    65    56
5      S0006  affectionate_margulis    90    86    70    85    89    40
6      S0007        nervous_burnell    69    67    34    54    32    44
7      S0008        hardcore_snyder    52    94    66    65    83    31
8      S0009         amazing_jepsen    55    59    69    76    30    90
9      S0010      suspicious_leakey    70    72    98    65    68    65
10     S0011      objective_villani    78    70    95    76    78    56
11     S0012            sweet_ellis    56    37    40    60    88    57
12     S0013           elated_gould    56    99    78    71    74    50
13     S0014       fervent_mccarthy    71    65    98    75    89    34
14     S0015         relaxed_darwin    30    82    94    60    31    97
15     S0016          serene_bouman    69    81    58    91    64    48
16     S0017            nice_euclid    38    60    46    83    68    56
17     S0018       fervent_herschel    68    47    59    54    96    41
18     S0019          upbeat_joliot    46    84    79    39    72    33
19     S0020   competent_mirzakhani    53    87    36    46    59    77
```

```
C:\temp\ipykernel_11736\2953684037.py:6: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
  df.fillna(df.mean(), inplace=True)
```

Question 2

In [ ]:

```python
def check_pass_fail(row):
    return ['Passed' if score >= 35 else 'Failed' for score in row]

df[['Sub1_status', 'Sub2_status', 'Sub3_status', 'Sub4_status', 'Sub5_status', 'Sub6_status']] = df[['Sub1', 'Sub2', 'Sub3', 'Sub
print("\nDataFrame with pass/fail status:")
print(df)
```

```
DataFrame with pass/fail status:
   StudentID                   Name  Sub1  Sub2  Sub3  Sub4  Sub5  Sub6  \
0      S0001             cool_allen    48    60    84    43    42    53
1      S0002            gifted_bell    37    85    62    60    88    49
2      S0003         dreamy_murdock    89    58    84    57    39    68
3      S0004      relaxed_chaplygin    33    38    54    45    72    94
4      S0005          frosty_bhabha    34    80    77    81    65    56
5      S0006  affectionate_margulis    90    86    70    85    89    40
6      S0007        nervous_burnell    69    67    34    54    32    44
7      S0008        hardcore_snyder    52    94    66    65    83    31
8      S0009         amazing_jepsen    55    59    69    76    30    90
9      S0010      suspicious_leakey    70    72    98    65    68    65
10     S0011      objective_villani    78    70    95    76    78    56
11     S0012            sweet_ellis    56    37    40    60    88    57
12     S0013           elated_gould    56    99    78    71    74    50
13     S0014       fervent_mccarthy    71    65    98    75    89    34
14     S0015         relaxed_darwin    30    82    94    60    31    97
15     S0016          serene_bouman    69    81    58    91    64    48
16     S0017            nice_euclid    38    60    46    83    68    56
17     S0018       fervent_herschel    68    47    59    54    96    41
18     S0019          upbeat_joliot    46    84    79    39    72    33
19     S0020   competent_mirzakhani    53    87    36    46    59    77

   Sub1_status Sub2_status Sub3_status Sub4_status Sub5_status Sub6_status
0       Passed      Passed      Passed      Passed      Passed      Passed
1       Passed      Passed      Passed      Passed      Passed      Passed
2       Passed      Passed      Passed      Passed      Passed      Passed
3       Failed      Passed      Passed      Passed      Passed      Passed
4       Failed      Passed      Passed      Passed      Passed      Passed
5       Passed      Passed      Passed      Passed      Passed      Passed
6       Passed      Passed      Failed      Passed      Failed      Passed
7       Passed      Passed      Passed      Passed      Passed      Failed
8       Passed      Passed      Passed      Passed      Failed      Passed
9       Passed      Passed      Passed      Passed      Passed      Passed
10      Passed      Passed      Passed      Passed      Passed      Passed
11      Passed      Passed      Passed      Passed      Passed      Passed
12      Passed      Passed      Passed      Passed      Passed      Passed
13      Passed      Passed      Passed      Passed      Passed      Failed
14      Failed      Passed      Passed      Passed      Failed      Passed
15      Passed      Passed      Passed      Passed      Passed      Passed
16      Passed      Passed      Passed      Passed      Passed      Passed
17      Passed      Passed      Passed      Passed      Passed      Passed
18      Passed      Passed      Passed      Passed      Passed      Failed
19      Passed      Passed      Passed      Passed      Passed      Passed
```

Question 3

In [ ]:

```python
df['AverageScore'] = df[['Sub1', 'Sub2', 'Sub3', 'Sub4', 'Sub5', 'Sub6']].mean(axis=1)

filtered_df1 = df[df['AverageScore'] < 60]
print("\nFiltered DataFrame with students having average score < 60:")
print(filtered_df1)

filtered_df2 = df[df['AverageScore'] >= 60]
print("\nFiltered DataFrame with students having average score >= 60:")
print(filtered_df2)
```

```
Filtered DataFrame with students having average score < 60:
   StudentID                  Name  Sub1  Sub2  Sub3  Sub4  Sub5  Sub6  \
0      S0001            cool_allen    48    60    84    43    42    53
3      S0004     relaxed_chaplygin    33    38    54    45    72    94
6      S0007       nervous_burnell    69    67    34    54    32    44
11     S0012           sweet_ellis    56    37    40    60    88    57
16     S0017           nice_euclid    38    60    46    83    68    56
18     S0019         upbeat_joliot    46    84    79    39    72    33
19     S0020  competent_mirzakhani    53    87    36    46    59    77

   Sub1_status Sub2_status Sub3_status Sub4_status Sub5_status Sub6_status  \
0       Passed      Passed      Passed      Passed      Passed      Passed
3       Failed      Passed      Passed      Passed      Passed      Passed
6       Passed      Passed      Failed      Passed      Failed      Passed
11      Passed      Passed      Passed      Passed      Passed      Passed
16      Passed      Passed      Passed      Passed      Passed      Passed
18      Passed      Passed      Passed      Passed      Passed      Failed
19      Passed      Passed      Passed      Passed      Passed      Passed

    AverageScore
0      55.000000
3      56.000000
6      50.000000
11     56.333333
16     58.500000
18     58.833333
19     59.666667

Filtered DataFrame with students having average score >= 60:
   StudentID                   Name  Sub1  Sub2  Sub3  Sub4  Sub5  Sub6  \
1      S0002            gifted_bell    37    85    62    60    88    49
2      S0003         dreamy_murdock    89    58    84    57    39    68
4      S0005          frosty_bhabha    34    80    77    81    65    56
5      S0006  affectionate_margulis    90    86    70    85    89    40
7      S0008        hardcore_snyder    52    94    66    65    83    31
8      S0009         amazing_jepsen    55    59    69    76    30    90
9      S0010      suspicious_leakey    70    72    98    65    68    65
10     S0011      objective_villani    78    70    95    76    78    56
12     S0013           elated_gould    56    99    78    71    74    50
13     S0014       fervent_mccarthy    71    65    98    75    89    34
14     S0015         relaxed_darwin    30    82    94    60    31    97
15     S0016          serene_bouman    69    81    58    91    64    48
17     S0018       fervent_herschel    68    47    59    54    96    41

   Sub1_status Sub2_status Sub3_status Sub4_status Sub5_status Sub6_status  \
1       Passed      Passed      Passed      Passed      Passed      Passed
2       Passed      Passed      Passed      Passed      Passed      Passed
4       Failed      Passed      Passed      Passed      Passed      Passed
5       Passed      Passed      Passed      Passed      Passed      Passed
7       Passed      Passed      Passed      Passed      Passed      Failed
8       Passed      Passed      Passed      Passed      Failed      Passed
9       Passed      Passed      Passed      Passed      Passed      Passed
10      Passed      Passed      Passed      Passed      Passed      Passed
12      Passed      Passed      Passed      Passed      Passed      Passed
13      Passed      Passed      Passed      Passed      Passed      Failed
14      Failed      Passed      Passed      Passed      Failed      Passed
15      Passed      Passed      Passed      Passed      Passed      Passed
17      Passed      Passed      Passed      Passed      Passed      Passed

    AverageScore
1      63.500000
2      65.833333
4      65.500000
5      76.666667
7      65.166667
8      63.166667
9      73.000000
10     75.500000
12     71.333333
13     72.000000
14     65.666667
15     68.500000
17     60.833333
```

Question 4

In [ ]:
```
dictionary = {'Name1': "Itadori", 'Name2': "Gojo", 'Name3': "Fushiguro", 'Name4': "Sukuna", 'Name5': "Toji "}
series = pd.Series(dictionary)
print("\nConverted Series from dictionary:")
print(series)
```

```
Converted Series from dictionary:
Name1      Itadori
Name2         Gojo
Name3    Fushiguro
Name4       Sukuna
Name5        Toji
dtype: object
```