simpli·learn

Assignment: Storing Test Results

The comments/sections provided are your cues to perform the assignment. You don't need to limit yourself to the number of rows/cells provided. You can add additional rows in each section to add more lines of code.

If at any point in time you need help on solving this assignment, view our demo video to understand the different steps of the code.

Happy coding!

Storing Test Results

Problem Statement:

From the raw data below create a data frame:

'first_name': ['Jason', 'Molly', 'Tina', 'Jake', 'Amy'],

'last_name': ['Miller', 'Jacobson', ".", 'Milner', 'Cooze'],

'age': [42, 52, 36, 24, 73], 'preTestScore': [4, 24, 31, ".", "."],

'postTestScore': ["25,000", "94,000", 57, 62, 70]

Objective:

Perform data processing on raw data:

- Save the data frame into a csv file as project.csv
- Read the project.csv and print the data frame
- Read the project.csv without column heading
- Read the project.csv and make the index columns as 'First Name' and 'Last Name'
- Print the data frame in a Boolean form as True or False. True for Null/ NaN values and false for non-null values
- Read the data frame by skipping first 3 rows and print the data frame

Table of Contents

In [1]:
```python
# Impor the Pandas Library
import pandas as pd

#Raw Data
raw_data = {'first_name': ['Jason', 'Molly', 'Tina', 'Jake', 'Amy'],
        'last_name': ['Miller', 'Jacobson', ".", 'Milner', 'Cooze'],
        'age': [42, 52, 36, 24, 73],
        'preTestScore': [4, 24, 31, ".", "."],
        'postTestScore': ["25,000", "94,000", 57, 62, 70]}

#Create a new DataFrame using Raw Data
df = pd.DataFrame(raw_data, columns = ['first_name', 'last_name', 'age', 'preTestScore
print(df)
```

```
  first_name last_name  age preTestScore postTestScore
0      Jason    Miller   42            4        25,000
1      Molly  Jacobson   52           24        94,000
2       Tina         .   36           31            57
3       Jake    Milner   24            .            62
4        Amy     Cooze   73            .            70
```

Save the data frame into a csv file as project.csv

In [2]:
```python
#Save the DataFrame using 'to_csv'
df.to_csv('project.csv')
df
```

Out[2]:

	first_name	last_name	age	preTestScore	postTestScore
0	Jason	Miller	42	4	25,000
1	Molly	Jacobson	52	24	94,000
2	Tina	.	36	31	57
3	Jake	Milner	24	.	62
4	Amy	Cooze	73	.	70

Read the project.csv and print the data frame

In [3]:
```python
#Read the csv file using 'read_csv'
df = pd.read_csv('project.csv')
print(df)
```

```
   Unnamed: 0 first_name last_name  age preTestScore postTestScore
0           0      Jason    Miller   42            4        25,000
1           1      Molly  Jacobson   52           24        94,000
2           2       Tina         .   36           31            57
3           3       Jake    Milner   24            .            62
4           4        Amy     Cooze   73            .            70
```

Read the project.csv without column heading

In [4]:
```python
#Read the csv file without Column heading
df = pd.read_csv('project.csv', header=None)
print(df)
```

```
     0           1          2    3             4             5
0  NaN  first_name  last_name  age  preTestScore  postTestScore
1  0.0       Jason     Miller   42             4        25,000
2  1.0       Molly   Jacobson   52            24        94,000
3  2.0        Tina          .   36            31            57
4  3.0        Jake     Milner   24             .            62
5  4.0         Amy      Cooze   73             .            70
```

Read the project.csv and make the index columns as 'First Name' and 'Last Name'

In [5]:
```python
#Read the csv file with index columns
df = pd.read_csv('project.csv', index_col=['First Name', 'Last Name'], names=['UID',
print(df)
```

```
                      UID  Age Pre-Test Score Post-Test Score
First Name Last Name
first_name last_name  NaN  age   preTestScore   postTestScore
Jason      Miller     0.0   42              4         25,000
Molly      Jacobson   1.0   52             24         94,000
Tina       .          2.0   36             31             57
Jake       Milner     3.0   24              .             62
Amy        Cooze      4.0   73              .             70
```

Print the data frame in a Boolean form as True or False. True for Null/ NaN values and false for non-null values

In [6]:
```python
#Print the DataFrame in Boolean Form
df = pd.read_csv('project.csv', na_values=['.'])
print(pd.isnull(df))
```

```
   Unnamed: 0  first_name  last_name    age  preTestScore  postTestScore
0       False       False      False  False         False          False
1       False       False      False  False         False          False
2       False       False       True  False         False          False
3       False       False      False  False          True          False
4       False       False      False  False          True          False
```

Read the data frame by skipping first 3 rows and print the data frame

In [7]:
```python
#Read the DataFrame by skipping first 3 rows
df = pd.read_csv('project.csv', skiprows=3)
print(df)
```

```
   2  Tina      .  36  31  57
0  3  Jake  Milner  24   .  62
1  4   Amy   Cooze  73   .  70
```