MUHYDEEN OLAKUNLE ADELANI
PANDAS PROJECT ONE
FROM BASIC TO ADVANCE

Q1

Console 1

```
Python 3.8.3 (default, Jul  2 2020, 16:21:59)
Type 'copyright', 'credits' or 'license' for more information
IPython 7.16.1 -- An enhanced Interactive Python. Type '?' for help.
```

```
[1]: import pandas
     mydata = {
       'cars': ["BMW", "Volvo", "Ford"],
       'passings': [3, 7, 2]
     }

     mvar = pandas.DataFrame(mydata)

     print(mvar)

        cars  passings
     0   BMW         3
     1  Volvo        7
     2   Ford        2
```

Q2

Create a simple Pandas Series from a list

```
[2]: import pandas as pd

     a = [1, 5, 10, 7, 8, 9,2]

     mvar = pd.Series(a)

     print(mvar)

     0     1
     1     5
     2    10
     3     7
     4     8
     5     9
     6     2
     dtype: int64
```

Q3

Using the label access first and fifth numbers.

[3]:
```
import pandas as pd

a = [1, 5, 10, 7, 8, 9,2]

mvar = pd.Series(a)

print(mvar[0])
```

```
1
```

[4]:
```
import pandas as pd

a = [1, 5, 10, 7, 8, 9,2]

mvar = pd.Series(a)

print(mvar[5])
```

```
9
```

Q4

Create you own labels for this list

[5]:
```
import pandas as pd

a = [1, 5, 10, 7, 8, 9, 2]

mvar = pd.Series(a, index = ["m", "n", "o", "p", "q", "r", "s"])

print(mvar)
```

```
m     1
n     5
o    10
p     7
q     8
r     9
s     2
dtype: int64
```

Q5

Create a simple Pandas Series from a dictionary

[6]:
```
import pandas as pd

cal = {"week1": 120, "week2": 170, "week3": 220}

mvar = pd.Series(cal)

print(mvar)
```

```
week1    120
week2    170
week3    220
dtype: int64
```

Q6

Create a Series using only data from "week1" and "week3"

```
[7]: import pandas as pd

     cal = {"week1": 120, "week2": 170, "week3": 220}

     mvar = pd.Series(cal, index = ["week1", "week3"])

     print(mvar)
```

```
week1    120
week3    220
dtype: int64
```

Q7
Create a
from three

DataFrame
Series

```
[8]: import pandas as pd

     data = {
       "cal": [120, 480, 600],
       "dur": [25, 20, 60],
       "tak": [50, 70, 89]
     }

     mvar = pd.DataFrame(data)

     print(mvar)
```

```
   cal  dur  tak
0  120   25   50
1  480   20   70
2  600   60   89
```

Q8
Create a
Pandas

simple

DataFrame from question 7

```
[9]: import pandas as pd

     data = {
       "cal": [120, 480, 600],
       "dur": [25, 20, 60],
       "tak": [50, 70, 89]
     }

     df = pd.DataFrame(data)

     print(df)
```

```
   cal  dur  tak
0  120   25   50
1  480   20   70
2  600   60   89
```

Q9

From question 8, return row 2 and row 0 with 1

Q10
Using

```
[10]: print(df.loc[2])

[11]: print(df.loc[[0, 1]])

      cal  dur  tak
0  120   25   50
1  480   20   70
```

question 8 data, create a label

```
[12]: import pandas as pd

data = {
  "cal": [120, 480, 600],
  "dur": [25, 20, 60],
  "tak": [50, 70, 89]
}

df = pd.DataFrame(data, index = ["week1", "week2", "week3"])

print(df)

       cal  dur  tak
week1  120   25   50
week2  480   20   70
week3  600   60   89
```

Q11
Return week 3 from Q10 data

```
[13]: print(df.loc["week3"])

cal    600
dur     60
tak     89
Name: week3, dtype: int64
```

Q12
Using Pandas
load a comma

separated file (CSV file) into a DataFrame

```
[1]: import pandas as pd

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

     print(df)
```

```
      FNAME      LNAME          CELL                   EMAIL SALUTATION  \
0      John      Dodge  555-888-8888       jdodge@example.com        Mr.
1     Sarah      Smith  555-999-8888       ssnith@example.com        Ms.
2    George     Dapper  555-999-0000      gdapper@example.com        Dr.
3      Jack      Dodge  555-888-8888      jdodge2@example.com        Mr.
4   Jessica      Jones  555-999-8888       jjones@example.com       Mrs.
5    Dragon     Davich  555-999-0000      dragonr@example.com        Dr.
6     Steve      Curio  555-555-5454      curious@example.com        Mr.
7     Laura      Black  555-555-3434        black@example.com        Ms.
8   Wilhelm      Blake  555-555-5543       wblake@example.com        Mr.
9    Winter      Drake  555-555-5554     dragon2@example.com        Mr.
10      Sal      Sands  555-555-5553  sandssquared@example.com        Mr.
11    Julia      Wilds  555-555-9988       jwilds@example.com        Ms.
12  Juniper     Birsch  555-555-5432       woodsy@example.com       Mrs.
13  Valerie     Duncan  555-555-3433      vduncan@example.com        Ms.
14    Patti      Patel  555-555-4334       ppatel@example.com       Mrs.
15   Justin      Short  555-445-3343       jshort@example.com        Mr.
16     John     Downes  555-444-0117       bungie@example.com        Mr.
17   Xavier      Birch  555-000-0333          xjb@example.com        Mr.
18    Basil  Wenceslas  555-666-3433        hansa@example.com   Chairman
19    Alice      Black  555-555-5555    blackalice@example.com        Ms.

                          TITLE
0                   SVP of Sales
1                Senior Sales Rep
2          Director of Healthcare
3                   EVP of Sales
4                Senior Sales Rep
5   Senior Member of Medical Staff
6                         Janitor
7                       Sales Rep
8                             CEO
9                             CFO
10                            CTO
11                            CMO
12         Director of Engineering
13  Senior Member, Technical Staff
14    Head of Software Engineering
15                           Actor
16                  Master Chief
17             Chair of Genealogy
18                        Chairman
19                        Botanist
```

Q13

Using the given dataset, load it with the Pandas to DataFrame and show all the data in the csv

```
[2]: import pandas as pd

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

     print(df.to_string())
```

```
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
5          60    102       127     300.0
6          60    110       136     374.0
7          45    104       134     253.3
8          30    109       133     195.1
9          60     98       124     269.0
10         60    103       147     329.3
11         60    100       120     250.7
12         60    106       128     345.3
13         60    104       132     379.3
14         60     98       123     275.0
15         60     98       120     215.2
16         60    100       120     300.0
17         45     90       112       NaN
18         60    103       123     323.0
19         45     97       125     243.0
20         60    108       131     364.2
```

Q14

From Q13, only show part of the data

```
[3]: import pandas as pd

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

     print(df)
```

```
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
..        ...    ...       ...       ...
164        60    105       140     290.8
165        60    110       145     300.0
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4

[169 rows x 4 columns]
```

Q15

Load the given JSON file into a DataFrame

```
[4]: import pandas as pd

     df = pd.read_json('data.js')

     print(df.to_string())
```

```
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
5          60    102       127     300.5
6          60    110       136     374.0
7          45    104       134     253.3
8          30    109       133     195.1
9          60     98       124     269.0
10         60    103       147     329.3
11         60    100       120     250.7
12         60    106       128     345.3
13         60    104       132     379.3
14         60     98       123     275.0
15         60     98       120     215.2
16         60    100       120     300.0
17         45     90       112       NaN
18         60    103       123     323.0
19         45     97       125     243.0
20         60    108       131     364.2
21         45    100       119     282.0
```

Q16

Display the first 10 rows from the data.csv and data.js files

```
[6]: import pandas as pd

     df = pd.read_csv('data.csv')
```

```
[7]: import pandas as pd

     df = pd.read_json('data.js')

     print(df.head(10))
```

```
   Duration  Pulse  Maxpulse  Calories
0        60    110       130     409.1
1        60    117       145     479.0
2        60    103       135     340.0
3        45    109       175     282.4
4        45    117       148     406.0
```

Q17

Print the last five rows from Q16

```
[8]: print(df.tail())

     Duration  Pulse  Maxpulse  Calories
164        60    105       140     290.8
165        60    110       145     300.4
166        60    115       145     310.2
167        75    120       150     320.4
168        75    125       150     330.4
```

Q18

Run a command to display details about the data.csv file and explain your result.

```
[9]: print(df.info())

<class 'pandas.core.frame.DataFrame'>
Int64Index: 169 entries, 0 to 168
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Duration  169 non-null    int64
 1   Pulse     169 non-null    int64
 2   Maxpulse  169 non-null    int64
 3   Calories  164 non-null    float64
dtypes: float64(1), int64(3)
memory usage: 6.6 KB
None
```

The result tells us there are 169 rows and 4 columns, and the name of each column, with the data type. In our data set it seems like there are 164 of 169 Non-Null values in the "Calories" column. Which means that there are 5 rows with no value at all, in the "Calories" column.

Q19

From data.csv file, remove the empty rows the file an return the new DataFrame.

Q20

From data.csv, remove all rows with NULL values

```
[10]: import pandas as pd

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

      new_df = df.dropna()

      print(new_df.to_string())

     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
```

```
[11]: import pandas as pd

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

      df.dropna(inplace = True)

      print(df.to_string())

     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
```

Notice in the result that some rows

have been removed (row 17 and 27). These rows had cells with empty values.

Q21

Form data.csv, replace the NULL value with 130 and show the syntax

```
[12]: import pandas as pd
      df = pd.read_csv('data.csv')
      df.fillna(130, inplace = True)
```

Q22

Replace NULL values in the "Calories" columns with the number 150 from the data.csv, and show the syntax.

```
[16]: import pandas as pd
      df = pd.read_csv('data.csv')
      df["Calories"].fillna(150, inplace = True)
```

Q23

Calculate the MEAN, and replace any empty values with it in the given data set

```
[17]: import pandas as pd
      df = pd.read_csv('data.csv')
      x = df["Calories"].mean()
      df["Calories"].fillna(x, inplace = True)
```

Q24

Calculate the MEDIAN, and replace any empty values in data.csv by showing syntax

```
[20]: import pandas as pd
      df = pd.read_csv('data.csv')
      x = df["Calories"].median()
      df["Calories"].fillna(x, inplace = True)
```

Q25

Calculate the MODE, and replace any empty values with it in data.csv file by showing syntax

```
[21]: import pandas as pd
      df = pd.read_csv('data.csv')
      x = df["Calories"].mode()[0]
      df["Calories"].fillna(x, inplace = True)
```

Q26

In the dirtydata.csv, under Date column, there is wrong format. Write a Pandas code to correct that.

Q27 Remove rows with a NULL value in "Date" from the column Q26

```
[3]: import pandas as pd

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

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

     print(df.to_string())

    Duration       Date  Pulse  Maxpulse  Calories
0         60 2020-12-01    110       130     409.1
1         60 2020-12-02    117       145     479.0
2         60 2020-12-03    103       135     340.0
3         45 2020-12-04    109       175     282.4
4    __   45 2020-12-05__ _117   __   148  ___406.0_____

[4]:
     df.dropna(subset=['Date'], inplace = True)

9         60 2020-12-10     98       124     269.0
10        60 2020-12-11    103       147     329.3
11        60 2020-12-12    100       120     250.7
12        60 2020-12-12    100       120     250.7
13        60 2020-12-13    106       128     345.3
14        60 2020-12-14    104       132     379.3
15        60 2020-12-15     98       123     275.0
16        60 2020-12-16     98       120     215.2
17        60 2020-12-17    100       120     300.0
18        45 2020-12-18     90       112       NaN
19        60 2020-12-19    103       123     323.0
20        45 2020-12-20     97       125     243.0
21        60 2020-12-21    108       131     364.2
22        45        NaT    100       119     282.0
23        60 2020-12-23    130       101     300.0
24        45 2020-12-24    105       132     246.0
25
26  [5]: df.loc[7, 'Duration'] = 45
27
28 ____________________________________________________
29        60 2020-12-29    100       132     280.0
30        60 2020-12-30    102       129     380.3
31        60 2020-12-31     92       115     243.0
```

Q28 From under there is error in correct replacing with 45 Q26, Duration typing row 7, that by the value

Q29 Returns True for every row that is a duplicate, othwerwise False in data.csv

```
[6]: print(df.duplicated())

     0     False
     1     False
     2     False
     3     False
     4     False
     5     False
     6     False
     7     False
     8     False
     9     False
     10    False
     11    False
     12     True
     13    False
     14    False
     15    False
     16    False
     17    False
     18    False
     19    False
     20    False
     21    False
     23    False
     24    False
     25    False
```

Q30

Remove all duplicate from data.csv file.

```
[8]: import pandas as pd

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

     df.drop_duplicates(inplace = True)

     print(df.to_string())
```

```
     Duration  Pulse  Maxpulse  Calories
0          60    110       130     409.1
1          60    117       145     479.0
2          60    103       135     340.0
3          45    109       175     282.4
4          45    117       148     406.0
5          60    102       127     300.0
6          60    110       136     374.0
7          45    104       134     253.3
8          30    109       133     195.1
9          60     98       124     269.0
10         60    103       147     329.3
11         60    100       120     250.7
12         60    106       128     345.3
13         60    104       132     379.3
14         60     98       123     275.0
15         60     98       120     215.2
16         60    100       120     300.0
17         45     90       112       NaN
18         60    103       123     323.0
19         45     97       125     243.0
20         60    108       131     364.2
21         45    100       119     282.0
22         60    130       101     300.0
23         45    105       132     246.0
24         60    102       126     334.5
25         60    100       120     250.0
26         60     92       118     241.0
27         60    103       132       NaN
28         60    100       132     280.0
```

Q31

Calculate the correlation in data.csv file.

```
[9]: import pandas as pd

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

     print(df.corr())
```

```
              Duration     Pulse  Maxpulse  Calories
Duration      1.000000 -0.155408  0.009403  0.922717
Pulse        -0.155408  1.000000  0.786535  0.025121
Maxpulse      0.009403  0.786535  1.000000  0.203813
Calories      0.922717  0.025121  0.203813  1.000000
```

Q32

Visualize the data.csv file using Pandas-Plotting

[10]:
```
import pandas as pd
import matplotlib.pyplot as plt

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

df.plot()

plt.show()
```