# Usama Arif Rollno 14

# Task 1: Load data as a data frame display first five and last five entities.

In [68]:
```
1 import pandas as pd
2 df=pd.read_excel('Final Evaluation.xlsx')
3 df
```

Out[68]:

| | Unnamed: 0 | Unnamed: 1 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 | Unnamed: 5 | Unnamed: 6 | Unna |
|---|---|---|---|---|---|---|---|---|
| 0 | NaN | ASSIGNMENTS EVALUATION | NaN | NaN | NaN | NaN | NaN | |
| 1 | NaN | Sr. No. | Roll No. | Name of Student | 2022-08-03 00:00:00 | 2022-09-03 00:00:00 | 2022-10-03 00:00:00 | 14/3 |
| 2 | NaN | 1 | AI21-C1-01 | Rabiya Mubeen | 10 | 10 | 10 | |
| 3 | NaN | 2 | AI21-C1-02 | Sehrish Nazir | 10 | 10 | 8 | |
| 4 | NaN | 3 | AI21-C1-03 | Areeba Amin | 0 | NaN | 2 | |
| 5 | NaN | 4 | AI21-C1-04 | Khadija Amin | 10 | 10 | 10 | |
| 6 | NaN | 5 | AI21-C1- | Sadia | 0 | 0 | NaN | |

# first five

In [69]:
```
1 df.head(5)
```

Out[69]:

| | Unnamed: 0 | Unnamed: 1 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 | Unnamed: 5 | Unnamed: 6 | Unnamed 7 |
|---|---|---|---|---|---|---|---|---|
| **0** | NaN | ASSIGNMENTS EVALUATION | NaN | NaN | NaN | NaN | NaN | NaN |
| **1** | NaN | Sr. No. | Roll No. | Name of Student | 2022-08-03 00:00:00 | 2022-09-03 00:00:00 | 2022-10-03 00:00:00 | 14/3/2022 |
| **2** | NaN | 1 | AI21-C1-01 | Rabiya Mubeen | 10 | 10 | 10 | 10 |
| **3** | NaN | 2 | AI21-C1-02 | Sehrish Nazir | 10 | 10 | 8 | 10 |
| **4** | NaN | 3 | AI21-C1-03 | Areeba Amin | 0 | NaN | 2 | NaN |

5 rows × 21 columns

# last five

In [70]:
```
1 df.tail(5)
```

Out[70]:

| | Unnamed: 0 | Unnamed: 1 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 | Unnamed: 5 | Unnamed: 6 | Unnamed: 7 |
|---|---|---|---|---|---|---|---|---|
| **48** | NaN | 47 | AI21-C2-22 | Huzaifa Tariq Butt | 10 | 0 | 10 | 0 |
| **49** | NaN | 48 | AI21-C2-23 | Mukarram Munir | 0 | 10 | NaN | 10 |
| **50** | NaN | 49 | AI21-C2-24 | Shahadat Ali | 0 | NaN | 4 | 10 |
| **51** | NaN | 50 | AI21-C2-25 | Mohammad Awais | 10 | 10 | 10 | 10 |
| **52** | NaN | NaN | NaN | NaN | 1 | 288 | 338 | 298 |

5 rows × 21 columns

# Task 2:Load all spread sheets in different dataframe and combine all of them in a single one.

In [71]:
```
excel_file=pd.read_excel('Final Evaluation.xlsx')
csv_file=pd.read_csv('Automobile_data.csv')
combined_file=pd.concat((excel_file,csv_file),axis=0)
new_File=combined_file.to_excel('new_File.xlsx')
#loading new combined file
pd.read_excel('new_File.xlsx')
```

Out[71]:

| | Unnamed: 0.1 | Unnamed: 0 | Unnamed: 1 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 | Unnamed: 5 | Unnam |
|---|---|---|---|---|---|---|---|---|
| **0** | 0 | NaN | ASSIGNMENTS EVALUATION | NaN | NaN | NaN | NaN | N |
| **1** | 1 | NaN | Sr. No. | Roll No. | Name of Student | 2022-08-03 00:00:00 | 2022-09-03 00:00:00 | 2022- 00:00 |
| **2** | 2 | NaN | 1 | AI21-C1-01 | Rabiya Mubeen | 10 | 10 | |
| **3** | 3 | NaN | 2 | AI21-C1-02 | Sehrish Nazir | 10 | 10 | |
| **4** | 4 | NaN | 3 | AI21-C1-03 | Areeba Amin | 0 | NaN | |
| **...** | ... | ... | ... | ... | ... | ... | ... | |
| **109** | 56 | NaN | NaN | NaN | NaN | NaN | NaN | N |
| **110** | 57 | NaN | NaN | NaN | NaN | NaN | NaN | N |
| **111** | 58 | NaN | NaN | NaN | NaN | NaN | NaN | N |
| **112** | 59 | NaN | NaN | NaN | NaN | NaN | NaN | N |
| **113** | 60 | NaN | NaN | NaN | NaN | NaN | NaN | N |

114 rows × 32 columns

# Task 3:Mark zero in place of missing data.

In [72]:
```
df=df.fillna(0)
df              #data is filled with 0
```

Out[72]:

| | Unnamed: 0 | Unnamed: 1 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 | Unnamed: 5 | Unnamed: 6 | Unna |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | ASSIGNMENTS EVALUATION | 0 | 0 | 0 | 0 | 0 | |
| 1 | 0.0 | Sr. No. | Roll No. | Name of Student | 2022-08-03 00:00:00 | 2022-09-03 00:00:00 | 2022-10-03 00:00:00 | 14/3 |
| 2 | 0.0 | 1 | AI21-C1-01 | Rabiya Mubeen | 10 | 10 | 10 | |
| 3 | 0.0 | 2 | AI21-C1-02 | Sehrish Nazir | 10 | 10 | 8 | |
| 4 | 0.0 | 3 | AI21-C1-03 | Areeba Amin | 0 | 0 | 2 | |
| 5 | 0.0 | 4 | AI21-C1-04 | Khadija Amin | 10 | 10 | 10 | |
| | | | AI21-C1- | Sadia | | | | |

# Task 4:Mark fail to those students who got less than 50 marks

In [73]:
```
df=df.fillna(0) #filling no numrical value
df['Result']='Nill'
df.iloc[:, 20] = pd.to_numeric(df.iloc[:, 20], errors='coerce')
df.loc[df.iloc[:, 20] < 50, 'Result'] = 'Fail'
df.loc[df.iloc[:,20]>=50,'Result']='Pass'
df
```

```
C:\Users\usama\AppData\Local\Temp\ipykernel_9320\3950582623.py:3: DeprecationWarning: In a future version, `df.iloc[:, i] = newvals` will attempt to set the values inplace instead of always setting a new array. To retain the old behavior, use either `df[df.columns[i]] = newvals` or, if columns are non-unique, `df.isetitem(i, newvals)`
  df.iloc[:, 20] = pd.to_numeric(df.iloc[:, 20], errors='coerce')
```

Out[73]:

| | Unnamed: 0 | Unnamed: 1 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 | Unnamed: 5 | Unnamed: 6 | Unna |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | ASSIGNMENTS EVALUATION | 0 | 0 | 0 | 0 | 0 | |
| 1 | 0.0 | Sr. No. | Roll No. | Name of Student | 2022-08-03 00:00:00 | 2022-09-03 00:00:00 | 2022-10-03 00:00:00 | 14/3 |
| 2 | 0.0 | 1 | AI21-C1-01 | Rabiya Mubeen | 10 | 10 | 10 | |
| 3 | 0.0 | 2 | AI21-C1- | Sehrish | 10 | 10 | 8 | |

# Task 5:

Make a new column as “GRADE” and assign grade according to this manner 90-100 A+ ,80-90 A-,70-80 B,60-70 C,50-59-D,Below 50 Fail

In [74]:
```
df['Grade']='Nill'

```

Out[74]:

| | Unnamed: 0 | Unnamed: 1 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 | Unnamed: 5 | Unnamed: 6 | Unna |
|---|---|---|---|---|---|---|---|---|
| **0** | 0.0 | ASSIGNMENTS EVALUATION | 0 | 0 | 0 | 0 | 0 | |
| **1** | 0.0 | Sr. No. | Roll No. | Name of Student | 2022-08-03 00:00:00 | 2022-09-03 00:00:00 | 2022-10-03 00:00:00 | 14/3 |
| **2** | 0.0 | 1 | AI21-C1-01 | Rabiya Mubeen | 10 | 10 | 10 | |
| **3** | 0.0 | 2 | AI21-C1-02 | Sehrish Nazir | 10 | 10 | 8 | |
| **4** | 0.0 | 3 | AI21-C1-03 | Areeba Amin | 0 | 0 | 2 | |
| **5** | 0.0 | 4 | AI21-C1-04 | Khadija Amin | 10 | 10 | 10 | |
| | | | AI21-C1- | Sadia | | | | |

In [80]:
```
df.fillna(0)
f=pd.read_excel('Final Evaluation.xlsx')
df.iloc[:, 20] = pd.to_numeric(df.iloc[:, 20], errors='coerce')
df['Grade'] = 'Nill'

df.loc[df.iloc[:, 20] <= 50, 'Grade'] = 'A+'
df.loc[df.iloc[:, 20] < 45, 'Grade'] = 'A'
df.loc[df.iloc[:, 20] < 40, 'Grade'] = 'B'
df.loc[df.iloc[:, 20] < 30, 'Grade'] = 'C'
df.loc[df.iloc[:, 20] < 20, 'Grade'] = 'D'
df
```

Out[80]:

| | Unnamed: 0 | Unnamed: 1 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 | Unnamed: 5 | Unnamed: 6 | Unna |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | ASSIGNMENTS EVALUATION | 0 | 0 | 0 | 0 | 0 | |
| 1 | 0.0 | Sr. No. | Roll No. | Name of Student | 2022-08-03 00:00:00 | 2022-09-03 00:00:00 | 2022-10-03 00:00:00 | 14/3 |
| 2 | 0.0 | 1 | AI21-C1-01 | Rabiya Mubeen | 10 | 10 | 10 | |
| 3 | 0.0 | 2 | AI21-C1-02 | Sehrish Nazir | 10 | 10 | 8 | |
| 4 | 0.0 | 3 | AI21-C1-03 | Areeba Amin | 0 | 0 | 2 | |
| 5 | 0.0 | 4 | AI21-C1-04 | Khadija Amin | 10 | 10 | 10 | |
| | | | AI21-C1- | Sadia | | | | |

# Task 6:Rank your data frame with respect to total marks.

In [81]:
```
#Task 6:
#Rank your data frame with respect to total marks.
import pandas as pd
import numpy as np
df =df.sort_values(by = 'Unnamed: 20',ascending = False)
df.reset_index(drop=True, inplace=True)
df.to_excel('df_sorted.xlsx',index = False)
df

```

Out[81]:

| | Unnamed: 0 | Unnamed: 1 | Unnamed: 2 | Unnamed: 3 | Unnamed: 4 | Unnamed: 5 | Unnamed: 6 | Unna |
|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | 1 | AI21-C1-01 | Rabiya Mubeen | 10 | 10 | 10 | |
| 1 | 0.0 | 50 | AI21-C2-25 | Mohammad Awais | 10 | 10 | 10 | |
| 2 | 0.0 | 6 | AI21-C1-06 | Amna Naveed | 10 | 10 | 10 | |
| 3 | 0.0 | 24 | AI21-C1-24 | Abdul Ahad Shahzad | 10 | 10 | 10 | |
| 4 | 0.0 | 23 | AI21-C1-23 | Ch. Bilal Ur Rehman Sandhu | 10 | 10 | 10 | |
| 5 | 0.0 | 20 | AI21-C1-20 | Muhammad Ibtisam Ahmad | 10 | 10 | 10 | |

# Task 7:Calculate the average of total marks

In [85]:
```
df.fillna(0)
print('mean',df.iloc[:,20].mean())
```

```
mean 29.096153846153847
```

# Task 8:Display the name of first three position holders.

In [86]:
```
df.iloc[1:5,3]
```

Out[86]:
```
1                Mohammad Awais
2                  Amna Naveed
3             Abdul Ahad Shahzad
4    Ch. Bilal Ur Rehman Sandhu
Name: Unnamed: 3, dtype: object
```

# Task 9:Save your data in excel file as a single spreadsheet

In [ ]:
```
df.to_excel('fileIMP.xlsx',index = False)
```

# Part B

# Task 1: From the given dataset print the first and last five rows.

In [98]:
```
df=pd.read_csv('Automobile_data.csv')
df.head(5)
```

Out[98]:

| | index | company | body-style | wheel-base | length | engine-type | num-of-cylinders | horsepower | average-mileage | price |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | alfa-romero | convertible | 88.6 | 168.8 | dohc | four | 111 | 21 | 13495 |
| **1** | 1 | alfa-romero | convertible | 88.6 | 168.8 | dohc | four | 111 | 21 | 16500 |
| **2** | 2 | alfa-romero | hatchback | 94.5 | 171.2 | ohcv | six | 154 | 19 | 16500 |
| **3** | 3 | audi | sedan | 99.8 | 176.6 | ohc | four | 102 | 24 | 13950 |
| **4** | 4 | audi | sedan | 99.4 | 176.6 | ohc | five | 115 | 18 | n.a |

In [99]:
```
df.tail(5)
```

Out[99]:

| | index | company | body-style | wheel-base | length | engine-type | num-of-cylinders | horsepower | average-mileage | price |
|---|---|---|---|---|---|---|---|---|---|---|
| **56** | 81 | volkswagen | sedan | 97.3 | 171.7 | ohc | four | 85 | 27 | 7975 |
| **57** | 82 | volkswagen | sedan | 97.3 | 171.7 | ohc | four | 52 | 37 | 7995 |
| **58** | 86 | volkswagen | sedan | 97.3 | 171.7 | ohc | four | 100 | 26 | 9995 |
| **59** | 87 | volvo | sedan | 104.3 | 188.8 | ohc | four | 114 | 23 | 12940 |
| **60** | 88 | volvo | wagon | 104.3 | 188.8 | ohc | four | 114 | 23 | 13415 |

# Task 2 : Replace all column values which contain ?, n.a, or NaN.

In [101]:
```
df.replace(['n.a','NaN'],0,inplace=True)
df
```

Out[101]:

| | index | company | body-style | wheel-base | length | engine-type | num-of-cylinders | horsepower | average-mileage | pric |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | alfa-romero | convertible | 88.6 | 168.8 | dohc | four | 111 | 21 | 1349 |
| **1** | 1 | alfa-romero | convertible | 88.6 | 168.8 | dohc | four | 111 | 21 | 1650 |
| **2** | 2 | alfa-romero | hatchback | 94.5 | 171.2 | ohcv | six | 154 | 19 | 1650 |
| **3** | 3 | audi | sedan | 99.8 | 176.6 | ohc | four | 102 | 24 | 1395 |
| **4** | 4 | audi | sedan | 99.4 | 176.6 | ohc | five | 115 | 18 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **56** | 81 | volkswagen | sedan | 97.3 | 171.7 | ohc | four | 85 | 27 | 797 |
| **57** | 82 | volkswagen | sedan | 97.3 | 171.7 | ohc | four | 52 | 37 | 799 |
| **58** | 86 | volkswagen | sedan | 97.3 | 171.7 | ohc | four | 100 | 26 | 999 |
| **59** | 87 | volvo | sedan | 104.3 | 188.8 | ohc | four | 114 | 23 | 1294 |
| **60** | 88 | volvo | wagon | 104.3 | 188.8 | ohc | four | 114 | 23 | 1341 |

61 rows × 10 columns

# Task 3: Print most expensive car's company name and price.

In [128]:
```
df=pd.read_csv('Automobile_data.csv',na_values=['n.a'])
max_p=df.loc[df['price'].idxmax()]
car_company=max_p['company']
car_price=max_p['price']

print('car company\n',car_company)
print('car price\n',car_price)
```

```
car company
 mercedes-benz
car price
 45400.0
```

# Task 4: Count total cars per company

In [130]:
```
counts=df['company'].value_counts
print(counts)
```

```
<bound method IndexOpsMixin.value_counts of 0     alfa-romero
1     alfa-romero
2     alfa-romero
3            audi
4            audi
         ...
56     volkswagen
57     volkswagen
58     volkswagen
59          volvo
60          volvo
Name: company, Length: 61, dtype: object>
```

# Task 5: Find each company's Highest price car

In [137]:
```
for index, row in max_price.iterrows():
    comp = row['company']
    maxprc = row['price']
    print("Company: " , comp)
    print("Highest: " , maxprc)

```

```
Company:  alfa-romero
Highest:  16500.0
Company:  audi
Highest:  18920.0
Company:  bmw
Highest:  41315.0
Company:  chevrolet
Highest:  6295.0
Company:  dodge
Highest:  6377.0
Company:  honda
Highest:  12945.0
Company:  isuzu
Highest:  6785.0
Company:  jaguar
Highest:  36000.0
Company:  mazda
Highest:  18344.0
Company:  mercedes-benz
Highest:  45400.0
Company:  mitsubishi
Highest:  8189.0
Company:  nissan
Highest:  13499.0
Company:  porsche
Highest:  37028.0
Company:  toyota
Highest:  15750.0
Company:  volkswagen
Highest:  9995.0
Company:  volvo
Highest:  13415.0
```

# Task 6: Find the average mileage of each car making company

In [138]:
```
import pandas as pd
average = file.groupby('company')['average-mileage'].mean().reset_index()
for index, row in average.iterrows():
    comp = row['company']
    maxprc = row['average-mileage']
    print("Company: " , comp)
    print("Average mileage: " , average)
```

```
Company:  alfa-romero
Average mileage:           company  average-mileage
0     alfa-romero        20.333333
1            audi        20.000000
2             bmw        19.000000
3       chevrolet        41.000000
4           dodge        31.000000
5           honda        26.333333
6           isuzu        33.333333
7          jaguar        14.333333
8           mazda        28.000000
9   mercedes-benz        18.000000
10     mitsubishi        29.500000
11         nissan        31.400000
12        porsche        17.000000
13         toyota        28.714286
14     volkswagen        31.750000
15          volvo        23.000000
Company:  audi
```

# Task 7: Sort all cars by Price column

In [143]:
```
sorted_col=df.sort_values(['price'],ascending=False)
sorted_col
```

Out[143]:

| | index | company | body-style | wheel-base | length | engine-type | num-of-cylinders | horsepower | average-mileage | pri |
|---|---|---|---|---|---|---|---|---|---|---|
| **35** | 47 | mercedes-benz | hardtop | 112.0 | 199.2 | ohcv | eight | 184 | 14 | 45400 |
| **11** | 14 | bmw | sedan | 103.5 | 193.8 | ohc | six | 182 | 16 | 41315 |
| **34** | 46 | mercedes-benz | sedan | 120.9 | 208.1 | ohcv | eight | 184 | 14 | 40960 |
| **46** | 62 | porsche | convertible | 89.5 | 168.9 | ohcf | six | 207 | 17 | 37028 |
| **12** | 15 | bmw | sedan | 110.0 | 197.0 | ohc | six | 182 | 15 | 36880 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **15** | 18 | chevrolet | sedan | 94.5 | 158.8 | ohc | four | 70 | 38 | Na |
| **20** | 29 | honda | sedan | 96.5 | 169.1 | ohc | four | 100 | 25 | Na |
| **22** | 31 | isuzu | sedan | 94.5 | 155.9 | ohc | four | 70 | 38 | Na |
| **23** | 32 | isuzu | sedan | 94.5 | 155.9 | ohc | four | 70 | 38 | Na |
| **47** | 63 | porsche | hatchback | 98.4 | 175.7 | dohcv | eight | 288 | 17 | Na |

61 rows × 10 columns

In [ ]:
```
1
```