In [1]:

```
#Q1

import pandas as pd
```

In [4]:

```
course_name = ['Data Science','Machine Learning','Big Data','Data Engineer']
duration = [2,3,6,4]
df = pd.DataFrame(data = {'course_name' : course_name, 'duration' : duration})
```

In [5]:

```
df
```

Out[5]:

	course_name	duration
0	Data Science	2
1	Machine Learning	3
2	Big Data	6
3	Data Engineer	4

In [35]:

```
df1=df.loc[2:2:,['course_name','duration']]
```

In [38]:

```
print(df1)
```

```
  course_name  duration
2    Big Data         6
```

In [ ]:

```
#Q2

#The loc() function is label based data selecting method which means that we have t
#This method includes the last element of the range passed in it, unlike iloc(). Lo

#The iloc() function is an indexed-based selecting method which means that we have
#This method does not include the last element of the range passed in it unlike loc
```

In [39]:

```
#Q3
import pandas as pd
import numpy as np
columns = ['column_1', 'column_2', 'column_3', 'column_4', 'column_5', 'column_6']
indices = [1,2,3,4,5,6]
#Creating a dataframe:
df1 = pd.DataFrame(np.random.rand(6,6), columns = columns, index = indices)
```

In [55]:

```
df2=df1.reset_index()
```

In [60]:

```
df2
```

Out[60]:

	index	column_1	column_2	column_3	column_4	column_5	column_6
0	1	0.073837	0.247818	0.351250	0.788346	0.064270	0.222501
1	2	0.662983	0.632124	0.303729	0.782460	0.942008	0.877542
2	3	0.945667	0.011179	0.773359	0.060965	0.650647	0.165449
3	4	0.558882	0.175578	0.772523	0.131472	0.029056	0.902447
4	5	0.977657	0.049105	0.857711	0.128770	0.327251	0.402914
5	6	0.517236	0.652354	0.094722	0.341616	0.204269	0.141864

In [57]:

```
new_df=df2.reindex([3,0,1,2])
```

In [58]:

```
new_df.loc[2]
```

Out[58]:

```
index       3.000000
column_1    0.945667
column_2    0.011179
column_3    0.773359
column_4    0.060965
column_5    0.650647
column_6    0.165449
Name: 2, dtype: float64
```

In [59]:

```
new_df.iloc[2]
```

Out[59]:

```
index       2.000000
column_1    0.662983
column_2    0.632124
column_3    0.303729
column_4    0.782460
column_5    0.942008
column_6    0.877542
Name: 1, dtype: float64
```

In [61]:

```
#Iloc method does not take last element of the range but loc method takes last elem
#of the range.
```

In [89]:

```
#Q4


import pandas as pd
import numpy as np
columns = ['column_1', 'column_2', 'column_3', 'column_4', 'column_5', 'column_6']
indices = [1,2,3,4,5,6]
#Creating a dataframe:
df1 = pd.DataFrame(np.random.rand(6,6), columns = columns, index = indices)
```

In [97]:

```
df1
```

Out[97]:

	column_1	column_2	column_3	column_4	column_5	column_6
1	0.792149	0.636017	0.806261	0.369878	0.164785	0.349752
2	0.587950	0.215909	0.608798	0.457786	0.245141	0.127062
3	0.304354	0.808332	0.245999	0.122468	0.277741	0.001888
4	0.362385	0.310034	0.338448	0.522670	0.090103	0.669712
5	0.955414	0.021008	0.787939	0.462172	0.338987	0.806665
6	0.826672	0.788710	0.108915	0.481134	0.153006	0.874174

In [98]:

```
df1[['column_1','column_2','column_3','column_4','column_5','column_6']].mean()
```

Out[98]:

```
column_1    0.638154
column_2    0.463335
column_3    0.482727
column_4    0.402685
column_5    0.211627
column_6    0.471542
dtype: float64
```

In [99]:

```
df1['column_2'].std()
```

Out[99]:

```
0.3271344379317016
```

In [106…

```
#Q5


df1 = {'column_2': ['0.215909','sts' ]}
df3= pd.DataFrame(df1)
```

In [107…

```
df3
```

Out[107]:

	column_2
0	0.215909
1	sts

In [ ]:

```
#Q6

#Pandas Window functions are functions where the input values are taken from a “win
# rolling function:-This function can be applied on a series of data. Specify the w
#expanding functon:-This function can be applied on a series of data. Specify the m

#.ewm():-ewm is applied on a series of data. Specify any of the com, span, halflife
```

In [115…

```
#Q7


# importing date class from datetime module
from datetime import date
```

```
# creating the date object of today's date
todays_date = date.today()

# printing todays date
print("Current date: ", todays_date)

# fetching the current year, month and day of today
print("Current year:", todays_date.year)
print("Current month:", todays_date.month)
print("Current day:", todays_date.day)
```

```
Current date:  2023-09-28
Current year: 2023
Current month: 9
Current day: 28
```

In [16]:

```
#Q8

t1 = pd.to_datetime('1/1/2015 01:00')
t2 = pd.to_datetime('10/1/2015 03:30')

print(pd.Timedelta(t2 - t1))
print(pd.Timedelta(t2 - t1).seconds/60.0)
print(pd.Timedelta(t2 - t1).seconds/3600.0)
```

```
273 days 02:30:00
150.0
2.5
```

In [8]:

```
#Q9


import pandas as pd
import numpy as np
import seaborn as sns
```

In [9]:

```
df=pd.read_csv("penguins.csv")
```

In [10]:

```
df.head(2)
```

Out[10]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	ye
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	male	20
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	female	20

In [11]:

```
df['species'].unique()
```

Out[11]:

```
array(['Adelie', 'Gentoo', 'Chinstrap'], dtype=object)
```

In [12]:

```
df['island'].unique()
```

Out[12]:

```
array(['Torgersen', 'Biscoe', 'Dream'], dtype=object)
```

In [14]:
```
df.columns
```

Out[14]:
```
Index(['species', 'island', 'bill_length_mm', 'bill_depth_mm',
       'flipper_length_mm', 'body_mass_g', 'sex', 'year'],
      dtype='object')
```

In [15]:
```
for col_name in df.columns:
    if(df[col_name].dtype=='object'):
        df[col_name]=df[col_name].astype('category')
        df[col_name]=df[col_name].cat.codes
```

In [16]:
```
df.head(3)
```

Out[16]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex	year
0	0	2	39.1	18.7	181.0	3750.0	1	2007
1	0	2	39.5	17.4	186.0	3800.0	0	2007
2	0	2	40.3	18.0	195.0	3250.0	0	2007

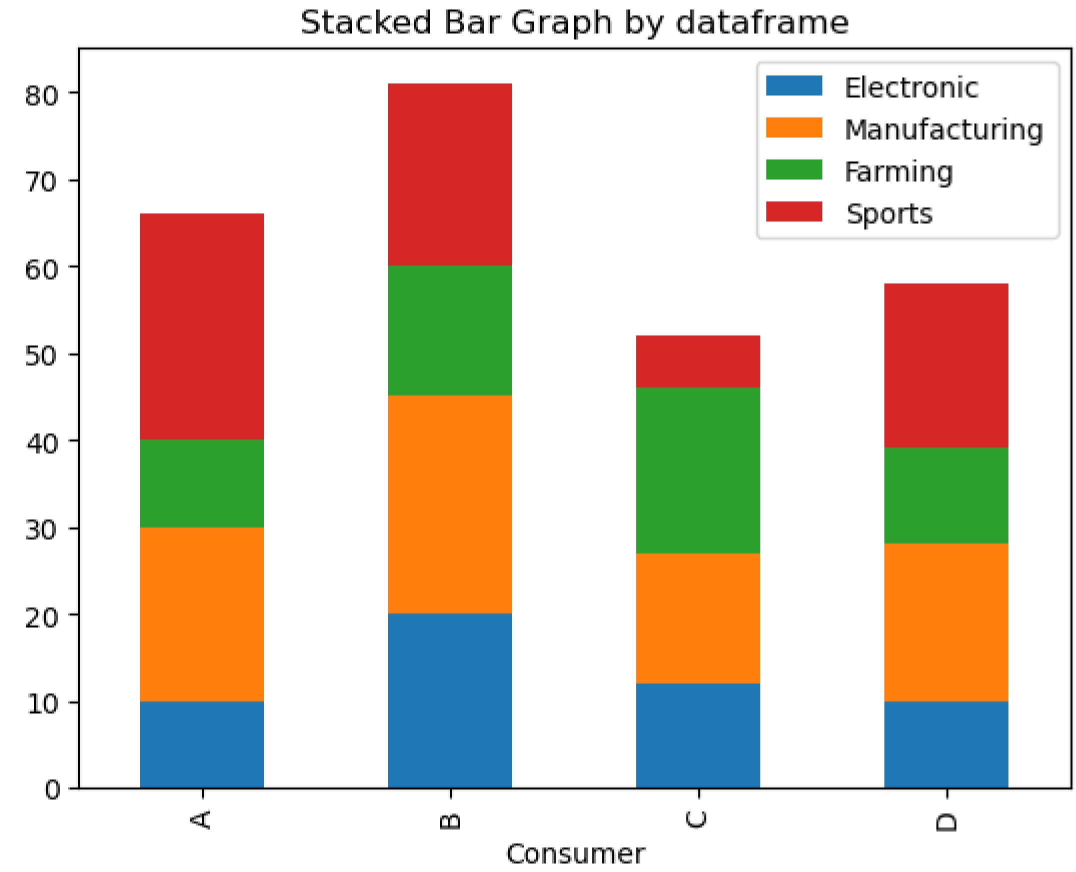
In [31]:
```
#Q10


import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# create data
df = pd.DataFrame([['A', 10, 20, 10, 26], ['B', 20, 25, 15, 21], ['C', 12, 15, 19,
                   ['D', 10, 18, 11, 19]],
                  columns=['Consumer', 'Electronic', 'Manufacturing', 'Farming', 'S
# view data
print(df)

# plot data in stack manner of bar type
df.plot(x='Consumer', kind='bar', stacked=True,
        title='Stacked Bar Graph by dataframe')
plt.show()
```

```
  Consumer  Electronic  Manufacturing  Farming  Sports
0        A          10             20       10      26
1        B          20             25       15      21
2        C          12             15       19       6
3        D          10             18       11      19
```



In [32]:
```
#Q11


import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [33]:
```
df=pd.read_csv("stud.csv")
```

In [35]:
```
df.head(2)
```

Out[35]:

	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_scor
0	female	group B	bachelor's degree	standard	none	7
1	female	group C	some college	standard	completed	6

In [54]:
```
df[['math_score','reading_score','writing_score']].mean()
```

Out[54]:
```
math_score       66.089
reading_score    69.169
writing_score    68.054
dtype: float64
```

In [55]:
```
df[['math_score','reading_score','writing_score']].mode()
```

Out[55]:

	math_score	reading_score	writing_score
0	65	72	74

In [56]:

```
df[['math_score','reading_score','writing_score']].median()
```

Out[56]:

```
math_score       66.0
reading_score    70.0
writing_score    69.0
dtype: float64
```

In [72]:

```
dict={'mean':[66.089,69.169,68.054],
      'mode':[65,72,74],
      'median':[66,70,69]
     }
df = pd.DataFrame(dict)
```

In [79]:

```
df
```

Out[79]:

	mean	mode	median
0	66.089	65	66
1	69.169	72	70
2	68.054	74	69

In [ ]:

```

```