In [2]:	Q1.Create a data frame called DataFrame1. It should contain data about five employees. Each employee data consists of employee number, name, salary, department number and designation. # Creating the dataframe for the Employee import pandas as pd
	<pre>DataFrame1 = pd.DataFrame([['Emp101','Aravindh',150000,'Computer','Data Scientist'],</pre>
Out[2]:	0Emp101Aravindh15000ComputerData Scientist1Emp102Alekhya38000MarketingSenior Manager2Emp103Gayu18000Project LeadProject Head3Emp104Kani21000Project LeadProject Senior Head4Emp105Kavya15000Disposal TeamDisposal Head
In [3]:	Q2.In the DataFrame1 defined in Q.No. 1, add appropriate column names to the columns. Find the mean of the salary in the DataFrame1. DataFrame1.columns = ['Employee Number', 'Employee Name', 'Salary', 'Department Name', 'Designation'] print(DataFrame1) print("\nMean of the Employee Salary: ",DataFrame1.Salary.mean())
	Employee Number Employee Name Salary Department Name Designation 0 Emp101 Aravindh 150000 Computer Data Scientist 1 Emp102 Alekhya 38000 Marketing Senior Manager 2 Emp103 Gayu 18000 Project Lead Project Head 3 Emp104 Kani 21000 Project Lead Project Senior Head 4 Emp105 Kavya 15000 Disposal Team Disposal Head Mean of the Employee Salary: 48400.0
In []:	Q3.Create a dictionary called dept. In this dictionary, the columns are deptno, deptname and location. Create a data frame called DataFrame2 from the dictionary dept.
<pre>In [4]: Out[4]:</pre>	'Department Name' : ('Computer','Marketing','Project Lead'), 'Location' : ('USA','Chennai','Erode')} DataFrame2 = pd.DataFrame(dept) DataFrame2
In []:	
	Q4.Consider a sales.csv file with the following data: Dinesh 7800 6500 4590 Mahesh 8900 3790
	Ramesh10800 2300 Suresh 5600 6790 2300
	Basha 7600 2300 2400 The data represents salesman name and sales in three regions: region1, region2 and region3. Load sles.csv into a data frame called DataFrame3. Give appropriate column names to the columns in DataFrame3. Display the contents of DataFrame3.
<pre>In [5]: Out[5]:</pre>	<pre>import pandas as pd DataFrame3 = pd.read_csv('.\\sales.csv',header = None, names=['Salesman','region1','region2','region3']) DataFrame3 Salesman region1 region2 region3 O Dinesh 7800 6500.0 4590.0</pre>
	1 Mahesh 8900 NaN 3790.0 2 Ramesh 10800 2300.0 NaN 3 Suresh 5600 6790.0 2300.0 4 Basha 7600 2300.0 2400.0
In []:	Q5.In this exercise, demonstrate different methods of handling missing data in data frames. For reference, consider the DataFrame3 created in Q.No. 4. In DataFrame3, do the following: fill missing values with zeroes, fill missing values with the mean of their respective column, drop the rows containing missing values, drop the columns containing missing values.
<pre>In [6]: Out[6]:</pre>	# Missing Values with zeros DataFrame3.fillna(0) Salesman region1 region2 region3 O Dinesh 7800 6500.0 4590.0
In [7]:	1 Mahesh 8900 0.0 3790.0 2 Ramesh 10800 2300.0 0.0 3 Suresh 5600 6790.0 2300.0 4 Basha 7600 2300.0 2400.0 # Fill Missing Values with the mean of their respective columns
Out[7]:	0 Dinesh 7800 6500.0 4590.0
In [8]:	1 Mahesh 8900 4472.5 3790.0 2 Ramesh 10800 2300.0 3270.0 3 Suresh 5600 6790.0 2300.0 4 Basha 7600 2300.0 2400.0 DataFrame3['region1'].fillna(DataFrame3['region1'].mean())
Out[8]:	<pre>0 7800 1 8900 2 10800 3 5600 4 7600 Name: region1, dtype: int64</pre> DataFrame3['region2'].fillna(DataFrame3['region2'].mean())
Out[9]: In [10]:	<pre>0 6500.0 1 4472.5 2 2300.0 3 6790.0 4 2300.0 Name: region2, dtype: float64 DataFrame3['region3'].fillna(DataFrame3['region3'].mean()) 0 4590.0</pre>
Out[10]:	1 3790.0 2 3270.0 3 2300.0 4 2400.0 Name: region3, dtype: float64 # Drop the rows containing mising Values DataFrame3.dropna()
Out[11]: In [12]:	Salesman region1 region2 region3 0 Dinesh 7800 6500.0 4590.0 3 Suresh 5600 6790.0 2300.0 4 Basha 7600 2300.0 2400.0
Out[12]:	DataFrame3.dropna(axis=1) Salesman region1 0 Dinesh 7800 1 Mahesh 8900 2 Ramesh 10800
In []:	3 Suresh 5600 4 Basha 7600 Q6.From DataFrame1 create a new data frame called DataFrame4. In DataFrame4, set the employee
In [13]:	number as an in-place index. Display data from the 3rd row and 4th column using label-based and integer-based indexing. DataFrame4 = DataFrame1 DataFrame4.set_index('Employee Number',inplace=True) DataFrame4 DataFrame4
Out[13]:	Employee NameSalaryDepartment NameDesignationEmployee NumberEmp101Aravindh150000ComputerData ScientistEmp102Alekhya38000MarketingSenior ManagerEmp103Gayu18000Project LeadProject HeadEmp104Kani21000Project LeadProject Senior Head
<pre>In [14]: Out[14]:</pre>	Emp105 Kavya 15000 Disposal Team Disposal Head #Using Label based Index DataFrame4.loc['Emp103','Designation']
<pre>In [15]: Out[15]: In []:</pre>	
<pre>In [16]: Out[16]:</pre>	Q7.In DataFrame3 created in Q.No. 5, find the salesmen whose sales are greater than 7000 in region1 but less than 3000 in region2 # using Boolean index DataFrame3[(DataFrame3['region1'] > 7000) & (DataFrame3['region2'] < 3000)] Salesman region1 region2 region3
In []:	2 Ramesh 10800 2300.0 NaN 4 Basha 7600 2300.0 2400.0 Q8.In DataFrame1 created in Q. No. 1, find the employees with the highest salary in each department.
In [17]:	Use the group by clause and the max() function. Similarly, in DataFrame3 created in Q. No. 5, find the salesman with the highest amount of sales for each region. DataFrame1 DataFrame1.groupby(['Department Name']).max()['Salary']
Out[17]: In [19]:	Department Name Computer 150000 Disposal Team 15000 Marketing 38000 Project Lead 21000 Name: Salary, dtype: int64 DataFrame3.groupby(['Salesman']).max()
Out[19]:	region1 region3 Salesman Basha 7600 2300.0 2400.0 Dinesh 7800 6500.0 4590.0 Mahesh 8900 NaN 3790.0 Ramesh 10800 2300.0 NaN
In [20]: Out[20]:	Suresh 5600 6790.0 2300.0 DataFrame3.groupby(['Salesman']).max()['region1'] Salesman Basha 7600 Dinesh 7800 Mahesh 8900
<pre>In [21]: Out[21]:</pre>	Ramesh 10800 Suresh 5600 Name: region1, dtype: int64 DataFrame3.groupby(['Salesman']).max()['region2'] Salesman Basha 2300.0 Dinesh 6500.0
<pre>In [22]: Out[22]:</pre>	Mahesh NaN Ramesh 2300.0 Suresh 6790.0 Name: region2, dtype: float64 DataFrame3.groupby(['Salesman']).max()['region3'] Salesman Basha 2400.0 Dinesh 4590.0
In []:	Mahesh 3790.0 Ramesh NaN Suresh 2300.0 Name: region3, dtype: float64
<pre>In [18]: Out[18]:</pre>	value as 'Chennai' for all the existing rows DataFrame1['Address'] =DataFrame1.shape[0]*['Chennai'] DataFrame1 Employee Name Salary Department Name Designation Address
	Employee NumberEmp101Aravindh150000ComputerData ScientistChennaiEmp102Alekhya38000MarketingSenior ManagerChennaiEmp103Gayu18000Project LeadProject HeadChennaiEmp104Kani21000Project LeadProject Senior HeadChennaiEmp105Kavya15000Disposal TeamDisposal HeadChennai
In []:	Q10.Create a line graph and bar chart to visualize the two data frames, DataFrame1 and DataFrame3.
Out[23]:	<pre># Using DataFrame1 for line graph import pandas as pd DataFrame1_visu = pd.DataFrame(DataFrame1.groupby(['Employee Name']).mean()['Salary']) DataFrame1_visu.plot()</pre>
	120000 - 100000 - 80000 - 60000 - 40000 -
In [24]:	# Using DataFrame1 for bar graph DataFrame1_visu.plot.bar() <axessubplot:xlabel='employee name'=""></axessubplot:xlabel='employee>
Out[24]:	140000 - 120000 - 100000 - 80000 - 40000 -
In [25]:	
Out[25]:	<pre>DataFrame3_visu = pd.DataFrame(DataFrame3.groupby(['Salesman']).mean()) DataFrame3_visu.plot()</pre>
	6000 - 4000 - 2000
<pre>In [26]: Out[26]:</pre>	# Using DataFrame3 for bar graph DataFrame3_visu.plot.bar() <axessubplot:xlabel='salesman'> region1 # Using DataFrame3 for bar graph DataFrame3_visu.plot.bar()</axessubplot:xlabel='salesman'>
	10000 - region2 region3 -
In []:	Salesman Sales as a series as