Salary Analysis using Python Pandas

In [4]:

```
import pandas as pd
import csv
```

In [5]:

```
data= pd.read_csv(r'C:\Users\KIIT\Desktop\pandas project\Salary project\Salaries.csv')
```

```
C:\Users\KIIT\anaconda3\lib\site-packages\IPython\core\interactiveshell.p
y:3146: DtypeWarning: Columns (3,4,5,6,12) have mixed types.Specify dtype
option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

In [7]:

```
data
```

Out[7]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411	0	400184	NaN
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966	245132	137811	NaN
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739	106088	16452.6	NaN
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916	56120.7	198307	NaN
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	134402	9737	182235	NaN
...	...	...	...	...	...	...	...
148649	148650	Roy I Tillery	Custodian	0.00	0.00	0.00	0.00
148650	148651	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided
148651	148652	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided
148652	148653	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided
148653	148654	Joe Lopez	Counselor, Log Cabin Ranch	0.00	0.00	-618.13	0.00

148654 rows × 13 columns

1. Display Top 10 Rows of the Dataset

In [10]:

```
data.head(10)
```

Out[10]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayB
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411	0	400184	NaN	567595.43	567
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966	245132	137811	NaN	538909.28	538
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739	106088	16452.6	NaN	335279.91	335
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE	77916	56120.7	198307	NaN	332343.61	332

2. Display Last 10 rows of the Dataset

In [11]:

```
data.tail(10)
```

148645	148646	Carolyn A Wilson	Human Services Technician	0.00	0.00	0.00	0.00	0.00
148646	148647	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided	0.00
148647	148648	Joann Anderson	Communications Dispatcher 2	0.00	0.00	0.00	0.00	0.00
148648	148649	Leon Walker	Custodian	0.00	0.00	0.00	0.00	0.00
148649	148650	Roy I Tillery	Custodian	0.00	0.00	0.00	0.00	0.00
148650	148651	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided	0.00
148651	148652	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided	0.00
148652	148653	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided	0.00

3. Find Shape of our Dataset (Number of Rows and Columns)

In [12]:

```
data.shape
```

Out[12]:

```
(148654, 13)
```

In [13]:

```
print("Number of Rows : ", data.shape[0])
print("Number of Columns : ", data.shape[1])
```

```
Number of Rows :  148654
Number of Columns :  13
```

4. Getting Information About our Datasetlike Total Number of Rows,Total Number of coloums,Datatypes of each column and Memory requirement

In [15]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148654 entries, 0 to 148653
Data columns (total 13 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   Id                148654 non-null  int64  
 1   EmployeeName      148654 non-null  object 
 2   JobTitle          148654 non-null  object 
 3   BasePay           148049 non-null  object 
 4   OvertimePay       148654 non-null  object 
 5   OtherPay          148654 non-null  object 
 6   Benefits          112495 non-null  object 
 7   TotalPay          148654 non-null  float64
 8   TotalPayBenefits  148654 non-null  float64
 9   Year              148654 non-null  int64  
 10  Notes             0 non-null       float64
 11  Agency            148654 non-null  object 
 12  Status            38119 non-null   object 
dtypes: float64(3), int64(2), object(8)
memory usage: 14.7+ MB
```

5.Check Null Values in the Dataset .

In [17]:

```
data.isnull().sum()
```

Out[17]:

```
Id                   0
EmployeeName         0
JobTitle             0
BasePay            605
OvertimePay          0
OtherPay             0
Benefits         36159
TotalPay             0
TotalPayBenefits     0
Year                 0
Notes           148654
Agency               0
Status          110535
dtype: int64
```

6. Drop ID, Notes ,Agency and Status Column .

In [19]:

```
data.columns
```

Out[19]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherP
ay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agenc
y',
       'Status'],
      dtype='object')
```

In [24]:

```
data=data.drop(['Id','Notes','Agency','Status'],axis=1)
```

In [25]:

```
data.head(1)
```

Out[25]:

	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	T
0	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411	0	400184	NaN	567595.43	

In [26]:

```
data.head()
```

Out[26]:

	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	T
0	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411	0	400184	NaN	567595.43	
1	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966	245132	137811	NaN	538909.28	
2	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739	106088	16452.6	NaN	335279.91	
3	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916	56120.7	198307	NaN	332343.61	
4	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	134402	9737	182235	NaN	326373.19	

7. Get Overall Statistics About the Dataframe.

In [8]:

```
data.describe(include='all')
```

Out[8]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits
count	148654.000000	148654	148654	148049.0	148654.0	148654.0	112495.0
unique	NaN	110811	2159	109900.0	66555.0	84968.0	99635.0
top	NaN	Kevin Lee	Transit Operator	0.0	0.0	0.0	0.0
freq	NaN	13	7036	875.0	66103.0	35218.0	1053.0
mean	74327.500000	NaN	NaN	NaN	NaN	NaN	NaN
std	42912.857795	NaN	NaN	NaN	NaN	NaN	NaN
min	1.000000	NaN	NaN	NaN	NaN	NaN	NaN
25%	37164.250000	NaN	NaN	NaN	NaN	NaN	NaN
50%	74327.500000	NaN	NaN	NaN	NaN	NaN	NaN
75%	111490.750000	NaN	NaN	NaN	NaN	NaN	NaN
max	148654.000000	NaN	NaN	NaN	NaN	NaN	NaN

8. Find Occurance of the Employee Names (Top 5)

In [9]:

```
data.columns
```

Out[9]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherP
ay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agenc
y',
       'Status'],
      dtype='object')
```

In [10]:

```
data['EmployeeName'].value_counts()
```

Out[10]:

```
Kevin Lee         13
Richard Lee       11
William Wong      11
Steven Lee        11
KEVIN LEE          9
                  ..
HUNG-MING CHU      1
Jordan S Barry     1
Kirstin Walker     1
Dinei Leao         1
YU-LANG CHEN       1
Name: EmployeeName, Length: 110811, dtype: int64
```

In [11]:

```
data['EmployeeName'].value_counts().head()
```

Out[11]:

```
Kevin Lee       13
Richard Lee     11
William Wong    11
Steven Lee      11
KEVIN LEE        9
Name: EmployeeName, dtype: int64
```

9. Find the number of Unique Job Titles .

In [13]:

```
data.columns
```

Out[13]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherP
ay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agenc
y',
       'Status'],
      dtype='object')
```

In [15]:

```
data['JobTitle'].nunique()
```

Out[15]:

```
2159
```

10. Total Number of Job Titles Contain Captain .

In [16]:

```
data.columns
```

Out[16]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherP
ay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agenc
y',
       'Status'],
      dtype='object')
```

In [17]:

```
data['JobTitle']
```

Out[17]:

```
0         GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY
1                        CAPTAIN III (POLICE DEPARTMENT)
2                        CAPTAIN III (POLICE DEPARTMENT)
3                  WIRE ROPE CABLE MAINTENANCE MECHANIC
4            DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)
                              ...
148649                                         Custodian
148650                                     Not provided
148651                                     Not provided
148652                                     Not provided
148653                          Counselor, Log Cabin Ranch
Name: JobTitle, Length: 148654, dtype: object
```

In [19]:

```
data[data['JobTitle'].str.contains('CAPTAIN')]
```

Out[19]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966	245132	137811	NaN
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739	106088	16452.6	NaN
11	12	PATRICIA JACKSON	CAPTAIN III (POLICE DEPARTMENT)	99722	87082.6	110804	NaN
17	18	SEBASTIAN WONG	CAPTAIN, EMERGENCYCY MEDICAL SERVICES	140547	119397	18625.1	NaN
22	23	GEORGE GARCIA	CAPTAIN, FIRE SUPPRESSION	140547	93200.6	39955.2	NaN
...	...	...	...	...	...	...	...
8684	8685	JEANNE SEYLER	CAPTAIN, FIRE SUPPRESSION	95055.3	0	9197.14	NaN
10485	10486	JANE SMITH	CAPTAIN, EMERGENCYCY MEDICAL SERVICES	74592	1538.59	18804.7	NaN
11198	11199	KATHRYN BROWN	CAPTAIN III (POLICE DEPARTMENT)	10684.5	0	81244.9	NaN
31297	31298	MARCO CARNIGLIA	CAPTAIN, EMERGENCYCY MEDICAL SERVICES	9839.72	0	1203.77	NaN
34124	34125	JOHN FORBES-3	CAPTAIN, FIRE SUPPRESSION	0	982.06	2277.34	NaN

141 rows × 13 columns

In [21]:

```
len(data[data['JobTitle'].str.contains('CAPTAIN',case=False)])
```

Out[21]:

552

11. Display All the Employee Name From Fire Department .

In [25]:

```
data.columns
```

Out[25]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherP
ay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agenc
y',
       'Status'],
      dtype='object')
```

In [29]:

```
data[data['JobTitle'].str.contains('fire',case = False)]
```

Out[29]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	134402	9737	182235	NaN
6	7	ALSON LEE	BATTALION CHIEF, (FIRE DEPARTMENT)	92492	89062.9	134426	NaN
8	9	MICHAEL MORRIS	BATTALION CHIEF, (FIRE DEPARTMENT)	176933	86362.7	40132.2	NaN
9	10	JOANNE HAYES-WHITE	CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	285262	0	17115.7	NaN
10	11	ARTHUR KENNEY	ASSISTANT CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	194999	71344.9	33149.9	NaN
...	...	...	...	...	...	...	...
145956	145957	Kenneth C Farris	Firefighter	0.00	0.00	0.00	4645.56
147556	147557	Edward A Dunn	Firefighter	1063.24	0.00	132.90	385.66
148021	148022	Kari A Johnson	Firefighter	688.71	0.00	0.00	143.39
148209	148210	Sheryl K Lee	Firefighter	459.14	0.00	0.00	95.59
148554	148555	Lawrence F Gatt	Fire Alarm Dispatcher	73.33	0.00	0.00	0.73

5879 rows × 13 columns

In [30]:

```
data[data['JobTitle'].str.contains('fire',case = False)]['EmployeeName']
```

Out[30]:

```
4            PATRICK GARDNER
6                  ALSON LEE
8              MICHAEL MORRIS
9          JOANNE HAYES-WHITE
10              ARTHUR KENNEY
                   ...
145956       Kenneth C Farris
147556          Edward A Dunn
148021         Kari A Johnson
148209           Sheryl K Lee
148554        Lawrence F Gatt
Name: EmployeeName, Length: 5879, dtype: object
```

12. Find Maximum , Minimum and Average of TotalPay .

In [31]:

```
data.columns
```

Out[31]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherP
ay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agenc
y',
       'Status'],
      dtype='object')
```

In [43]:

```
data['TotalPay'].describe()
```

Out[43]:

```
count    148654.000000
mean      74768.321972
std       50517.005274
min        -618.130000
25%       36168.995000
50%       71426.610000
75%      105839.135000
max      567595.430000
Name: TotalPay, dtype: float64
```

13. Replace 'Not Provided' in Employeename Column to NaN

In [45]:

```
data.columns
```

Out[45]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherP
ay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agenc
y',
       'Status'],
      dtype='object')
```

In [47]:

```
import numpy as np
data['EmployeeName']=data['EmployeeName'].replace('Not provided',np.nan)
```

In [48]:

```
data['EmployeeName']
```

Out[48]:

```
0            NATHANIEL FORD
1              GARY JIMENEZ
2            ALBERT PARDINI
3         CHRISTOPHER CHONG
4           PATRICK GARDNER
                 ...
148649        Roy I Tillery
148650                  NaN
148651                  NaN
148652                  NaN
148653            Joe Lopez
Name: EmployeeName, Length: 148654, dtype: object
```

14. Drop the Rows having 5 missing Values

In [51]:

```
data.drop(data[data.isnull().sum(axis=1)==5].index,axis=0,inplace=True)
```

In [52]:

```
data.isnull().sum(axis=1)
```

Out[52]:

```
0         3
1         3
2         3
3         3
4         3
         ..
148649    1
148650    3
148651    3
148652    3
148653    1
Length: 148654, dtype: int64
```

15. Find Job Title of ALBERT PARDINI

In [53]:

```
data.columns
```

Out[53]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherP
ay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agenc
y',
       'Status'],
      dtype='object')
```

In [54]:

```
data[data['EmployeeName']=='ALBERT PARDINI']['JobTitle']
```

Out[54]:

```
2    CAPTAIN III (POLICE DEPARTMENT)
Name: JobTitle, dtype: object
```

16. How much ALBERT PARDINI Make (Include Benefits) ?

In [55]:

```
data.columns
```

Out[55]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherP
ay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agenc
y',
       'Status'],
      dtype='object')
```

In [58]:

```
data[data['EmployeeName']=='ALBERT PARDINI']['TotalPayBenefits']
```

Out[58]:

```
2    335279.91
Name: TotalPayBenefits, dtype: float64
```

17 . Display Name of the Person Having The Highest TotalPay.

In [59]:

```
data.columns
```

Out[59]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherP
ay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agenc
y',
       'Status'],
      dtype='object')
```

In [71]:

```
data[data['TotalPay'].max()==data['TotalPay']]['EmployeeName']
```

Out[71]:

```
0    NATHANIEL FORD
Name: EmployeeName, dtype: object
```

18. Find Average TotalPay of All Employee per year

In [72]:

```
data.columns
```

Out[72]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherP
ay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agenc
y',
       'Status'],
      dtype='object')
```

In [74]:

```
data.groupby('Year').mean()['TotalPay']
```

Out[74]:

```
Year
2011    71744.103871
2012    74113.262265
2013    77611.443142
2014    75463.918140
Name: TotalPay, dtype: float64
```

19. Find Average Totalpay of all Employee per JobTitle.

In [75]:

```
data.columns
```

Out[75]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherP
ay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agenc
y',
       'Status'],
      dtype='object')
```

In [76]:

```
data.groupby('JobTitle').mean()['TotalPay']
```

Out[76]:

```
JobTitle
ACCOUNT CLERK                                          44035.664337
ACCOUNTANT                                             47429.268000
ACCOUNTANT INTERN                                      29031.742917
ACPO,JuvP, Juv Prob (SFERS)                            62290.780000
ACUPUNCTURIST                                          67594.400000
                                                           ...
X-RAY LABORATORY AIDE                                  52705.880385
X-Ray Laboratory Aide                                  50823.942700
YOUTH COMMISSION ADVISOR, BOARD OF SUPERVISORS         53632.870000
Youth Comm Advisor                                     41414.307500
ZOO CURATOR                                            66686.560000
Name: TotalPay, Length: 2159, dtype: float64
```

20. Find Average TotalPay of Employee having Job Title ACCOUNTANT

In [77]:

```
data.columns
```

Out[77]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherP
ay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agenc
y',
       'Status'],
      dtype='object')
```

In [78]:

```
data[data['JobTitle']=='ACCOUNTANT']['BasePay'].mean()
```

Out[78]:

```
46643.172
```

21. Find Top 5 Most Common Jobs

In [79]:

```
data.columns
```

Out[79]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherP
ay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agenc
y',
       'Status'],
      dtype='object')
```

In [81]:

```
data['JobTitle'].value_counts().head()
```

Out[81]:

```
Transit Operator                7036
Special Nurse                   4389
Registered Nurse                3736
Public Svc Aide-Public Works    2518
Police Officer 3                2421
Name: JobTitle, dtype: int64
```

In [ ]:

```

```