Section A: Data handling with pandas

1. Import pandas as pd.

In [1]:

```
import pandas as pd
```

2. Read Salaries.csv as a dataframe called sal.

In [2]:

```
sal=pd.read_csv('Salaries.csv')
sal
```

```
C:\Users\hp\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:314
6: 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,
```

Out[2]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411	0	400184	NaN	5
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966	245132	137811	NaN	5
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739	106088	16452.6	NaN	3
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916	56120.7	198307	NaN	3
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	134402	9737	182235	NaN	3
...	...	...	...	...	...	...	...	
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

3. Check the head of the DataFrame.

In [3]:

```
sal.head() #column name with first 5 rows
```

Out[3]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411	0	400184	NaN	567595.43
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966	245132	137811	NaN	538909.28
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739	106088	16452.6	NaN	335279.91
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916	56120.7	198307	NaN	332343.61
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	134402	9737	182235	NaN	326373.19

4. Use the .info() method to find out how many entries there are.

In [4]:

```
sal.info() #it will show record of each column
```

```
<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. What is the average BasePay ?

In [5]:

```
sal['BasePay'] = pd.to_numeric(sal['BasePay'],errors='coerce')
sal['BasePay'].mean()
```

Out[5]:

```
66325.44884050643
```

6. What is the highest amount of OvertimePay in the dataset?

In [6]:

```
sal['OvertimePay'] = pd.to_numeric(sal['OvertimePay'],errors='coerce')
max(sal['OvertimePay'])
```

Out[6]:

```
245131.88
```

7. What is the job title of JOSEPH DRISCOLL?

In [7]:

```
new = sal[(sal['EmployeeName'] == "JOSEPH DRISCOLL")]
new["JobTitle"]
```

Out[7]:

```
24    CAPTAIN, FIRE SUPPRESSION
Name: JobTitle, dtype: object
```

8. How much does JOSEPH DRISCOLL make (including benefits)?

In [8]:

```
new['TotalPayBenefits']
```

Out[8]:

```
24    270324.91
Name: TotalPayBenefits, dtype: float64
```

9. What is the name of highest paid person (including benefits)?

In [9]:

```
sal[sal['TotalPayBenefits'] == max(sal['TotalPayBenefits'])]["EmployeeName"]
```

Out[9]:

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

10. What is the name of lowest paid person (including benefits)?

In [10]:

```
sal[sal['TotalPayBenefits'] == sal['TotalPayBenefits'].min()]['EmployeeName']
```

Out[10]:

```
148653    Joe Lopez
Name: EmployeeName, dtype: object
```

11. What was the average (mean) BasePay of all employees per year? (2011-2014) ?

In [11]:

```
sal.groupby('Year').mean()['BasePay']
```

Out[11]:

```
Year
2011    63595.956517
2012    65436.406857
2013    69630.030216
2014    66564.421924
Name: BasePay, dtype: float64
```

12. How many unique job titles are there? ¶

In [12]:

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

Out[12]:

```
2159
```

13. What are the top 5 most common jobs?

In [13]:

```
sal['JobTitle'].value_counts().head(5)
```

Out[13]:

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

14. How many Job Titles were represented by only one person in 2013? (e.g. Job Titles with only one occurrence in 2013?)

In [14]:

```
sum(sal[sal['Year']==2013]['JobTitle'].value_counts() == 1)
```

Out[14]:

```
202
```

15. How many people have the word Chief in their job title?

In [15]:

```
def chief_string(title):
    if "chief" in title.lower().split():
        return True
    else:
        return False
sum(sal["JobTitle"].apply(lambda x:chief_string(x)))
```

Out[15]:

```
477
```

16. Is there a correlation between length of the Job Title string and Salary?

In [16]:

```
sal['Title_len']=sal['JobTitle'].apply(len)
sal[['Title_len','TotalPayBenefits']].corr()
```

Out[16]:

	Title_len	TotalPayBenefits
Title_len	1.000000	-0.036878
TotalPayBenefits	-0.036878	1.000000