Practical 4

Aim: Write a program to perform following operation

- Load the data from file
- Find out null and missing value
- Handle missing Value using different approach Plot the data using scatter plot, histogram, box plot

In [2]:
```
import pandas as pd
```

In [3]:
```
data = pd.read_csv('Salaries.csv')
data
```

Out[3]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	B
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.00	400184.25	
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	106088.18	16452.60	
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.00	56120.71	198306.90	NaN
148652	148653	Not provided	Not provided	NaN	NaN	NaN	
4	5	PATRICK GARDNER	DEPARTMENT Counselor, Log	134401.60	9737.00	182234.59	
148653	148654	Joe Lopez	(FIRE Cabin Ranch	0.00	0.00	0.00	-618.13
...	148654 rows × 13 columns						
148649	148650	Roy I Tillery	Custodian	0.00	0.00	0.00	

1.Display Top 10 Rows of The Dataset

In [5]:
```
data.head(10)
```

Out[5]:

0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.00	400184.25	NaN	5
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN	5
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	106088.18	16452.60	NaN	3
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.00	56120.71	198306.90	NaN	3
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	134401.60	9737.00	182234.59	NaN	3
5	6	DAVID SULLIVAN	ASSISTANT DEPUTY CHIEF II	118602.00	8601.00	189082.74	NaN	3
6	7	ALSON LEE	BATTALION CHIEF, (FIRE DEPARTMENT)	92492.01	89062.90	134426.14	NaN	3
7	8	DAVID KUSHNER	DEPUTY DIRECTOR OF INVESTMENTS	256576.96	0.00	51322.50	NaN	3
8	9	MICHAEL MORRIS	BATTALION CHIEF, (FIRE DEPARTMENT)	176932.64	86362.68	40132.23	NaN	3
9	10	JOANNE HAYES-WHITE	CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	285262.00	0.00	17115.73	NaN	3

2. Check Last 10 Rows of The Dataset

In [7]:
```
data.tail(10)
```

Out[7]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Be
148644	148645	Randy D Winn	Stationary Eng, Sewage Plant	0.0	0.0	0.00	
148645	148646	Carolyn A Wilson	Human Services Technician	0.0	0.0	0.00	
148646	148647	Not provided	Not provided	NaN	NaN	NaN	
148647	148648	Joann Anderson	Communications Dispatcher 2	0.0	0.0	0.00	
148648	148649	Leon Walker	Custodian	0.0	0.0	0.00	
148649	148650	Roy I Tillery	Custodian	0.0	0.0	0.00	
148650	148651	Not provided	Not provided	NaN	NaN	NaN	
148651	148652	Not provided	Not provided	NaN	NaN	NaN	
148652	148653	Not provided	Not provided	NaN	NaN	NaN	
148653	148654	Joe Lopez	Counselor, Log Cabin Ranch	0.0	0.0	-618.13	

3. Find Shape of Our Dataset (Number of Rows And Number of Columns)

In [9]:
```
data.shape
```

Out[9]: (148654, 13)

4. Getting Information About Our Dataset Like Total Number Rows,

Total Number of Columns, Datatypes of Each Column And Memory Requirement

In [11]:
```
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            148045 non-null  float64
 4   OvertimePay        148650 non-null  float64
 5   OtherPay           148650 non-null  float64
 6   Benefits           112491 non-null  float64
 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             0 non-null       float64
dtypes: float64(8), int64(2), object(3)
memory usage: 14.7+ MB
```

5. Check Null Values In The Dataset

In [13]:
```
data.isna().sum()
```

Out[13]:
```
Id                       0
EmployeeName             0
JobTitle                 0
BasePay                609
OvertimePay              4
OtherPay                 4
Benefits             36163
TotalPay                 0
TotalPayBenefits         0
Year                     0
Notes               148654
Agency                   0
Status              148654
dtype: int64
```

6. Drop ID, Notes, Agency and Status Columns

In [15]:
```
data=data.drop(["Id", "Notes", "Agency", "Status"], axis=1)
```

In [16]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148654 entries, 0 to 148653
Data columns (total 9 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   EmployeeName      148654 non-null  object 
 1   JobTitle          148654 non-null  object 
 2   BasePay           148045 non-null  float64
 3   OvertimePay       148650 non-null  float64
 4   OtherPay          148650 non-null  float64
 5   Benefits          112491 non-null  float64
 6   TotalPay          148654 non-null  float64
 7   TotalPayBenefits  148654 non-null  float64
 8   Year              148654 non-null  int64  
dtypes: float64(6), int64(1), object(2)
memory usage: 10.2+ MB
```

7. Get Overall Statistics About The Dataframe

In [18]:
```
data.describe()
```

Out[18]:

	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	Tot
count	148045.000000	148650.000000	148650.000000	112491.000000	148654.000000	1
mean	66325.448840	5066.059886	3648.767297	25007.893151	74768.321972	
std	42764.635495	11454.380559	8056.601866	15402.215858	50517.005274	
min	-166.010000	-0.010000	-7058.590000	-33.890000	-618.130000	
25%	33588.200000	0.000000	0.000000	11535.395000	36168.995000	
50%	65007.450000	0.000000	811.270000	28628.620000	71426.610000	
75%	94691.050000	4658.175000	4236.065000	35566.855000	105839.135000	1
max	319275.010000	245131.880000	400184.250000	96570.660000	567595.430000	5

8. Find Occurrence of The Employee Names (Top 5)

In [20]:
```
top_5_employee_names = data['EmployeeName'].value_counts().head(5)
print(top_5_employee_names)
```

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

9. Find The Number of Unique Job Titles

In [22]:
```
group=data.groupby(["JobTitle"]).count()
```

In [23]:
```
group.shape[0]
```

Out[23]: 2159

10. Total Number of Job Titles Contain Captain

In [25]:
```
total_captain_titles = data[data['JobTitle'].str.contains('Captain', case=False)
total_count = total_captain_titles.shape[0]

print(f'Total Number of Job Titles Containing "Captain": {total_count}')
```

```
Total Number of Job Titles Containing "Captain": 552
```

11.Display All the Employee Names From Fire Department

In [27]:
```
fire_employees = data[data['JobTitle'].str.contains('FIRE DEPARTMENT', case=Fals
for name in fire_employees:
    print(name)
```

```
PATRICK GARDNER
ALSON LEE
MICHAEL MORRIS
JOANNE HAYES-WHITE
ARTHUR KENNEY
DAVID FRANKLIN
MARTY ROSS
VICTOR WYRSCH
RAYMOND GUZMAN
MONICA FIELDS
JOSE VELO
BRENDAN WARD
MICHAEL THOMPSON
THOMAS ABBOTT
THOMAS SIRAGUSA
BRYAN RUBENSTEIN
KEN YEE
KIRK RICHARDSON
KENNETH SMITH
CHARLES CRANE
```

12. Find Minimum, Maximum and Average BasePay

In [29]:
```
data["BasePay"].min()
```

Out[29]: -166.01

In [30]:
```
data["BasePay"].max()
```

Out[30]: 319275.01

In [31]:
```
data["BasePay"].mean()
```

Out[31]: 66325.4488404877

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

In [33]:
```
import numpy as np
data['EmployeeName'] = data['EmployeeName'].replace('Not Provided', np.nan)
```

14. Drop The Rows Having 5 Missing Values

In [35]:
```
data_cleaned = data.dropna(thresh=len(data.columns) - 5)
```

In [36]:
```
print(data_cleaned.head())
```

```
        EmployeeName                                          JobTitle  \
0     NATHANIEL FORD  GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY
1       GARY JIMENEZ                 CAPTAIN III (POLICE DEPARTMENT)
2     ALBERT PARDINI                 CAPTAIN III (POLICE DEPARTMENT)
3  CHRISTOPHER CHONG            WIRE ROPE CABLE MAINTENANCE MECHANIC
4    PATRICK GARDNER    DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)

     BasePay  OvertimePay   OtherPay  Benefits   TotalPay  TotalPayBenefits  \
0  167411.18          0.00  400184.25       NaN  567595.43          567595.43
1  155966.02     245131.88  137811.38       NaN  538909.28          538909.28
2  212739.13     106088.18   16452.60       NaN  335279.91          335279.91
3   77916.00      56120.71  198306.90       NaN  332343.61          332343.61
4  134401.60       9737.00  182234.59       NaN  326373.19          326373.19

   Year
0  2011
1  2011
2  2011
3  2011
4  2011
```

15. Find Job Title of ALBERT PARDINI

In [38]:
```
data.columns
```

Out[38]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

In [39]:
```
data[data['EmployeeName']=="ALBERT  PARDINI"]["JobTitle"]
```

Out[39]:
```
2    CAPTAIN III (POLICE DEPARTMENT)
Name: JobTitle, dtype: object
```

16. Display Name of The Person Having The Highest BasePay

In [43]:
```
data.columns
```

Out[43]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

In [44]:
```
data[data["BasePay"].max()==data["BasePay"]]["EmployeeName"]
```

Out[44]:
```
72925    Gregory P Suhr
Name: EmployeeName, dtype: object
```

17. Find Average BasePay of All Employee Per Year

In [46]:
```
average= data.groupby('Year')['TotalPay'].mean()

print(average)
```

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

18. Find Average BasePay of All Employee Per JobTitle

In [48]:
```
average = data.groupby('JobTitle')['TotalPay'].mean()
print(average)
```

```
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
```

19. Find Average BasePay of Employee Having Job Title ACCOUNTANT

In [50]:
```
accountant_data = data[data['JobTitle'] == 'ACCOUNTANT']

average = accountant_data['TotalPay'].mean()

print("Average BasePay for ACCOUNTANT:", average)
```

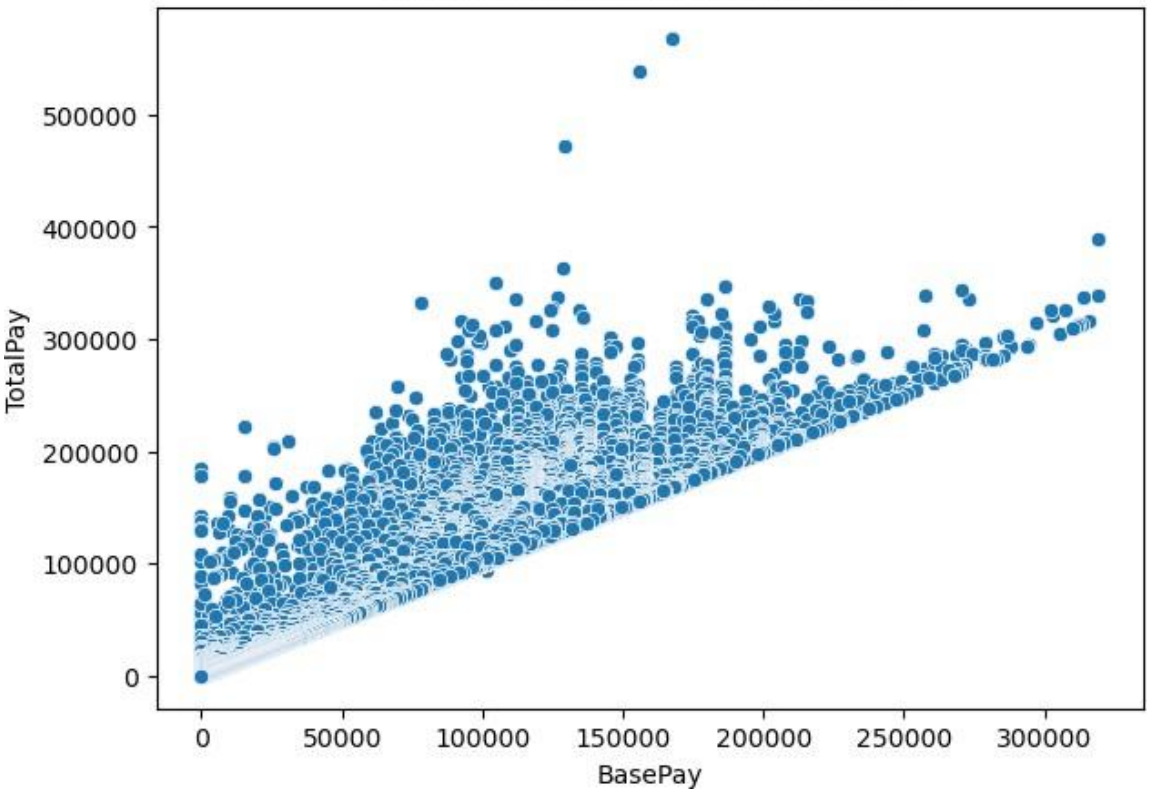
```
Average BasePay for ACCOUNTANT: 47429.268
```

20. Find Top 5 Most Common Jobs

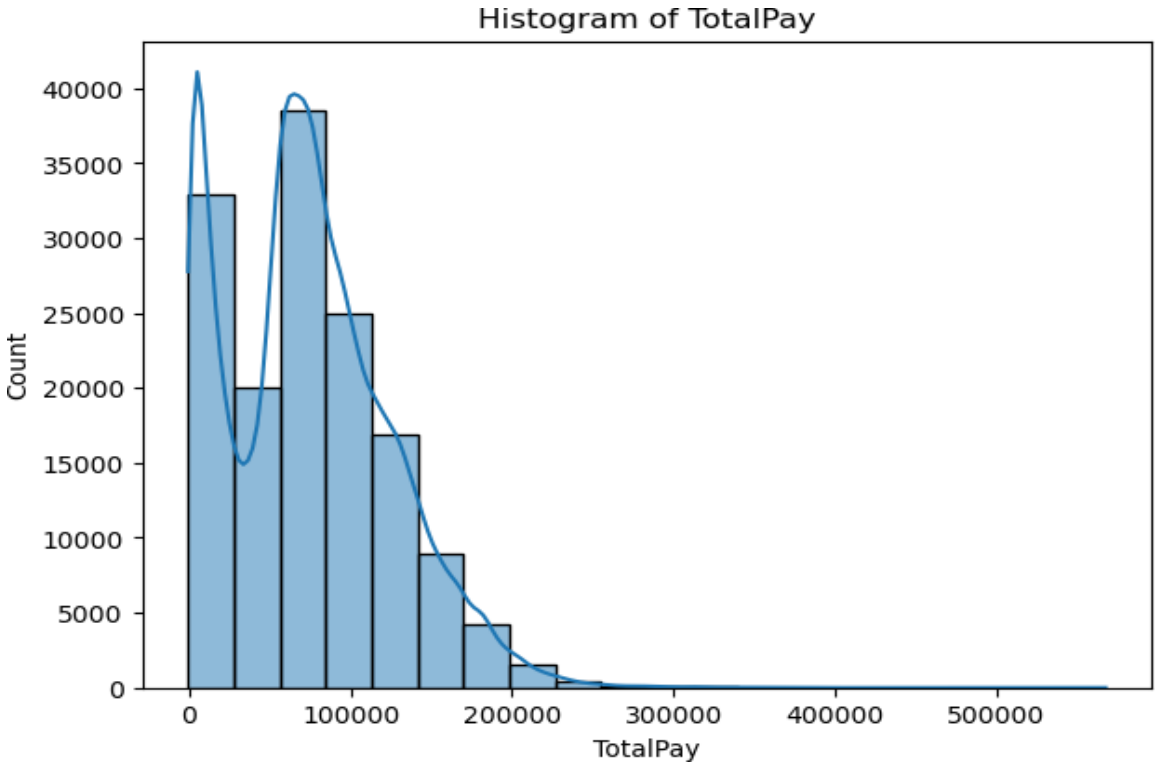
In [52]:
```
top_5_jobs = data['JobTitle'].value_counts().head(5)
print("Top 5 Most Common Job Titles:")
print(top_5_jobs)
```

```
Top 5 Most Common Job Titles:
JobTitle
Transit Operator                7036
Special Nurse                   4389
Registered Nurse      3736 Public Svc
Aide-Public Works                2518
Police Officer 3                 2421
Name: count, dtype: int64
```

In [54]:
```
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(7, 5))
sns.scatterplot(x='BasePay', y='TotalPay', data=data)
plt.title('Scatter Plot of BasePay vs TotalPay')
plt.show()
```



In [56]:
```
plt.figure(figsize=(7, 5))
sns.histplot(data['TotalPay'], bins=20, kde=True)
plt.title('Histogram of TotalPay')
plt.show()
```



In [58]:
```
plt.figure(figsize=(7, 5))
sns.boxplot(x='TotalPay', data=data)
plt.title('Box Plot of TotalPay')
plt.show()
```