SF Salaries

Explore San Francisco city employee salary data

About Dataset

- One way to understand how a city government works is by looking at who it employs and how its employees are compensated.
- This data contains the names, job title, and compensation for San Francisco city employees on an annual basis from 2011 to 2014.

In [3]:

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

In [7]:

```
df=pd.read_csv("Salaries.csv")
df.head()
```

```
C:\Users\sanad\AppData\Local\Temp\ipykernel_11200\1921512307.py:1: DtypeWarning: Columns (3,4,5,6,12) have mixed types. Specify
dtype option on import or set low_memory=False.
  df=pd.read_csv("Salaries.csv")
```

Out[7]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year	Notes	Agency
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.0	400184.25	NaN	567595.43	567595.43	2011	NaN	San Francisco
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN	538909.28	538909.28	2011	NaN	San Francisco
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	106088.18	16452.6	NaN	335279.91	335279.91	2011	NaN	San Francisco
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.0	56120.71	198306.9	NaN	332343.61	332343.61	2011	NaN	San Francisco
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	134401.6	9737.0	182234.59	NaN	326373.19	326373.19	2011	NaN	San Francisco

In [8]:

```
df.head()
```

Out[8]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year	Notes	Agency
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.0	400184.25	NaN	567595.43	567595.43	2011	NaN	San Francisco
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN	538909.28	538909.28	2011	NaN	San Francisco
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	106088.18	16452.6	NaN	335279.91	335279.91	2011	NaN	San Francisco
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.0	56120.71	198306.9	NaN	332343.61	332343.61	2011	NaN	San Francisco
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)	134401.6	9737.0	182234.59	NaN	326373.19	326373.19	2011	NaN	San Francisco

In [9]:

```
df.tail()
```

Out[9]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year	Notes	Agenc
148649	148650	Roy I Tillery	Custodian	0.00	0.00	0.00	0.00	0.00	0.00	2014	NaN	Sa Francisc
148650	148651	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided	0.00	0.00	2014	NaN	Sa Francisc
148651	148652	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided	0.00	0.00	2014	NaN	Sa Francisc
148652	148653	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided	0.00	0.00	2014	NaN	Sa Francisc
148653	148654	Joe Lopez	Counselor, Log Cabin Ranch	0.00	0.00	-618.13	0.00	-618.13	-618.13	2014	NaN	Sa Francisc

In [10]:

```
df.shape
```

Out[10]:

```
(148654, 13)
```

In [11]:

```
print("Number of rows: ", df.shape[0])
print("Number of columns: ", df.shape[1])
```

```
Number of rows:  148654
Number of columns:  13
```

In [12]:

```
df.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
```

In [13]:

```
# Check null values in the dataset?
df.isnull().sum()
```

Out[13]:

```
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
```

In [14]:

```
# Drop ID, Notes, Agency and Status columns
```

```
df.columns
```

Out[14]:

```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agency',
       'Status'],
      dtype='object')
```

In [15]:

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

In [16]:

```
df.isnull().sum()
```

Out[16]:

```
EmployeeName            0
JobTitle                0
BasePay               605
OvertimePay             0
OtherPay                0
Benefits            36159
TotalPay                0
TotalPayBenefits        0
Year                    0
dtype: int64
```

In [17]:

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

Out[17]:

	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year
0	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.0	400184.25	NaN	567595.43	567595.43	2011
1	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN	538909.28	538909.28	2011

In [18]:

```
# Get overall statistics about the dataframe

df.describe()
```

Out[18]:

	TotalPay	TotalPayBenefits	Year
count	148654.000000	148654.000000	148654.000000
mean	74768.321972	93692.554811	2012.522643
std	50517.005274	62793.533483	1.117538
min	-618.130000	-618.130000	2011.000000
25%	36168.995000	44065.650000	2012.000000
50%	71426.610000	92404.090000	2013.000000
75%	105839.135000	132876.450000	2014.000000
max	567595.430000	567595.430000	2014.000000

In [19]:

```
df.describe(include="all") # Overall Statistics
```

Out[19]:

	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year
count	148654	148654	148049.0	148654.0	148654.0	112495.0	148654.000000	148654.000000	148654.000000
unique	110811	2159	109900.0	66555.0	84968.0	99635.0	NaN	NaN	NaN
top	Kevin Lee	Transit Operator	0.0	0.0	0.0	0.0	NaN	NaN	NaN
freq	13	7036	875.0	66103.0	35218.0	1053.0	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	NaN	NaN	74768.321972	93692.554811	2012.522643
std	NaN	NaN	NaN	NaN	NaN	NaN	50517.005274	62793.533483	1.117538
min	NaN	NaN	NaN	NaN	NaN	NaN	-618.130000	-618.130000	2011.000000
25%	NaN	NaN	NaN	NaN	NaN	NaN	36168.995000	44065.650000	2012.000000
50%	NaN	NaN	NaN	NaN	NaN	NaN	71426.610000	92404.090000	2013.000000
75%	NaN	NaN	NaN	NaN	NaN	NaN	105839.135000	132876.450000	2014.000000
max	NaN	NaN	NaN	NaN	NaN	NaN	567595.430000	567595.430000	2014.000000

In [20]:

```
# Find occurrence of the employee names (top 5)
df.columns
```

Out[20]:

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

In [21]:

```
df["EmployeeName"].value_counts().head(5)
```

Out[21]:

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

In [22]:
```
# Find the number of unique job titles
df.columns
```

Out[22]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

In [23]:
```
df["JobTitle"].nunique()
```

Out[23]:
```
2159
```

In [24]:
```
# Total number of job titles contain Captain

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

Out[24]:
```
552
```

In [25]:
```
df[df['JobTitle'].str.contains('CAPTAIN', case=False)].count()
```

Out[25]:
```
EmployeeName        552
JobTitle            552
BasePay             551
OvertimePay         552
OtherPay            552
Benefits            411
TotalPay            552
TotalPayBenefits    552
Year                552
dtype: int64
```

In [26]:
```
# Display all the employee names from fire department

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

Out[26]:
```
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
```

In [28]:
```
# Find minimum, maximum and average BasePay

df['BasePay'].describe()
```

Out[28]:
```
count     148049.0
unique    109900.0
top            0.0
freq         875.0
Name: BasePay, dtype: float64
```

In [31]:
```
# Replace 'Not Provided' in EmployeeName column to NaN

df['EmployeeName']=df['EmployeeName'].replace('Not provided', np.nan)
```

In [32]:
```
df['EmployeeName']
```

Out[32]:

```
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
```

- To make it permanent we stored it inside dataframe "data['EmployeeName']"

In [33]:

```
# 14. Drop the rows having 5 missing values

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

In [34]:

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

Out[34]:

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

- "axis=0" because we need to drop rows.
- here we've to find index of the rows having 5 missing values. So type ".index"
- to make this change permanent type inplace=true

Find job title of ALBERT PARDINI

In [36]:
```
df[df['EmployeeName']=='ALBERT PARDINI']['JobTitle']
```

Out[36]:
```
2    CAPTAIN III (POLICE DEPARTMENT)
Name: JobTitle, dtype: object
```

In [37]:
```
# How much ALBERT PARDINI make (Include Benefits) ?

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

Out[37]:
```
2    335279.91
Name: TotalPayBenefits, dtype: float64
```

In [38]:
```
# Display name of the person having the highest BasePay

df['BasePay'] = pd.to_numeric(df['BasePay'], errors='coerce')

# This code converts BasePay column to numeric values,
# any non-convertible values will be replaced with NaN
```

In [39]:
```
df['BasePay']
```

Out[39]:
```
0         167411.18
1         155966.02
2         212739.13
3          77916.00
4         134401.60
            ...
148649         0.00
148650          NaN
148651          NaN
148652          NaN
148653         0.00
Name: BasePay, Length: 148654, dtype: float64
```

In [40]:
```
df[df['BasePay'].max()==df['BasePay']]['EmployeeName']
```

Out[40]:

```
72925    Gregory P Suhr
Name: EmployeeName, dtype: object
```

In [41]:

```
# Find average BasePay of all employee per year

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

In [ ]:

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

In [43]:

```
# Find average BasePay of all employee per JobTitle

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

In [ ]:

```
df.groupby('JobTitle').mean()['BasePay']
```

In [45]:

```
# Find average BasePay of employee having job title ACCOUNTANT?

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

Out[45]:

```
np.float64(46643.172)
```

In [46]:

```
# Find top 5 most common jobs

df['JobTitle'].value_counts()
```

Out[46]:

```
JobTitle
Transit Operator                       7036
Special Nurse                          4389
Registered Nurse                       3736
Public Svc Aide-Public Works           2518
Police Officer 3                       2421
                                       ... 
Light Rail Vehicle Equip Eng              1
Civil Case Settlmnt Specialist            1
ADMINISTRATOR, SFGH MEDICAL CENTER        1
CHIEF OF POLICE                           1
Special Assistant 8                       1
Name: count, Length: 2159, dtype: int64
```

Reference ::

- https://www.youtube.com/watch?v=qbW8AqEpLtU&list=PL_1pt6K-CLoDMEbYy2PcZuITWEjqMfyoA&index=3

In [ ]: