cisco-city-employee-salary-dataset

July 13, 2025

[1]:
```
import pandas as pd
```

Importing the .csv file

[2]:
```
data = pd.read_csv(r"Salaries.csv")
```

```
/tmp/ipython-input-2-301952016.py:1: DtypeWarning: Columns (3,4,5,6,12) have
mixed types. Specify dtype option on import or set low_memory=False.
  data = pd.read_csv(r"Salaries.csv")
```

[3]:
```
data
```

[3]:

	Id	EmployeeName	JobTitle	BasePay	\
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.0	
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)	134401.6	
…	…	…	…	…	
148649	148650	Roy I Tillery	Custodian	0.00	
148650	148651	Not provided	Not provided	Not Provided	
148651	148652	Not provided	Not provided	Not Provided	
148652	148653	Not provided	Not provided	Not Provided	
148653	148654	Joe Lopez	Counselor, Log Cabin Ranch	0.00	

```
         OvertimePay      OtherPay      Benefits   TotalPay  TotalPayBenefits  \
0                0.0     400184.25           NaN  567595.43         567595.43
1          245131.88     137811.38           NaN  538909.28         538909.28
2          106088.18       16452.6           NaN  335279.91         335279.91
3           56120.71      198306.9           NaN  332343.61         332343.61
4             9737.0     182234.59           NaN  326373.19         326373.19
…                …             …             …          …                 …
148649          0.00          0.00          0.00       0.00              0.00
148650  Not Provided  Not Provided  Not Provided       0.00              0.00
148651  Not Provided  Not Provided  Not Provided       0.00              0.00
148652  Not Provided  Not Provided  Not Provided       0.00              0.00
148653          0.00       -618.13          0.00    -618.13           -618.13

        Year  Notes         Agency Status
0       2011    NaN  San Francisco    NaN
1       2011    NaN  San Francisco    NaN
2       2011    NaN  San Francisco    NaN
3       2011    NaN  San Francisco    NaN
4       2011    NaN  San Francisco    NaN
…        …    …            …    …
148649  2014    NaN  San Francisco     PT
148650  2014    NaN  San Francisco    NaN
148651  2014    NaN  San Francisco    NaN
148652  2014    NaN  San Francisco    NaN
148653  2014    NaN  San Francisco     PT

[148654 rows x 13 columns]
```

Display Top 10 Rows of The Dataset

[4]:
```
data.head(10)
```

[4]:
```
   Id        EmployeeName                                      JobTitle  \
0   1      NATHANIEL FORD  GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY
1   2        GARY JIMENEZ                CAPTAIN III (POLICE DEPARTMENT)
2   3      ALBERT PARDINI                CAPTAIN III (POLICE DEPARTMENT)
3   4   CHRISTOPHER CHONG            WIRE ROPE CABLE MAINTENANCE MECHANIC
4   5     PATRICK GARDNER    DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)
5   6      DAVID SULLIVAN                        ASSISTANT DEPUTY CHIEF II
6   7           ALSON LEE             BATTALION CHIEF, (FIRE DEPARTMENT)
7   8       DAVID KUSHNER                 DEPUTY DIRECTOR OF INVESTMENTS
8   9      MICHAEL MORRIS             BATTALION CHIEF, (FIRE DEPARTMENT)
9  10  JOANNE HAYES-WHITE          CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)

     BasePay OvertimePay   OtherPay Benefits   TotalPay  TotalPayBenefits  \
0  167411.18         0.0  400184.25      NaN  567595.43         567595.43
1  155966.02   245131.88  137811.38      NaN  538909.28         538909.28
```

```
2  212739.13   106088.18    16452.6      NaN  335279.91         335279.91
3    77916.0    56120.71   198306.9      NaN  332343.61         332343.61
4   134401.6      9737.0  182234.59      NaN  326373.19         326373.19
5   118602.0      8601.0  189082.74      NaN  316285.74         316285.74
6   92492.01     89062.9  134426.14      NaN  315981.05         315981.05
7  256576.96         0.0    51322.5      NaN  307899.46         307899.46
8  176932.64    86362.68   40132.23      NaN  303427.55         303427.55
9   285262.0         0.0   17115.73      NaN  302377.73         302377.73

   Year  Notes         Agency Status
0  2011    NaN  San Francisco    NaN
1  2011    NaN  San Francisco    NaN
2  2011    NaN  San Francisco    NaN
3  2011    NaN  San Francisco    NaN
4  2011    NaN  San Francisco    NaN
5  2011    NaN  San Francisco    NaN
6  2011    NaN  San Francisco    NaN
7  2011    NaN  San Francisco    NaN
8  2011    NaN  San Francisco    NaN
9  2011    NaN  San Francisco    NaN
```

Display Last 10 Rows of The Dataset

[5]:
```
data.tail(10)
```

[5]:
```
            Id      EmployeeName                       JobTitle       BasePay  \
148644  148645      Randy D Winn  Stationary Eng, Sewage Plant          0.00
148645  148646  Carolyn A Wilson     Human Services Technician          0.00
148646  148647      Not provided                 Not provided  Not Provided
148647  148648    Joann Anderson   Communications Dispatcher 2          0.00
148648  148649       Leon Walker                     Custodian          0.00
148649  148650     Roy I Tillery                     Custodian          0.00
148650  148651      Not provided                 Not provided  Not Provided
148651  148652      Not provided                 Not provided  Not Provided
148652  148653      Not provided                 Not provided  Not Provided
148653  148654         Joe Lopez    Counselor, Log Cabin Ranch          0.00

         OvertimePay      OtherPay      Benefits  TotalPay  TotalPayBenefits  \
148644          0.00          0.00          0.00      0.00              0.00
148645          0.00          0.00          0.00      0.00              0.00
148646  Not Provided  Not Provided  Not Provided      0.00              0.00
148647          0.00          0.00          0.00      0.00              0.00
148648          0.00          0.00          0.00      0.00              0.00
148649          0.00          0.00          0.00      0.00              0.00
148650  Not Provided  Not Provided  Not Provided      0.00              0.00
148651  Not Provided  Not Provided  Not Provided      0.00              0.00
148652  Not Provided  Not Provided  Not Provided      0.00              0.00
```

```
148653          0.00       -618.13          0.00   -618.13           -618.13
```

```
        Year  Notes         Agency Status
148644  2014    NaN  San Francisco     PT
148645  2014    NaN  San Francisco     PT
148646  2014    NaN  San Francisco    NaN
148647  2014    NaN  San Francisco     PT
148648  2014    NaN  San Francisco     PT
148649  2014    NaN  San Francisco     PT
148650  2014    NaN  San Francisco    NaN
148651  2014    NaN  San Francisco    NaN
148652  2014    NaN  San Francisco    NaN
148653  2014    NaN  San Francisco     PT
```

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

[7]:
```
data.shape
```

[7]:
```
(148654, 13)
```

Getting Information About Our Dataset Like Total Number Rows, Total Number of Columns, Datatypes of Each Column And Memory Requirement

[8]:
```
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
```

Check Null Values In The Dataset

[9]:
```
data.isnull().sum()
```

[9]:
```
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
```

Drop ID, Notes, Agency, and Status Columns

When we are working with big data set at that some columns are not needed and they are no use of us so we have to drop those columns.

[10]:
```
data.columns
```

[10]:
```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agency',
       'Status'],
      dtype='object')
```

[12]:
```
data = data.drop(["Id", "Notes", "Agency", "Status"],axis = 1)
```

[13]:
```
data.columns
```

[13]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

Now you can see i removed those mentioned columns from the dataset

Display Overall Statistics About The Dataframe

[14]:
```
data.describe()
```

[14]:
```
            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
```

Find Occurrence of The Employee Names (Top 5)

[15]:
```
data.columns
```

[15]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

[17]:
```
data["EmployeeName"].value_counts().head(5)
```

[17]:
```
EmployeeName
Kevin Lee       13
William Wong    11
Richard Lee     11
Steven Lee      11
John Chan        9
Name: count, dtype: int64
```

Find The Number of Unique Job Titles

[18]:
```
data.columns
```

[18]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

[19]:
```
data["JobTitle"].nunique()
```

[19]:
```
2159
```

Total Number of Job Titles Contain Captain

[21]:
```
len(data[data["JobTitle"].str.contains("Captain", case = False)])
```

[21]:
```
552
```

Display All the Employee Names From Fire Department

[25]:
```
data[data["JobTitle"].str.contains("Fire", case = False)] ["EmployeeName"]
```

[25]:
```
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
```

Find Minimum, Maximum, and Average BasePay

[35]:
```
data["BasePay"] = pd.to_numeric(data["BasePay"], errors="coerce")
```

[36]:
```
print(data["BasePay"].describe())
```

```
count    148045.000000
mean      66325.448840
std       42764.635495
min        -166.010000
25%       33588.200000
50%       65007.450000
75%       94691.050000
max      319275.010000
Name: BasePay, dtype: float64
```

[37]:
```
min_pay = data["BasePay"].min()
max_pay = data["BasePay"].max()
avg_pay = data["BasePay"].mean()

print("Minimum BasePay:", min_pay)
print("Maximum BasePay:", max_pay)
print("Average BasePay:", avg_pay)
```

```
Minimum BasePay: -166.01
Maximum BasePay: 319275.01
Average BasePay: 66325.4488404877
```

Replace 'Not Provided' in EmployeeName' Column to NaN

[38]:
```
data["EmployeeName"]
```

[38]:
```
0            NATHANIEL FORD
1              GARY JIMENEZ
2            ALBERT PARDINI
3         CHRISTOPHER CHONG
4           PATRICK GARDNER
                 ...
148649        Roy I Tillery
148650         Not provided
148651         Not provided
```

```
148652         Not provided
148653            Joe Lopez
Name: EmployeeName, Length: 148654, dtype: object
```

[39]:
```
import numpy as np
data["EmployeeName"].replace("Not provided", np.nan)
```

[39]:
```
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
```

Why np.nan instead of "NaN" or None? * **np.nan** is the official missing value constant from the **NumPy** library. * **Pandas** is built on top of **NumPy** and understands **np.nan** natively. * It allows you to use pandas functions like **.isna()**, **.dropna()**, **.fillna()**, etc.

Drop The Rows Having 5 Missing Values

[41]:
```
data.drop(data[data.isnull().sum(axis = 1) == 5].index, axis = 0, inplace =␣
↪True)
```

[42]:
```
data.isnull().sum(axis = 1)
```

[42]:
```
0         1
1         1
2         1
3         1
4         1
         ..
148649    0
148650    1
148651    1
148652    1
148653    0
Length: 148654, dtype: int64
```

Find Job Title of ALBERT PARDINI

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

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

[51]:
```
data["EmployeeName"]
```

[51]:
```
0            NATHANIEL FORD
1              GARY JIMENEZ
2             ALBERT PARDINI
3          CHRISTOPHER CHONG
4            PATRICK GARDNER
                  ...
148649         Roy I Tillery
148650          Not provided
148651          Not provided
148652          Not provided
148653             Joe Lopez
Name: EmployeeName, Length: 148654, dtype: object
```

[49]:
```
data[data["EmployeeName"] == "ALBERT PARDINI"]["JobTitle"]
```

[49]:
```
2    CAPTAIN III (POLICE DEPARTMENT)
Name: JobTitle, dtype: object
```

How Much ALBERT PARDINI Make (Include Benefits)?*

[52]:
```
data.columns
```

[52]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

[53]:
```
data[data["EmployeeName"]== "ALBERT PARDINI"] ["TotalPayBenefits"]
```

[53]:
```
2    335279.91
Name: TotalPayBenefits, dtype: float64
```

Display Name of The Person Having The Highest BasePay

[55]:
```
data[data["BasePay"].max() == data["BasePay"]] ["EmployeeName"]
```

[55]:
```
72925    Gregory P Suhr
Name: EmployeeName, dtype: object
```

[56]:
```
data.loc[data["BasePay"].idxmax(), "EmployeeName"] #directly returns the name␣
↪at that index.
```

[56]:
```
'Gregory P Suhr'
```

Find Average BasePay of All Employee Per Year

[57]:
```
data.columns
```

[57]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

[61]:
```
data.groupby("Year")["BasePay"].mean()
```

[61]:
```
Year
2011    63595.956517
2012    65436.406857
2013    69630.030216
2014    66564.421924
Name: BasePay, dtype: float64
```

Find Average BasePay of All Employee Per JobTitle

[62]:
```
data.groupby("JobTitle")["BasePay"].mean()
```

[62]:
```
JobTitle
ACCOUNT CLERK                                   43300.806506
ACCOUNTANT                                      46643.172000
ACCOUNTANT INTERN                               28732.663958
ACPO,JuvP, Juv Prob (SFERS)                     62290.780000
ACUPUNCTURIST                                   66374.400000
                                                       ...
X-RAY LABORATORY AIDE                           47664.773077
X-Ray Laboratory Aide                           46086.387100
YOUTH COMMISSION ADVISOR, BOARD OF SUPERVISORS  52609.910000
Youth Comm Advisor                              39077.957500
ZOO CURATOR                                     43148.000000
Name: BasePay, Length: 2159, dtype: float64
```

Find Average BasePay of Employee Having Job Title ACCOUNTANT

[63]:
```
data[data["JobTitle"] == "ACCOUNTANT"]["BasePay"].mean()
```

[63]:
```
np.float64(46643.172)
```

Find Top 5 Most Common Jobs

[64]:
```
data["JobTitle"].value_counts().head(5)
```

[64]:
```
JobTitle
Transit Operator                7036
Special Nurse                   4389
Registered Nurse                3736
```

```
Public Svc Aide-Public Works    2518
Police Officer 3                2421
Name: count, dtype: int64
```

[ ]: