Data Analysis case study

June 9, 2024

[46]:
```
import pandas as pd
import numpy as np
```

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

[6]:
```
data.head(2)
```

[6]:
```
   Id    EmployeeName                                  JobTitle  \
0   1  NATHANIEL FORD  GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY
1   2    GARY JIMENEZ                 CAPTAIN III (POLICE 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

   Year  Notes         Agency  Status
0  2011    NaN  San Francisco     NaN
1  2011    NaN  San Francisco     NaN
```

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

[7]:
```
            Id      EmployeeName                      JobTitle  BasePay  \
148644  148645      Randy D Winn  Stationary Eng, Sewage Plant      0.0
148645  148646  Carolyn A Wilson     Human Services Technician      0.0
148646  148647      Not provided                Not provided      NaN
148647  148648    Joann Anderson   Communications Dispatcher 2      0.0
148648  148649       Leon Walker                     Custodian      0.0
148649  148650     Roy I Tillery                     Custodian      0.0
148650  148651      Not provided                Not provided      NaN
148651  148652      Not provided                Not provided      NaN
148652  148653      Not provided                Not provided      NaN
148653  148654         Joe Lopez    Counselor, Log Cabin Ranch      0.0

        OvertimePay  OtherPay  Benefits  TotalPay  TotalPayBenefits  Year  \
148644          0.0      0.00       0.0      0.00              0.00  2014
148645          0.0      0.00       0.0      0.00              0.00  2014
148646          NaN       NaN       NaN      0.00              0.00  2014
148647          0.0      0.00       0.0      0.00              0.00  2014
```

```
148648          0.0      0.00       0.0      0.00              0.00  2014
148649          0.0      0.00       0.0      0.00              0.00  2014
148650          NaN       NaN       NaN      0.00              0.00  2014
148651          NaN       NaN       NaN      0.00              0.00  2014
148652          NaN       NaN       NaN      0.00              0.00  2014
148653          0.0   -618.13       0.0   -618.13           -618.13  2014

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

[8]:
```
data.shape
```

[8]:
```
(148654, 13)
```

[9]:
```
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
```

[12]:
```
data.dtypes
```

[12]:
```
Id                   int64
EmployeeName        object
JobTitle            object
BasePay            float64
OvertimePay        float64
OtherPay           float64
Benefits           float64
TotalPay           float64
TotalPayBenefits   float64
Year                 int64
Notes              float64
Agency              object
Status             float64
dtype: object
```

[13]:
```
data.isnull().sum()
```

[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
```

[14]:
```
data.columns
```

[14]:
```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agency',
       'Status'],
      dtype='object')
```

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

[17]:
```
data.head(1)
```

[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

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

[18]:
```
              BasePay    OvertimePay       OtherPay       Benefits  \
count   148045.000000  148650.000000  148650.000000  112491.000000
mean     66325.448840    5066.059886    3648.767297   25007.893151
std      42764.635495   11454.380559    8056.601866   15402.215858
min       -166.010000      -0.010000   -7058.590000     -33.890000
25%      33588.200000       0.000000       0.000000   11535.395000
50%      65007.450000       0.000000     811.270000   28628.620000
75%      94691.050000    4658.175000    4236.065000   35566.855000
max     319275.010000  245131.880000  400184.250000   96570.660000

             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
```

[19]:
```
data.columns
```

[19]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

[22]:
```
## Find the occurance of employee name?
data['EmployeeName'].value_counts().head(5)
```

[22]:
```
Kevin Lee       13
Richard Lee     11
Steven Lee      11
William Wong    11
Stanley Lee      9
Name: EmployeeName, dtype: int64
```

[24]:
```
## find the numbers of unique job title.
data['JobTitle'].nunique()
```

[24]:
```
2159
```

[28]:
```
## find the total number of job titles contain captain.
len(data[data['JobTitle'].str.contains('CAPTAIN',case=False)])
```

[28]:
```
552
```

[35]:
```
data[data['JobTitle'].str.contains('CAPTAIN',case=False)].count()
```

[35]:
```
EmployeeName        552
JobTitle            552
BasePay             551
OvertimePay         552
OtherPay            552
Benefits            411
TotalPay            552
TotalPayBenefits    552
Year                552
dtype: int64
```

[37]:
```
## display all the employees from fire department.
data.columns
```

[37]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

[44]:
```
data[data['JobTitle'].str.contains('FIRE␣
↪DEPARTMENT',case=False)]['EmployeeName']
```

[44]:
```
4              PATRICK GARDNER
6                    ALSON LEE
8                MICHAEL MORRIS
9            JOANNE HAYES-WHITE
10                ARTHUR KENNEY
                    …
32623              JAMES BARDEN
36162       Joanne Hayes-White
72926     Joanne M Hayes-White
102303          Robert E Evans
110535    Joanne M Hayes-White
Name: EmployeeName, Length: 226, dtype: object
```

[45]:
```
## Find the MAX ,MIN AVG,BASEPAY.
data['BasePay'].describe()
```

[45]:
```
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
```

[48]:
```
## repalce "not provided" in employee name column to nan.
data['EmployeeName'].replace('Not provided',np.nan)
```

[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
```

[57]:
```
## drop the rows having 5 missing values.
data.drop(data[data.isnull().sum(axis=1)==5].index,axis=0,inplace=True)
```

[58]:
```
data.isnull().sum(axis=1)
```

[58]:
```
0         1
1         1
2         1
3         1
4         1
         ..
148649    0
148650    4
148651    4
148652    4
148653    0
Length: 148654, dtype: int64
```

[59]:
```
## how much albert pardeni make(include benifite).
data.columns
```

[59]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

[61]:
```
data[data['EmployeeName']=='ALBERT PARDINI']['TotalPayBenefits']
```

[61]:
```
2    335279.91
Name: TotalPayBenefits, dtype: float64
```

[62]:
```
## diplay the name person  having highest basepay.
data.columns
```

[62]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

[72]:
```
data[data['BasePay'].max()==data['BasePay']]['EmployeeName']
```

[72]:
```
72925    Gregory P Suhr
Name: EmployeeName, dtype: object
```

[73]:
```
## find the average basepay all employees per year.
data.columns
```

[73]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

[83]:
```
data.groupby('Year').mean()['BasePay']
```

```
C:\Users\Vikas\AppData\Local\Temp\ipykernel_7100\3282093210.py:1: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a
future version, numeric_only will default to False. Either specify numeric_only
or select only columns which should be valid for the function.
  data.groupby('Year').mean()['BasePay']
```

[83]:
```
Year
2011    63595.956517
2012    65436.406857
2013    69630.030216
2014    66564.421924
Name: BasePay, dtype: float64
```

[85]:
```
## find the average basepay of all employees per job title.
data.columns
```

[85]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

[90]:
```
data.groupby('JobTitle').mean()['BasePay']
```

```
C:\Users\Vikas\AppData\Local\Temp\ipykernel_7100\3409582655.py:1: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a
future version, numeric_only will default to False. Either specify numeric_only
or select only columns which should be valid for the function.
  data.groupby('JobTitle').mean()['BasePay']
```

[90]:
```
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
```

[98]:
```
## find the average basepay employee having jobtitle acountant.
data[data['JobTitle']=='ACCOUNTANT']['BasePay'].mean()
```

[98]:
```
46643.172
```

[99]:
```
## find the top 5 most common jobs.
data.columns
```

[99]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

[103]:
```
data['JobTitle'].value_counts().head()
```

[103]:
```
Transit Operator                7036
Special Nurse                   4389
Registered Nurse                3736
Public Svc Aide-Public Works    2518
Police Officer 3                2421
Name: JobTitle, dtype: int64
```

[ ]:
```

```