SF Salaries Exercise

January 18, 2018

1 SF Salaries Exercise

Welcome to a quick exercise for you to practice your pandas skills! We will be using the SF Salaries Dataset from Kaggle! Just follow along and complete the tasks outlined in bold below. The tasks will get harder and harder as you go along.

** Import pandas as pd.**

```
In [6]: import pandas as pd
```

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

```
In [7]: sal = pd.read_csv('Salaries.csv')
```

** Check the head of the DataFrame. **

```
In [8]: sal.head()

Out[8]:    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)

             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  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
```

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

```
In [9]: sal.info() # 148654 Entries
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148654 entries, 0 to 148653
Data columns (total 13 columns):
Id                  148654 non-null int64
EmployeeName        148654 non-null object
JobTitle            148654 non-null object
BasePay             148045 non-null float64
OvertimePay         148650 non-null float64
OtherPay            148650 non-null float64
Benefits            112491 non-null float64
TotalPay            148654 non-null float64
TotalPayBenefits    148654 non-null float64
Year                148654 non-null int64
Notes               0 non-null float64
Agency              148654 non-null object
Status              0 non-null float64
dtypes: float64(8), int64(2), object(3)
memory usage: 14.7+ MB
```

What is the average BasePay ?

```
In [10]: sal['BasePay'].mean()
```

```
Out[10]: 66325.44884050643
```

** What is the highest amount of OvertimePay in the dataset ? **

```
In [11]: sal['OvertimePay'].max()
```

```
Out[11]: 245131.88
```

** What is the job title of JOSEPH DRISCOLL ? Note: Use all caps, otherwise you may get an answer that doesn't match up (there is also a lowercase Joseph Driscoll). **

```
In [12]: sal[sal['EmployeeName']=='JOSEPH DRISCOLL']['JobTitle']
```

```
Out[12]: 24    CAPTAIN, FIRE SUPPRESSION
         Name: JobTitle, dtype: object
```

** How much does JOSEPH DRISCOLL make (including benefits)? **

```
In [13]: sal[sal['EmployeeName']=='JOSEPH DRISCOLL']['TotalPayBenefits']
```

```
Out[13]: 24    270324.91
         Name: TotalPayBenefits, dtype: float64
```

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

```
In [14]: sal[sal['TotalPayBenefits']== sal['TotalPayBenefits'].max()]
```

```
Out[14]:    Id     EmployeeName                                        JobTitle  \
         0   1  NATHANIEL FORD  GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY

              BasePay  OvertimePay   OtherPay  Benefits   TotalPay  TotalPayBenefits  \
         0  167411.18          0.0  400184.25       NaN  567595.43         567595.43

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

** What is the name of lowest paid person (including benefits)? Do you notice something strange about how much he or she is paid?**

```
In [15]: sal[sal['TotalPayBenefits']== sal['TotalPayBenefits'].min()]
```

```
Out[15]:             Id EmployeeName                    JobTitle  BasePay  OvertimePay  \
         148653  148654    Joe Lopez  Counselor, Log Cabin Ranch      0.0          0.0

                 OtherPay  Benefits  TotalPay  TotalPayBenefits  Year  Notes  \
         148653   -618.13       0.0   -618.13           -618.13  2014    NaN

                        Agency  Status
         148653  San Francisco     NaN
```

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

```
In [16]: sal.groupby('Year').mean()['BasePay']
```

```
Out[16]: Year
         2011    63595.956517
         2012    65436.406857
         2013    69630.030216
         2014    66564.421924
         Name: BasePay, dtype: float64
```

** How many unique job titles are there? **

```
In [17]: sal['JobTitle'].nunique()
```

```
Out[17]: 2159
```

** What are the top 5 most common jobs? **

```
In [18]: sal['JobTitle'].value_counts().head(5)
```

```
Out[18]: Transit Operator                7036
         Special Nurse                   4389
         Registered Nurse                3736
         Public Svc Aide-Public Works    2518
         Police Officer 3                2421
         Name: JobTitle, dtype: int64
```

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

```
In [19]: sum(sal[sal['Year']==2013]['JobTitle'].value_counts() == 1)
```

```
Out[19]: 202
```

** How many people have the word Chief in their job title? (This is pretty tricky) **

```
In [20]: def chief_string(title):
             if 'chief' in title.lower():
                 return True
             else:
                 return False
```

```
In [21]: sum(sal['JobTitle'].apply(lambda x: chief_string(x)))
```

```
Out[21]: 477
```

** Bonus: Is there a correlation between length of the Job Title string and Salary? **

```
In [22]: sal['title_len'] = sal['JobTitle'].apply(len)
```

```
In [23]: sal[['title_len','TotalPayBenefits']].corr()
```

```
Out[23]:                    title_len  TotalPayBenefits
         title_len           1.000000         -0.036878
         TotalPayBenefits   -0.036878          1.000000
```

2 Great Job!