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

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

In [4]:
```
df=pd.read_csv("C:/Users/Mounika/Downloads/Salaries.csv")
```

```
C:\Users\Mounika\anaconda\lib\site-packages\IPython\core\interactiveshell.py:3146: DtypeWarning: Columns (3,4,5,6,12) have mixed types.Specify dtype option on import or set low_memory=False.
  has_raised = await self.run_ast_nodes(code_ast.body, cell_name,
```

**Check the head of the DataFrame.**

In [17]:
```
df.head()
```

Out[17]:

| | Id | EmployeeName | JobTitle | BasePay | OvertimePay | OtherPay | Benefits | TotalPay | TotalPayBenefits | Year | Notes | Agency | Status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | NATHANIEL FORD | GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY | 167411 | 0 | 400184 | NaN | 567595.43 | 567595.43 | 2011 | NaN | San Francisco | NaN |
| **1** | 2 | GARY JIMENEZ | CAPTAIN III (POLICE DEPARTMENT) | 155966 | 245132 | 137811 | NaN | 538909.28 | 538909.28 | 2011 | NaN | San Francisco | NaN |
| **2** | 3 | ALBERT PARDINI | CAPTAIN III (POLICE DEPARTMENT) | 212739 | 106088 | 16452.6 | NaN | 335279.91 | 335279.91 | 2011 | NaN | San Francisco | NaN |
| **3** | 4 | CHRISTOPHER CHONG | WIRE ROPE CABLE MAINTENANCE MECHANIC | 77916 | 56120.7 | 198307 | NaN | 332343.61 | 332343.61 | 2011 | NaN | San Francisco | NaN |
| **4** | 5 | PATRICK GARDNER | DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT) | 134402 | 9737 | 182234 | NaN | 326373.19 | 326373.19 | 2011 | NaN | San Francisco | NaN |

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

In [6]:
```
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
```

**What is the average BasePay ?**

In [30]:
```
df['BasePay'].mean()[0:148646]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-30-9c3fd0c81f24> in <module>
----> 1 df['BasePay'].mean()[0:148646]

~\anaconda\lib\site-packages\pandas\core\generic.py in stat_func(self, axis, skipna, level, numeric_only, **kwargs)
  11466         if level is not None:
  11467             return self._agg_by_level(name, axis=axis, level=level, skipna=skipna)
> 11468         return self._reduce(
  11469             func, name=name, axis=axis, skipna=skipna, numeric_only=numeric_only
  11470         )

~\anaconda\lib\site-packages\pandas\core\series.py in _reduce(self, op, name, axis, skipna, numeric_only, filter_type, **kwds)
   4234                 )
   4235             with np.errstate(all="ignore"):
-> 4236                 return op(delegate, skipna=skipna, **kwds)
   4237
   4238     def _reindex_indexer(self, new_index, indexer, copy):

~\anaconda\lib\site-packages\pandas\core\nanops.py in _f(*args, **kwargs)
     69             try:
     70                 with np.errstate(invalid="ignore"):
---> 71                     return f(*args, **kwargs)
     72             except ValueError as e:
     73                 # we want to transform an object array

~\anaconda\lib\site-packages\pandas\core\nanops.py in f(values, axis, skipna, **kwds)
    127                     result = alt(values, axis=axis, skipna=skipna, **kwds)
    128             else:
--> 129                 result = alt(values, axis=axis, skipna=skipna, **kwds)
    130
    131             return result

~\anaconda\lib\site-packages\pandas\core\nanops.py in nanmean(values, axis, skipna, mask)
    561     dtype_count = dtype
    562     count = _get_counts(values.shape, mask, axis, dtype=dtype_count)
--> 563     the_sum = _ensure_numeric(values.sum(axis, dtype=dtype_sum))
    564
    565     if axis is not None and getattr(the_sum, "ndim", False):

~\anaconda\lib\site-packages\numpy\core\_methods.py in _sum(a, axis, dtype, out, keepdims, initial, where)
     46 def _sum(a, axis=None, dtype=None, out=None, keepdims=False,
     47          initial=_NoValue, where=True):
---> 48     return umr_sum(a, axis, dtype, out, keepdims, initial, where)
     49
     50 def _prod(a, axis=None, dtype=None, out=None, keepdims=False,

TypeError: unsupported operand type(s) for +: 'float' and 'str'
```

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

In [19]:
```
df['OvertimePay'].max()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-19-322a328310f5> in <module>
----> 1 df['OvertimePay'].max()

~\anaconda\lib\site-packages\pandas\core\generic.py in stat_func(self, axis, skipna, level, numeric_only, **kwargs)
  11466         if level is not None:
  11467             return self._agg_by_level(name, axis=axis, level=level, skipna=skipna)
> 11468         return self._reduce(
  11469             func, name=name, axis=axis, skipna=skipna, numeric_only=numeric_only
  11470         )

~\anaconda\lib\site-packages\pandas\core\series.py in _reduce(self, op, name, axis, skipna, numeric_only, filter_type, **kwds)
   4234                 )
   4235             with np.errstate(all="ignore"):
-> 4236                 return op(delegate, skipna=skipna, **kwds)
   4237
   4238     def _reindex_indexer(self, new_index, indexer, copy):

~\anaconda\lib\site-packages\pandas\core\nanops.py in f(values, axis, skipna, **kwds)
    127                     result = alt(values, axis=axis, skipna=skipna, **kwds)
    128             else:
--> 129                 result = alt(values, axis=axis, skipna=skipna, **kwds)
    130
    131             return result

~\anaconda\lib\site-packages\pandas\core\nanops.py in reduction(values, axis, skipna, mask)
    871                 result = np.nan
    872             else:
--> 873                 result = getattr(values, meth)(axis)
    874
    875         result = _wrap_results(result, dtype, fill_value)

~\anaconda\lib\site-packages\numpy\core\_methods.py in _amax(a, axis, out, keepdims, initial, where)
     37 def _amax(a, axis=None, out=None, keepdims=False,
     38           initial=_NoValue, where=True):
---> 39     return umr_maximum(a, axis, None, out, keepdims, initial, where)
     40
     41 def _amin(a, axis=None, out=None, keepdims=False,

TypeError: '>=' not supported between instances of 'float' and 'str'
```

**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 [20]:
```
df[df['EmployeeName']=='JOSEPH DRISCOLL']['JobTitle']
```

Out[20]:
```
24    CAPTAIN, FIRE SUPPRESSION
Name: JobTitle, dtype: object
```

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

In [22]:
```
df[df['EmployeeName']=='JOSEPH DRISCOLL']['TotalPayBenefits']
```

Out[22]:
```
24    270324.91
Name: TotalPayBenefits, dtype: float64
```

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

In [25]:
```
df[df['TotalPayBenefits']==df['TotalPayBenefits'].max()]
```

Out[25]:

| | Id | EmployeeName | JobTitle | BasePay | OvertimePay | OtherPay | Benefits | TotalPay | TotalPayBenefits | Year | Notes | Agency | Status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | NATHANIEL FORD | GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY | 167411 | 0 | 400184 | NaN | 567595.43 | 567595.43 | 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 [26]:
```
df[df['TotalPayBenefits']==df['TotalPayBenefits'].min()]
```

Out[26]:

| | Id | EmployeeName | JobTitle | BasePay | OvertimePay | OtherPay | Benefits | TotalPay | TotalPayBenefits | Year | Notes | Agency | Status |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **148653** | 148654 | Joe Lopez | Counselor, Log Cabin Ranch | 0.00 | 0.00 | -618.13 | 0.00 | -618.13 | -618.13 | 2014 | NaN | San Francisco | PT |

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

In [27]:
```
df.groupby('Year').mean()['BasePay']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
~\anaconda\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
   2894                 try:
-> 2895                     return self._engine.get_loc(casted_key)
   2896                 except KeyError as err:

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\index.pyx in pandas._libs.index.IndexEngine.get_loc()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

pandas\_libs\hashtable_class_helper.pxi in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'BasePay'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
<ipython-input-27-d6ec57df777a> in <module>
----> 1 df.groupby('Year').mean()['BasePay']

~\anaconda\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
   2900             if self.columns.nlevels > 1:
   2901                 return self._getitem_multilevel(key)
-> 2902             indexer = self.columns.get_loc(key)
   2903             if is_integer(indexer):
   2904                 indexer = [indexer]

~\anaconda\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, key, method, tolerance)
   2895                 return self._engine.get_loc(casted_key)
   2896             except KeyError as err:
-> 2897                 raise KeyError(key) from err
   2898
   2899         if tolerance is not None:

KeyError: 'BasePay'
```

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

In [28]:
```
df['JobTitle'].nunique()
```

Out[28]:
```
2159
```

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

In [29]:
```
group=df.groupby('JobTitle').count()
top5=group.sort_values(by='Id',ascending=False)[:5]
top5['Id']
```

Out[29]:
```
JobTitle
Transit Operator                7036
Special Nurse                   4389
Registered Nurse                3736
Public Svc Aide-Public Works    2518
Police Officer 3                2421
Name: Id, 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 [32]:
```
copy_sal=df[df['Year']==2013]
group=copy_sal.groupby('JobTitle').count()
count=group[group['Id']==1]
count.count()['Id']
```

Out[32]:
```
202
```

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

In [35]:
```
def find_chief(job_title):
    if 'chief' in job_title.lower().split():
        return True
    else:
        return False
```

In [36]:
```
df=pd.read_csv('Salaries.csv')
sum(df['JobTitle'].apply(lambda x:find_chief(x)))
```

Out[36]:
```
477
```

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

In [37]:
```
df['title_len']=df['JobTitle'].apply(len)
```

In [38]:
```
df[['title_len','TotalPayBenefits']].corr()
```

Out[38]:

| | title_len | TotalPayBenefits |
|---|---|---|
| **title_len** | 1.000000 | -0.036878 |
| **TotalPayBenefits** | -0.036878 | 1.000000 |

# Great Job!