In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
f2 = {"family":"cambria","color":"g","size":50}
f3  = {"family":"cambria","color":"k","size":40}
f1 = {"family":"cambria","color":"b","size":30}
```

In [2]:
```
SF = pd.read_csv(r"C:\Users\Sinha Rahul\Downloads\Salaries.csv.zip")
```

In [3]:
```
SF
```

Out[3]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.0	400184.25	NaN
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	106088.18	16452.6	NaN
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.0	56120.71	198306.9	NaN
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	134401.6	9737.0	182234.59	NaN
...	...	...	...	...	...	...	...
148649	148650	Roy I Tillery	Custodian	0.00	0.00	0.00	0.00
148650	148651	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided
148651	148652	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided
148652	148653	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided
148653	148654	Joe Lopez	Counselor, Log Cabin Ranch	0.00	0.00	-618.13	0.00

148654 rows × 13 columns

Display Top 10 row in dataset

In [4]:
```
SF.head(10)
```

Out[4]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.0	400184.25	NaN	567595.43
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN	538909.28
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	106088.18	16452.6	NaN	335279.91
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.0	56120.71	198306.9	NaN	332343.61
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	134401.6	9737.0	182234.59	NaN	326373.19
5	6	DAVID SULLIVAN	ASSISTANT DEPUTY CHIEF II	118602.0	8601.0	189082.74	NaN	316285.74
6	7	ALSON LEE	BATTALION CHIEF, (FIRE DEPARTMENT)	92492.01	89062.9	134426.14	NaN	315981.05
7	8	DAVID KUSHNER	DEPUTY DIRECTOR OF INVESTMENTS	256576.96	0.0	51322.5	NaN	307899.46
8	9	MICHAEL MORRIS	BATTALION CHIEF, (FIRE DEPARTMENT)	176932.64	86362.68	40132.23	NaN	303427.55
9	10	JOANNE HAYES-WHITE	CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	285262.0	0.0	17115.73	NaN	302377.73

Display last 10 row in dataset

In [5]:
```
SF.tail(10)
```

Out[5]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	To
148644	148645	Randy D Winn	Stationary Eng, Sewage Plant	0.00	0.00	0.00	0.00	
148645	148646	Carolyn A Wilson	Human Services Technician	0.00	0.00	0.00	0.00	
148646	148647	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided	
148647	148648	Joann Anderson	Communications Dispatcher 2	0.00	0.00	0.00	0.00	
148648	148649	Leon Walker	Custodian	0.00	0.00	0.00	0.00	
148649	148650	Roy I Tillery	Custodian	0.00	0.00	0.00	0.00	
148650	148651	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided	
148651	148652	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided	
148652	148653	Not provided	Not provided	Not Provided	Not Provided	Not Provided	Not Provided	
148653	148654	Joe Lopez	Counselor, Log Cabin Ranch	0.00	0.00	-618.13	0.00	-(

Find the shape of our dataset(number of column & number of row)

In [6]:
```
SF.shape
```

Out[6]:
```
(148654, 13)
```

In [7]:
```
print("number of rows", SF.shape[0])
print("number of columns", SF.shape[1])
```

```
number of rows 148654
number of columns 13
```

Getting information about our dataset likes total number of rows, total number of columns,data type of each column and memory requirement

In [8]:
```
SF.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 [9]:
```
series_list=['BasePay','OvertimePay','OtherPay','Benefits']
for i in series_list:
    SF[i]=pd.to_numeric(SF[i],errors='coerce')
    print(SF[i])
```

```
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
0              0.00
1         245131.88
2         106088.18
3          56120.71
4           9737.00
             ...
148649         0.00
```

getting overall statistics about the dataframe

In [10]:
```
SF.describe()
```

Out[10]:

	Id	BasePay	OvertimePay	OtherPay	Benefits	TotalPa
count	148654.000000	148045.000000	148650.000000	148650.000000	112491.000000	148654.00000
mean	74327.500000	66325.448841	5066.059886	3648.767297	25007.893151	74768.32197
std	42912.857795	42764.635495	11454.380559	8056.601866	15402.215858	50517.00527
min	1.000000	-166.010000	-0.010000	-7058.590000	-33.890000	-618.13000
25%	37164.250000	33588.200000	0.000000	0.000000	11535.395000	36168.99500
50%	74327.500000	65007.450000	0.000000	811.270000	28628.620000	71426.61000
75%	111490.750000	94691.050000	4658.175000	4236.065000	35566.855000	105839.13500
max	148654.000000	319275.010000	245131.880000	400184.250000	96570.660000	567595.43000

In [11]:
```
SF.describe( include="all")  # include categarigal valuebb
```

Out[11]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPa
count	148654.000000	148654	148654	148045.000000	148650.000000	148650.00000
unique	NaN	110811	2159	NaN	NaN	Na
top	NaN	Kevin Lee	Transit Operator	NaN	NaN	Na
freq	NaN	13	7036	NaN	NaN	Na
mean	74327.500000	NaN	NaN	66325.448841	5066.059886	3648.76729
std	42912.857795	NaN	NaN	42764.635495	11454.380559	8056.60186
min	1.000000	NaN	NaN	-166.010000	-0.010000	-7058.59000
25%	37164.250000	NaN	NaN	33588.200000	0.000000	0.00000
50%	74327.500000	NaN	NaN	65007.450000	0.000000	811.27000
75%	111490.750000	NaN	NaN	94691.050000	4658.175000	4236.06500

In [12]:
```
SF.describe(exclude=[object]) # exclude categarigal values
```

Out[12]:

	Id	BasePay	OvertimePay	OtherPay	Benefits	TotalPa
count	148654.000000	148045.000000	148650.000000	148650.000000	112491.000000	148654.00000
mean	74327.500000	66325.448841	5066.059886	3648.767297	25007.893151	74768.32197
std	42912.857795	42764.635495	11454.380559	8056.601866	15402.215858	50517.00527
min	1.000000	-166.010000	-0.010000	-7058.590000	-33.890000	-618.13000
25%	37164.250000	33588.200000	0.000000	0.000000	11535.395000	36168.99500
50%	74327.500000	65007.450000	0.000000	811.270000	28628.620000	71426.61000
75%	111490.750000	94691.050000	4658.175000	4236.065000	35566.855000	105839.13500
max	148654.000000	319275.010000	245131.880000	400184.250000	96570.660000	567595.43000

In [13]:
```
SF.describe(include=[np.number])
```

Out[13]:

	Id	BasePay	OvertimePay	OtherPay	Benefits	TotalPa
count	148654.000000	148045.000000	148650.000000	148650.000000	112491.000000	148654.00000
mean	74327.500000	66325.448841	5066.059886	3648.767297	25007.893151	74768.32197
std	42912.857795	42764.635495	11454.380559	8056.601866	15402.215858	50517.00527
min	1.000000	-166.010000	-0.010000	-7058.590000	-33.890000	-618.13000
25%	37164.250000	33588.200000	0.000000	0.000000	11535.395000	36168.99500
50%	74327.500000	65007.450000	0.000000	811.270000	28628.620000	71426.61000
75%	111490.750000	94691.050000	4658.175000	4236.065000	35566.855000	105839.13500
max	148654.000000	319275.010000	245131.880000	400184.250000	96570.660000	567595.43000

In [14]:
```
SF.describe(exclude=[np.number])
```

Out[14]:

	EmployeeName	JobTitle	Agency	Status
count	148654	148654	148654	38119
unique	110811	2159	1	2
top	Kevin Lee	Transit Operator	San Francisco	FT
freq	13	7036	148654	22334

Drop ID, Notes, Agency,status Columns

In [15]:
```
SF.columns
```

Out[15]:
```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPa
y',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agency',
       'Status'],
      dtype='object')
```

In [16]:
```
SF = SF.drop(['Id','Notes','Agency','Status'],axis=1)
SF.head()
```

Out[16]:

	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	Tot
0	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.00	400184.25	NaN	567595.43	
1	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN	538909.28	
2	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	106088.18	16452.60	NaN	335279.91	
3	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.00	56120.71	198306.90	NaN	332343.61	
4	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	134401.60	9737.00	182234.59	NaN	326373.19	

In [17]:
```
SF.columns
```

Out[17]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

In [18]:
```
SF.describe()
```

Out[18]:

	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBen
count	148045.000000	148650.000000	148650.000000	112491.000000	148654.000000	148654.00
mean	66325.448841	5066.059886	3648.767297	25007.893151	74768.321972	93692.55
std	42764.635495	11454.380559	8056.601866	15402.215858	50517.005274	62793.53
min	-166.010000	-0.010000	-7058.590000	-33.890000	-618.130000	-618.13
25%	33588.200000	0.000000	0.000000	11535.395000	36168.995000	44065.65
50%	65007.450000	0.000000	811.270000	28628.620000	71426.610000	92404.09
75%	94691.050000	4658.175000	4236.065000	35566.855000	105839.135000	132876.45
max	319275.010000	245131.880000	400184.250000	96570.660000	567595.430000	567595.43

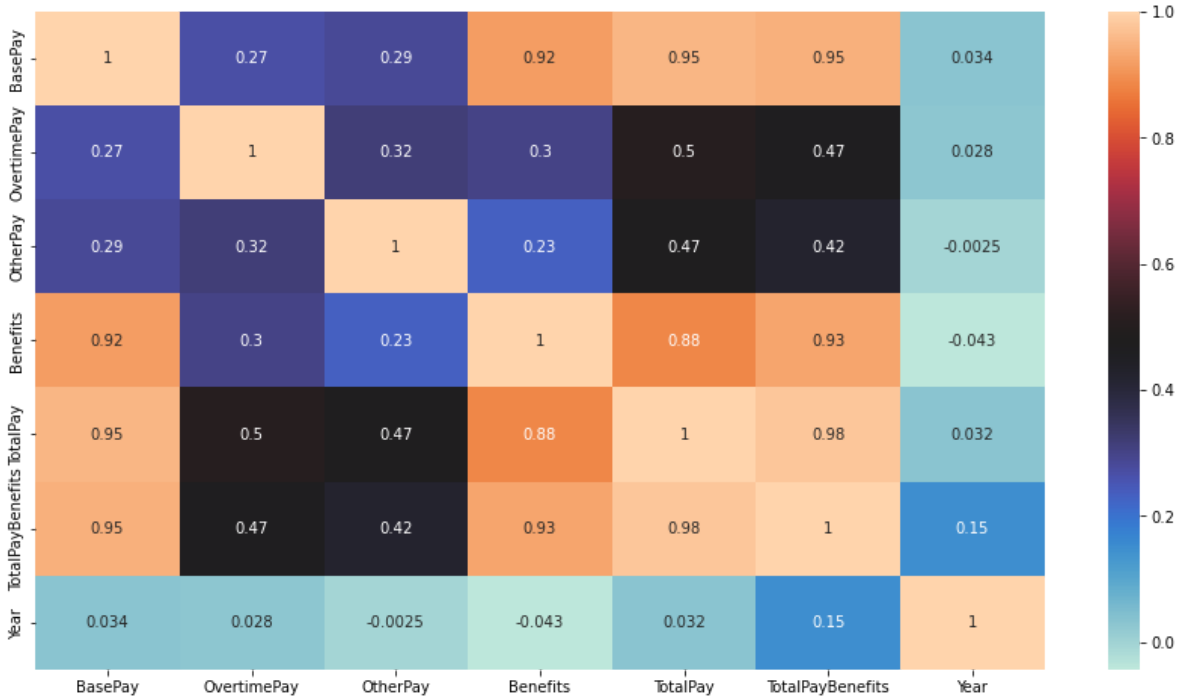
In [19]:
```
SF.corr()
```

Out[19]:

	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Y
BasePay	1.000000	0.266740	0.285655	0.918028	0.954494	0.946595	0.033
OvertimePay	0.266740	1.000000	0.316592	0.301207	0.504859	0.467981	0.027
OtherPay	0.285655	0.316592	1.000000	0.233178	0.470496	0.422341	-0.002
Benefits	0.918028	0.301207	0.233178	1.000000	0.884097	0.930140	-0.043
TotalPay	0.954494	0.504859	0.470496	0.884097	1.000000	0.977313	0.032
TotalPayBenefits	0.946595	0.467981	0.422341	0.930140	0.977313	1.000000	0.151
Year	0.033751	0.027887	-0.002499	-0.043136	0.032090	0.151947	1.000

In [20]:
```
plt.figure(figsize=(15,8))
sns.heatmap(SF.corr(),cmap = "icefire", annot = True)
```

Out[20]: <AxesSubplot:>



check the null value in our dataset

In [21]:
```
SF.isnull().sum()
```

Out[21]:
```
EmployeeName           0
JobTitle               0
BasePay              609
OvertimePay            4
OtherPay               4
Benefits           36163
TotalPay               0
TotalPayBenefits       0
Year                   0
dtype: int64
```

In [22]:
```
SF.isnull().sum().sort_values(ascending=False)
```

Out[22]:
```
Benefits            36163
BasePay               609
OvertimePay             4
OtherPay                4
EmployeeName            0
JobTitle                0
TotalPay                0
TotalPayBenefits        0
Year                    0
dtype: int64
```

In [23]:
```
SF.shape[0]
```

Out[23]:
```
148654
```

In [24]:
```
SF.shape[1]
```

Out[24]:
```
9
```

In [25]:
```
SF.isnull().sum().sum()
```

Out[25]:
```
36780
```

In [26]:
```
SF = SF.dropna()
```

In [27]:
```
SF.isnull().sum().sum()
```

Out[27]:
```
0
```

In [28]:
```
SF.isnull().sum()
```

Out[28]:
```
EmployeeName        0
JobTitle            0
BasePay             0
OvertimePay         0
OtherPay            0
Benefits            0
TotalPay            0
TotalPayBenefits    0
Year                0
dtype: int64
```

In [29]:
```
SF.isnull().any()
```

Out[29]:
```
EmployeeName        False
JobTitle            False
BasePay             False
OvertimePay         False
OtherPay            False
Benefits            False
TotalPay            False
TotalPayBenefits    False
Year                False
dtype: bool
```

In [30]:
```
SF.isnull().all()
```

Out[30]:
```
EmployeeName        False
JobTitle            False
BasePay             False
OvertimePay         False
OtherPay            False
Benefits            False
TotalPay            False
TotalPayBenefits    False
Year                False
dtype: bool
```

Find Occurrence of the employee Names (Top 5)

In [31]:
```
SF.columns
```

Out[31]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

In [32]:
```
SF['EmployeeName'].value_counts().head(5)
```

Out[32]:
```
Kevin Lee       13
Steven Lee      11
William Wong    11
Richard Lee     11
John Chan        9
Name: EmployeeName, dtype: int64
```

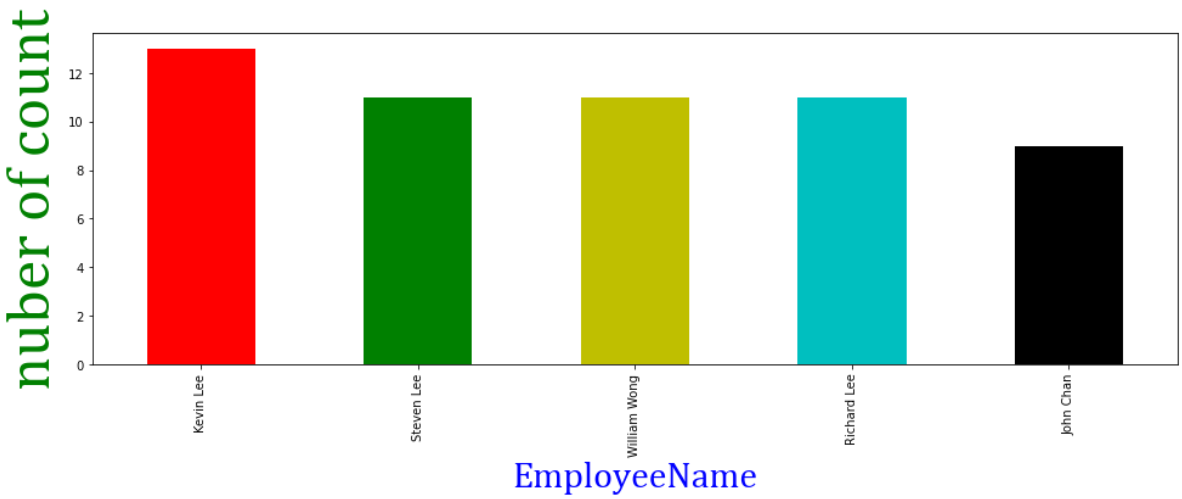
In [33]:
```
SF['EmployeeName'].value_counts().head(5).reset_index().rename(columns = {"ind
                                                                          "Empl
```

Out[33]:

	EmployeeName	count
0	Kevin Lee	13
1	Steven Lee	11
2	William Wong	11
3	Richard Lee	11
4	John Chan	9

data visualization

In [34]:
```
cm = ["r","g","y","c","k"]
SF['EmployeeName'].value_counts().head(5).plot(kind = "bar",figsize=(16,5), co
plt.xlabel("EmployeeName", fontdict=f1)
plt.ylabel("nuber of count",fontdict=f2)
plt.show()
```



In [35]:
```
SF.sort_values(by = "Benefits",ascending=False).head(5)
```

Out[35]:

	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	Tot
110533	William J Coaker Jr.	Chief Investment Officer	257340.00	0.00	82313.70	96570.66	339653.70	
110534	Gregory P Suhr	Chief of Police	307450.04	0.00	19266.72	91302.46	326716.76	
110535	Joanne M Hayes-White	Chief, Fire Department	302068.00	0.00	24165.44	91201.66	326233.44	
110537	John L Martin	Dept Head V	311298.55	0.00	0.00	89772.32	311298.55	
110532	Amy P Hart	Asst Med Examiner	318835.49	10712.95	60563.54	89540.23	390111.98	

In [36]:
```
SF.sort_values(by = "Benefits",ascending=False).head(5)["EmployeeName"]
```

Out[36]:
```
110533    William J Coaker Jr.
110534          Gregory P Suhr
110535    Joanne M Hayes-White
110537           John L Martin
110532              Amy P Hart
Name: EmployeeName, dtype: object
```

Find the number of unique job title

In [37]:
```
SF.columns
```

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

In [38]:
```
SF["JobTitle"].nunique()
```

Out[38]:
```
1109
```

In [39]:
```
SF["JobTitle"].unique()
```

Out[39]:
```
array(['Lieutenant, Fire Suppression', 'Chief of Police',
       'Electronic Maintenance Tech', ...,
       'Forensic Toxicologist Supervis', 'Conversion', 'Cashier 3'],
      dtype=object)
```

In [40]:
```
SF["JobTitle"]
```

Out[40]:
```
36159     Lieutenant, Fire Suppression
36160                  Chief of Police
36161      Electronic Maintenance Tech
36162           Chief, Fire Department
36163        EMT/Paramedic/Firefighter
                        ...
148645       Human Services Technician
148647     Communications Dispatcher 2
148648                       Custodian
148649                       Custodian
148653      Counselor, Log Cabin Ranch
Name: JobTitle, Length: 111886, dtype: object
```

In [41]:
```
type(SF["JobTitle"])
```

Out[41]:
```
pandas.core.series.Series
```

In [42]:
```
list1 =[]
for value in SF['JobTitle']:
    list1.append(value.split(','))
```

In [43]:
```
list1
```

```
 ['EMT/Paramedic/Firefighter'],
 ['Captain', ' Emergency Med Svcs'],
 ['Firefighter'],
 ['Anesthetist'],
 ['Nursing Supervisor'],
 ['Dep Dir V'],
 ['Manager VIII'],
 ['Supervising Physician Spec'],
 ['Battlion Chief', ' Fire Suppressi'],
 ['Wire Rope Cable Maint Sprv'],
 ['Transit Supervisor'],
 ['Battlion Chief', ' Fire Suppressi'],
 ['Firefighter'],
 ['Firefighter'],
 ['Captain 3'],
 ['Dep Dir V'],
 ['Firefighter'],
 ['Nurse Manager'],
 ['Firefighter'],
 ['Firefighter'],
```

In [44]:
```
len(list1)
```

Out[44]:
```
111886
```

In [45]:
```
one_d = []
for item in list1:
    for item1 in item:
        one_d.append(item1)
```

In [46]:
```
one_d
```

Out[46]:
```
['Lieutenant',
 ' Fire Suppression',
 'Chief of Police',
 'Electronic Maintenance Tech',
 'Chief',
 ' Fire Department',
 'EMT/Paramedic/Firefighter',
 'Dept Head V',
 'Gen Mgr',
 ' Public Trnsp Dept',
 'Dept Head V',
 'Captain 3',
 'Asst Chf of Dept (Fire Dept)',
 'Battlion Chief',
 ' Fire Suppressi',
 'Battlion Chief',
 ' Fire Suppressi',
 'Battlion Chief',
 ' Fire Suppressi',
```

In [47]:
```
len(one_d)
```

Out[47]:
```
117365
```

In [48]:
```
uni_list = []
for item in one_d:
    if item not in uni_list:
        uni_list.append(item)
```

In [49]:
```
uni_list
```

Out[49]:
```
['Lieutenant',
 ' Fire Suppression',
 'Chief of Police',
 'Electronic Maintenance Tech',
 'Chief',
 ' Fire Department',
 'EMT/Paramedic/Firefighter',
 'Dept Head V',
 'Gen Mgr',
 ' Public Trnsp Dept',
 'Captain 3',
 'Asst Chf of Dept (Fire Dept)',
 'Battlion Chief',
 ' Fire Suppressi',
 'Assistant Deputy Chief 2',
 'Transit Manager 2',
 'Asst Med Examiner',
 'Dep Chf of Dept (Fire Dept)',
 'Executive Contract Employee',
```

In [50]:
```
len(uni_list)
```

Out[50]:
```
1168
```

Total Number of job title contains captain

In [51]:
```
SF.columns
```

Out[51]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

In [52]:
```
SF["JobTitle"].unique()
```

Out[52]:
```
array(['Lieutenant, Fire Suppression', 'Chief of Police',
       'Electronic Maintenance Tech', ...,
       'Forensic Toxicologist Supervis', 'Conversion', 'Cashier 3'],
      dtype=object)
```

In [53]:
```
SF[SF["JobTitle"].str.contains("captain", case = False)]
```

Out[53]:

	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay
36167	John Goldberg	Captain 3	104404.00	0.00	245999.41	24287.23	350403.41
36180	Michael Rolovich	Captain, Fire Suppression	145659.03	125868.06	30474.97	45129.03	302002.06
36184	Darryl Hunter	Captain, Fire Suppression	145659.03	115673.73	32610.00	49571.51	293942.76
36186	Michael Delane	Captain, Fire Suppression	147069.71	113372.94	33012.58	45978.15	293455.23
36189	Philip Stevens	Captain, Fire Suppression	124573.50	53895.92	130200.40	28907.03	308669.82
...	...	...	...	...	...	...	...

In [54]:
```
len(SF[SF["JobTitle"].str.contains("cAPTAIN", case = False)])
```

Out[54]:
```
410
```

In [55]:
```
len(SF[SF["JobTitle"].str.contains("CAPTAIN")])
```

Out[55]:
```
0
```

In [56]:
```
len(SF[SF["JobTitle"].str.contains("CAPTAIN",case= False)])
```

Out[56]:
```
410
```

In [57]:
```
SF[SF["JobTitle"].str.contains("CAPTAIN")].count()
```

Out[57]:
```
EmployeeName        0
JobTitle            0
BasePay             0
OvertimePay         0
OtherPay            0
Benefits            0
TotalPay            0
TotalPayBenefits    0
Year                0
dtype: int64
```

In [58]:
```
SF[SF["JobTitle"].str.contains("CAPTAIN", case = False)].count()
```

Out[58]:
```
EmployeeName        410
JobTitle            410
BasePay             410
OvertimePay         410
OtherPay            410
Benefits            410
TotalPay            410
TotalPayBenefits    410
Year                410
dtype: int64
```

Display All THe Employee Names From Fire Department

In [59]:
```
SF.columns
```

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

In [60]:
```
SF[SF["JobTitle"].str.contains("Fire",case = False)]
```

Out[60]:

	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefit
36159	Gary Altenberg	Lieutenant, Fire Suppression	128808.87	220909.48	13126.31	44430.1
36162	Joanne Hayes-White	Chief, Fire Department	296943.01	0.00	17816.59	72047.8
36163	Frederick Binkley	EMT/Paramedic/Firefighter	126863.19	192424.49	17917.18	44438.2
36168	David Franklin	Asst Chf of Dept (Fire Dept)	204032.52	85503.16	26193.09	58486.1
36169	Brendan Ward	Battlion Chief, Fire Suppressi	174822.47	118215.58	28845.78	49648.0
...	...	...	...	...	...	.
145956	Kenneth C Farris	Firefighter	0.00	0.00	0.00	4645.5

In [61]:
```
SF["JobTitle"].str.contains("Fire",case = False)
```

Out[61]:
```
36159      True
36160     False
36161     False
36162      True
36163      True
          ...
148645    False
148647    False
148648    False
148649    False
148653    False
Name: JobTitle, Length: 111886, dtype: bool
```

In [62]:
```
len(SF[SF["JobTitle"].str.contains("Fire",case = False)])
```

Out[62]:
```
4399
```

In [63]:
```
len(SF[SF["JobTitle"].str.contains("Fire")])
```

Out[63]:
```
4399
```

In [64]:
```
SF[SF["JobTitle"].str.contains("Fire",case = False)]["EmployeeName"]
```

Out[64]:
```
36159         Gary Altenberg
36162     Joanne Hayes-White
36163      Frederick Binkley
36168         David Franklin
36169           Brendan Ward
                   ...
145956      Kenneth C Farris
147556         Edward A Dunn
148021        Kari A Johnson
148209          Sheryl K Lee
148554       Lawrence F Gatt
Name: EmployeeName, Length: 4399, dtype: object
```

In [65]:
```
SF[SF["JobTitle"].str.contains("Fire",case = False)][["EmployeeName","JobTitle
```

Out[65]:

	EmployeeName	JobTitle
36159	Gary Altenberg	Lieutenant, Fire Suppression
36162	Joanne Hayes-White	Chief, Fire Department
36163	Frederick Binkley	EMT/Paramedic/Firefighter
36168	David Franklin	Asst Chf of Dept (Fire Dept)
36169	Brendan Ward	Battlion Chief, Fire Suppressi
...	...	...
145956	Kenneth C Farris	Firefighter
147556	Edward A Dunn	Firefighter
148021	Kari A Johnson	Firefighter
148209	Sheryl K Lee	Firefighter
148554	Lawrence F Gatt	Fire Alarm Dispatcher

Find the minimum And Maximum avarage Basepay

In [66]:
```
SF.columns
```

Out[66]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

In [67]:
```
SF['BasePay'] = pd.to_numeric(SF['BasePay'], errors='coerce')
```

In [68]:
```
SF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 111886 entries, 36159 to 148653
Data columns (total 9 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   EmployeeName      111886 non-null  object 
 1   JobTitle          111886 non-null  object 
 2   BasePay           111886 non-null  float64
 3   OvertimePay       111886 non-null  float64
 4   OtherPay          111886 non-null  float64
 5   Benefits          111886 non-null  float64
 6   TotalPay          111886 non-null  float64
 7   TotalPayBenefits  111886 non-null  float64
 8   Year              111886 non-null  int64  
dtypes: float64(6), int64(1), object(2)
memory usage: 8.5+ MB
```

In [69]:
```
SF["BasePay"].describe()
```

Out[69]:
```
count    111886.000000
mean      67207.558425
std       43417.689463
min        -166.010000
25%       33644.427500
50%       65547.035000
75%       95229.030000
max      319275.010000
Name: BasePay, dtype: float64
```

In [70]:
```
SF["BasePay"].mean()
```

Out[70]:
```
67207.55842466283
```

In [71]:
```
SF["BasePay"].min()
```

Out[71]:
```
-166.01
```

In [72]:
```
SF["BasePay"].max()
```

Out[72]:
```
319275.01
```

REplace non provied in Employee name column to nan

In [73]:
```
SF["EmployeeName"].replace("Not provided", np.nan)
```

Out[73]:
```
36159         Gary Altenberg
36160           Gregory Suhr
36161             Khoa Trinh
36162     Joanne Hayes-White
36163      Frederick Binkley
                  ...
148645      Carolyn A Wilson
148647        Joann Anderson
148648           Leon Walker
148649         Roy I Tillery
148653             Joe Lopez
Name: EmployeeName, Length: 111886, dtype: object
```

In [74]:
```
SF["EmployeeName"] = SF["EmployeeName"].replace("Not provided", np.nan)
```

In [75]:
```
SF
```

Out[75]:

	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits
36159	Gary Altenberg	Lieutenant, Fire Suppression	128808.87	220909.48	13126.31	44430.12
36160	Gregory Suhr	Chief of Police	302578.00	0.00	18974.11	69810.19
36161	Khoa Trinh	Electronic Maintenance Tech	111921.00	146415.32	78057.41	53102.29
36162	Joanne Hayes-White	Chief, Fire Department	296943.01	0.00	17816.59	72047.88
36163	Frederick Binkley	EMT/Paramedic/Firefighter	126863.19	192424.49	17917.18	44438.25
...	...	...	...	...	...	...
148645	Carolyn A Wilson	Human Services Technician	0.00	0.00	0.00	0.00
148647	Joann Anderson	Communications Dispatcher 2	0.00	0.00	0.00	0.00
148648	Leon Walker	Custodian	0.00	0.00	0.00	0.00
148649	Roy I Tillery	Custodian	0.00	0.00	0.00	0.00
148653	Joe Lopez	Counselor, Log Cabin Ranch	0.00	0.00	-618.13	0.00

111886 rows × 9 columns

Find JOb title of albart pardini

In [76]:
```
SF[SF["EmployeeName"] == "ALBERT PARDINI"]
```

Out[76]:

EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefit

In [77]:
```
SF[SF["EmployeeName"] == "ALBERT PARDINI"]["JobTitle"]
```

Out[77]:
```
Series([], Name: JobTitle, dtype: object)
```

In [78]:
```
SF[SF["EmployeeName"] == "ALBERT PARDINI"][["EmployeeName","JobTitle"]]
```

Out[78]:

EmployeeName	JobTitle

How Many ALbart pardini make (include Benifit)?

In [79]:
```
SF[SF["EmployeeName"] == "ALBERT PARDINI"][["EmployeeName","TotalPayBenefits"]
```

Out[79]:

EmployeeName	TotalPayBenefits

Display name of person having the highest basepay

In [80]:
```
SF.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 111886 entries, 36159 to 148653
Data columns (total 9 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   EmployeeName      111886 non-null  object 
 1   JobTitle          111886 non-null  object 
 2   BasePay           111886 non-null  float64
 3   OvertimePay       111886 non-null  float64
 4   OtherPay          111886 non-null  float64
 5   Benefits          111886 non-null  float64
 6   TotalPay          111886 non-null  float64
 7   TotalPayBenefits  111886 non-null  float64
 8   Year              111886 non-null  int64  
dtypes: float64(6), int64(1), object(2)
memory usage: 8.5+ MB
```

In [81]:
```
SF[SF["BasePay"].max() == SF["BasePay"]]
```

Out[81]:

	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPa
72925	Gregory P Suhr	Chief of Police	319275.01	0.0	20007.06	86533.21	339282.07	

In [82]:
```
SF[SF["BasePay"].max() == SF["BasePay"]]["EmployeeName"]
```

Out[82]:
```
72925    Gregory P Suhr
Name: EmployeeName, dtype: object
```

In [83]:
```
SF[SF["BasePay"] == SF["BasePay"].max()][["BasePay","EmployeeName"]]
```

Out[83]:

	BasePay	EmployeeName
72925	319275.01	Gregory P Suhr

Find the avarge basepay of all employee per year

In [84]:
```
SF.groupby("Year")
```

Out[84]:
```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001A52AF8CDC0>
```

In [85]:
```
SF.groupby("Year").mean()
```

Out[85]:

Year	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits
2012	65436.406857	5023.417824	3653.437583	26439.966967	74113.262265	100553.229232
2013	69630.030216	5367.913512	3810.341313	24131.696305	78808.285041	102939.981346
2014	66564.421924	5401.993737	3505.421251	24789.601756	75471.836912	100261.438668

In [86]:
```
SF.groupby("Year").mean()["BasePay"].reset_index()
```

Out[86]:

	Year	BasePay
0	2012	65436.406857
1	2013	69630.030216
2	2014	66564.421924

In [87]:
```
SF.groupby("Year").mean()["BasePay"].reset_index().keys()
```

Out[87]:
```
Index(['Year', 'BasePay'], dtype='object')
```

In [88]:
```
SF.groupby("Year")["BasePay"].mean().reset_index()
```

Out[88]:

	Year	BasePay
0	2012	65436.406857
1	2013	69630.030216
2	2014	66564.421924

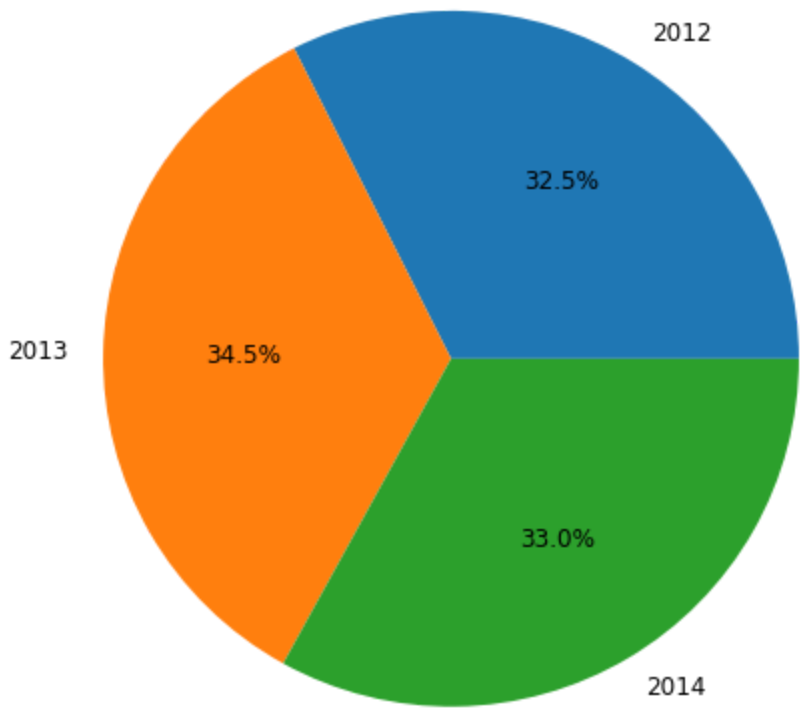
In [89]:
```
SF.groupby("Year").mean()["BasePay"]
```

Out[89]:
```
Year
2012    65436.406857
2013    69630.030216
2014    66564.421924
Name: BasePay, dtype: float64
```

In [90]:
```
SF.groupby("Year").mean()["BasePay"].keys()
```

Out[90]:
```
Int64Index([2012, 2013, 2014], dtype='int64', name='Year')
```

In [113]:
```
plt.pie(SF.groupby("Year").mean()["BasePay"],labels= SF.groupby("Year").mean()
)
plt.show()
```



Find the avarge basepay of all employee per jobtitle

In [92]:
```
SF.groupby("JobTitle").mean()
```

Out[92]:

JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPay
ACPO,JuvP, Juv Prob (SFERS)	62290.780000	0.000000	0.000000	17975.590000	62290.780000	8026
ASR Senior Office Specialist	60551.580167	410.988500	2556.794500	26604.874167	63519.363167	9012
ASR-Office Assistant	41253.471951	18.502683	239.527317	19250.591707	41511.501951	6076
Account Clerk	42372.579396	193.827547	579.346830	20100.588226	43145.753774	6324
Accountant I	61777.832500	0.000000	258.268750	26086.087500	62036.101250	8812
...	...	...	...	...	...	
Wire Rope Cable Maint Sprv	92751.746667	82446.923333	27835.050000	39084.603333	203033.720000	24211
Worker's Comp Supervisor 1	68867.296429	0.000000	1522.000714	25736.234286	70389.297143	9612
Worker's Compensation Adjuster	72363.278784	0.000000	885.991081	28072.155946	73249.269865	10132
X-Ray Laboratory Aide	46086.387100	3483.767100	1253.788500	18697.180500	50823.942700	6952
Youth Comm Advisor	39077.957500	0.000000	2336.350000	18704.242500	41414.307500	6011

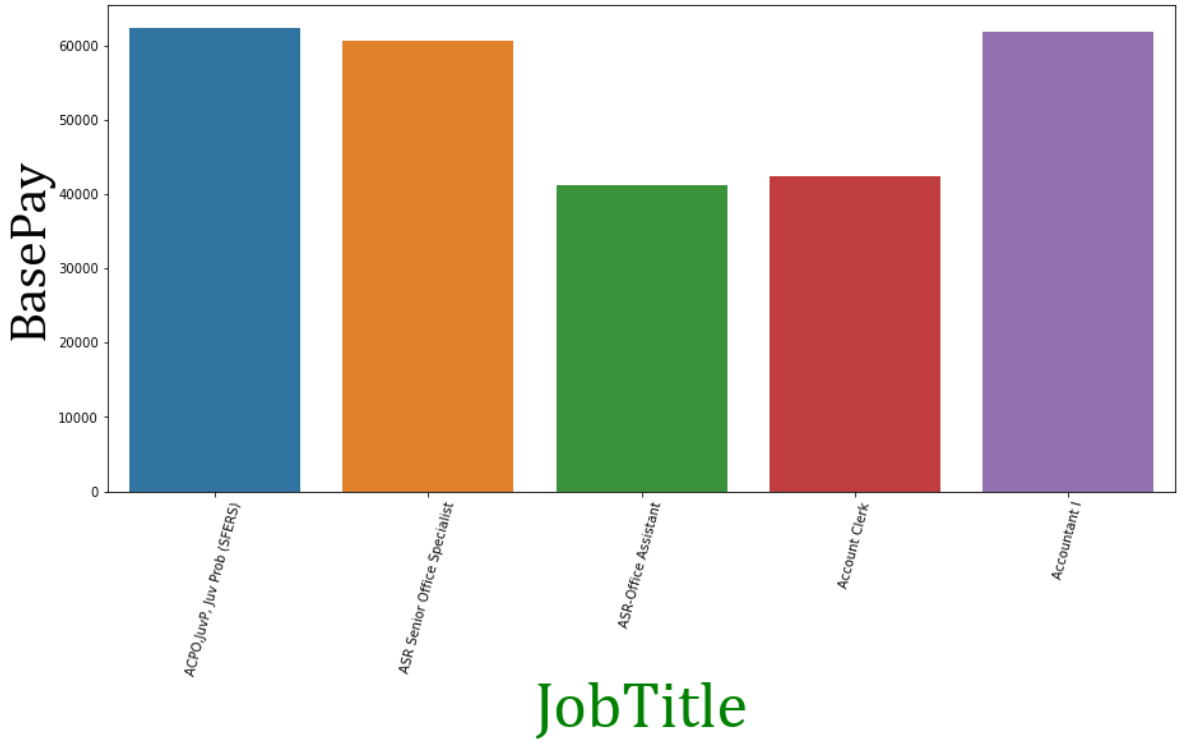
1109 rows × 7 columns

In [93]:
```
SF.groupby("JobTitle")["BasePay"].mean().reset_index().head()
```

Out[93]:

	JobTitle	BasePay
0	ACPO,JuvP, Juv Prob (SFERS)	62290.780000
1	ASR Senior Office Specialist	60551.580167
2	ASR-Office Assistant	41253.471951
3	Account Clerk	42372.579396
4	Accountant I	61777.832500

In [94]:
```
plt.figure(figsize = (15,7))
sns.barplot(x = "JobTitle", y = "BasePay", data = SF.groupby("JobTitle")["Base
plt.xlabel("JobTitle",fontdict=f2)
plt.ylabel("BasePay",fontdict=f3)
plt.xticks(rotation = 75)
plt.show()
```



In [95]:
```
plt.figure(figsize = (15,7))
sns.barplot(x = "BasePay", y = "JobTitle", data = SF.groupby("JobTitle")["Base
plt.xlabel("JobTitle",fontdict=f2)
plt.ylabel("BasePay",fontdict=f3)
plt.xticks(rotation = 75)
plt.show()
```

In [96]:
```
SF.groupby("JobTitle").mean()["BasePay"]
```

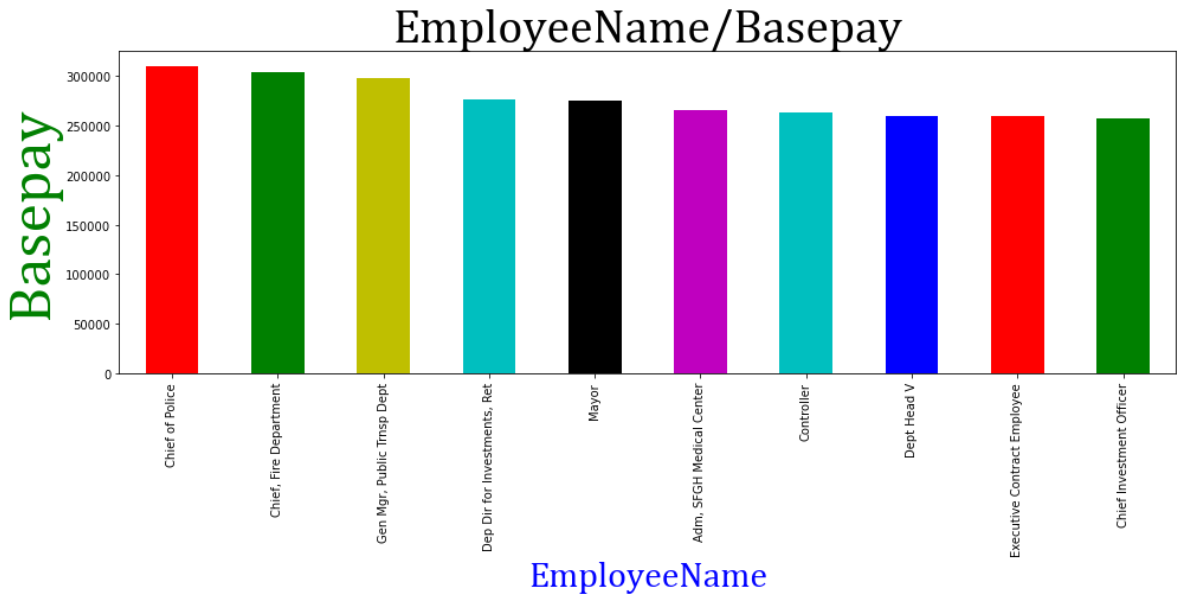
Out[96]:
```
JobTitle
ACPO,JuvP, Juv Prob (SFERS)       62290.780000
ASR Senior Office Specialist      60551.580167
ASR-Office Assistant              41253.471951
Account Clerk                     42372.579396
Accountant I                      61777.832500
                                      ...
Wire Rope Cable Maint Sprv        92751.746667
Worker's Comp Supervisor 1        68867.296429
Worker's Compensation Adjuster    72363.278784
X-Ray Laboratory Aide             46086.387100
Youth Comm Advisor                39077.957500
Name: BasePay, Length: 1109, dtype: float64
```

In [97]:
```
SF.groupby("JobTitle").mean()["BasePay"].sort_values(ascending=False).head(10)
```

Out[97]:

	JobTitle	BasePay
0	Chief of Police	309767.683333
1	Chief, Fire Department	304232.340000
2	Gen Mgr, Public Trnsp Dept	297769.413333
3	Dep Dir for Investments, Ret	276153.765000
4	Mayor	275852.530000
5	Adm, SFGH Medical Center	265218.780000
6	Controller	263588.753333
7	Dept Head V	259590.712222
8	Executive Contract Employee	259328.458333
9	Chief Investment Officer	257340.000000

In [98]:
```
cm = ["r","g","y","c","k","m","c","b"]
SF.groupby("JobTitle").mean()["BasePay"].sort_values(ascending=False).head(10)
plt.xlabel("EmployeeName", fontdict=f1)
plt.ylabel("Basepay",fontdict=f2)
plt.title("EmployeeName/Basepay", fontdict = f3)
plt.show()
```



Find the avarge basepay of employee having jobtitle accountant

In [99]:
```
SF.columns
```

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

In [100]:
```
SF["JobTitle"]=="ACCOUNTANT"
```

Out[100]:
```
36159     False
36160     False
36161     False
36162     False
36163     False
          ...
148645    False
148647    False
148648    False
148649    False
148653    False
Name: JobTitle, Length: 111886, dtype: bool
```

In [101]:
```
SF[SF['JobTitle'] == "ACCOUNTANT"]
```

Out[101]:

EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefit

In [102]:
```
SF[SF['JobTitle'] == "ACCOUNTANT"]["BasePay"]
```

Out[102]:
```
Series([], Name: BasePay, dtype: float64)
```

In [103]:
```
SF[SF['JobTitle'] == "ACCOUNTANT"]["BasePay"].mean()
```

Out[103]:
```
nan
```

In [104]:
```
SF.groupby('JobTitle')["BasePay"].mean().reset_index()
```

Out[104]:

	JobTitle	BasePay
0	ACPO,JuvP, Juv Prob (SFERS)	62290.780000
1	ASR Senior Office Specialist	60551.580167
2	ASR-Office Assistant	41253.471951
3	Account Clerk	42372.579396
4	Accountant I	61777.832500
...	...	...
1104	Wire Rope Cable Maint Sprv	92751.746667
1105	Worker's Comp Supervisor 1	68867.296429
1106	Worker's Compensation Adjuster	72363.278784
1107	X-Ray Laboratory Aide	46086.387100
1108	Youth Comm Advisor	39077.957500

1109 rows × 2 columns

In [105]:
```
SF.groupby('JobTitle')["BasePay"].mean().reset_index().head()
```

Out[105]:

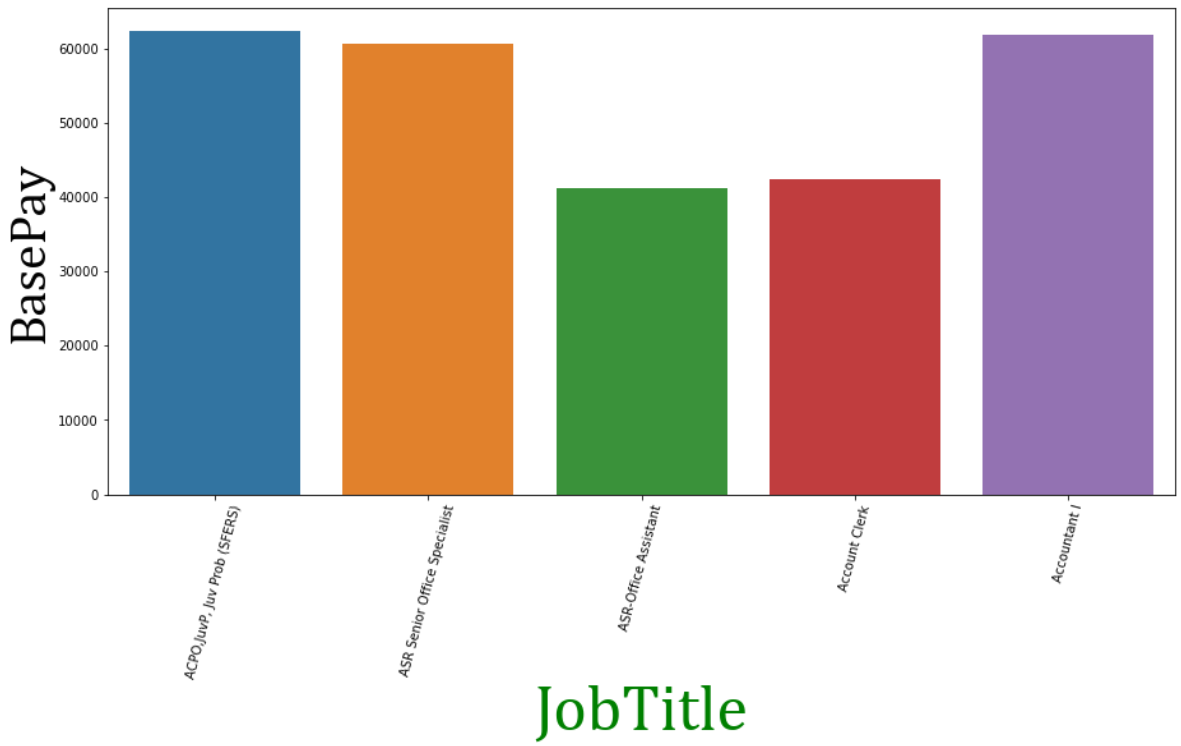
	JobTitle	BasePay
0	ACPO,JuvP, Juv Prob (SFERS)	62290.780000
1	ASR Senior Office Specialist	60551.580167
2	ASR-Office Assistant	41253.471951
3	Account Clerk	42372.579396
4	Accountant I	61777.832500

In [106]:
```
SF.groupby('JobTitle')["BasePay"].mean().reset_index().head().round()
```

Out[106]:

	JobTitle	BasePay
0	ACPO,JuvP, Juv Prob (SFERS)	62291.0
1	ASR Senior Office Specialist	60552.0
2	ASR-Office Assistant	41253.0
3	Account Clerk	42373.0
4	Accountant I	61778.0

In [107]:
```
plt.figure(figsize = (15,7))
sns.barplot(x = "JobTitle", y = "BasePay", data = SF.groupby('JobTitle')["Base
plt.xlabel("JobTitle",fontdict=f2)
plt.ylabel("BasePay",fontdict=f3)
plt.xticks(rotation = 75)
plt.show()
```



Find the 5 most common job

In [108]:
```
SF.columns
```

Out[108]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

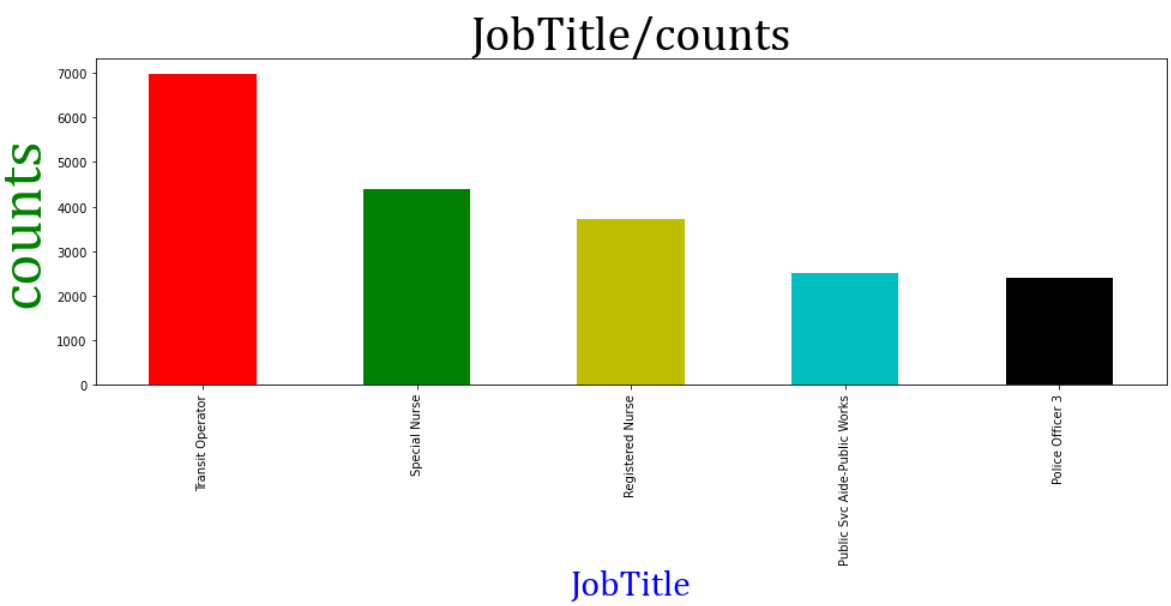
In [109]:
```
SF['JobTitle'].value_counts()
```

Out[109]:
```
Transit Operator                  6975
Special Nurse                     4382
Registered Nurse                  3725
Public Svc Aide-Public Works      2514
Police Officer 3                  2411
                                  ...
Commissioner 16.700c, No Pay         1
Chief Investment Officer             1
Chief Forensic Toxicologist          1
Lieutenant (Police Department)       1
Cashier 3                            1
Name: JobTitle, Length: 1109, dtype: int64
```

In [110]:
```
SF['JobTitle'].value_counts().head()
```

Out[110]:
```
Transit Operator                6975
Special Nurse                   4382
Registered Nurse                3725
Public Svc Aide-Public Works    2514
Police Officer 3                2411
Name: JobTitle, dtype: int64
```

In [111]:
```
SF['JobTitle'].value_counts().head().plot(kind = "bar", color = cm,figsize=(16
plt.xlabel('JobTitle', fontdict=f1)
plt.ylabel("counts",fontdict=f2)
plt.title("JobTitle/counts", fontdict = f3)
plt.show()
```



In [112]:
```
pwd
```

Out[112]:
```
'C:\\Users\\Sinha Rahul'
```