```
#Importing libraries
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# importing Dataset from local machine
df= pd.read_csv('D:\Vivek_Stuff\Learning Stuff\Python_Work\Python
Project\SF_Salaries.csv')
print('Dataset loaded !!')
```

```
Dataset loaded !!
```

```
C:\Users\180329775\AppData\Local\Temp\ipykernel_17072\2790591957.py:2:
DtypeWarning: Columns (3,4,5,6,12) have mixed types. Specify dtype
option on import or set low_memory=False.
  df= pd.read_csv('D:\Vivek_Stuff\Learning Stuff\Python_Work\Python
Project\SF_Salaries.csv')
```

Show Dataset top 5 Rows

```
# .head()
df.head()
```

```
   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         0.0  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

   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
```

Show Dataset last 5 Rows

```
# Show Dataset last 5 Rows
df.tail()

            Id   EmployeeName                     JobTitle
BasePay  \
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  \
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
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
```

Show dataset Shape

```
df.shape

# .shape is used to find the size of dataset
```

```
(148654, 13)
```

Show your dataset headers name

```
df.columns

# .columns returns the headers of Dataset
```

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

Show Complete information about the Dataset

```
df.info()

# .info() returnt the complete information of dataset , it returns the
column wise Total value count, datatype etc..
```

```
<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 your Dataset

```
df.isnull()
df.isna()

# isna() and isnull() both returns the True/False
```

```
            Id  EmployeeName  JobTitle  BasePay  OvertimePay  OtherPay
\
0        False         False     False    False        False     False

1        False         False     False    False        False     False

2        False         False     False    False        False     False

3        False         False     False    False        False     False

4        False         False     False    False        False     False

...        ...           ...       ...      ...          ...       ...

148649   False         False     False    False        False     False

148650   False         False     False    False        False     False

148651   False         False     False    False        False     False

148652   False         False     False    False        False     False

148653   False         False     False    False        False     False


        Benefits  TotalPay  TotalPayBenefits   Year  Notes  Agency
Status
0           True     False             False  False   True   False
True
1           True     False             False  False   True   False
True
2           True     False             False  False   True   False
True
3           True     False             False  False   True   False
True
4           True     False             False  False   True   False
True
...          ...       ...               ...    ...    ...     ...
...
```

```
148649     False     False             False  False   True   False
False
148650     False     False             False  False   True   False
True
148651     False     False             False  False   True   False
True
148652     False     False             False  False   True   False
True
148653     False     False             False  False   True   False
False

[148654 rows x 13 columns]
```

```
df.isna()

# .isna() returnt he Null status in True/False
```

```
           Id  EmployeeName  JobTitle  BasePay  OvertimePay  OtherPay
\
0       False         False     False    False        False     False

1       False         False     False    False        False     False

2       False         False     False    False        False     False

3       False         False     False    False        False     False

4       False         False     False    False        False     False

...       ...           ...       ...      ...          ...       ...

148649  False         False     False    False        False     False

148650  False         False     False    False        False     False

148651  False         False     False    False        False     False

148652  False         False     False    False        False     False

148653  False         False     False    False        False     False


        Benefits  TotalPay  TotalPayBenefits   Year  Notes  Agency
Status
0           True     False             False  False   True   False
True
1           True     False             False  False   True   False
True
2           True     False             False  False   True   False
True
3           True     False             False  False   True   False
```

```
True
4            True     False              False  False   True   False
True
...           ...       ...                ...    ...    ...     ...
...
148649     False     False              False  False   True   False
False
148650     False     False              False  False   True   False
True
148651     False     False              False  False   True   False
True
148652     False     False              False  False   True   False
True
148653     False     False              False  False   True   False
False

[148654 rows x 13 columns]
```

Check total Null Count in your Dataset

```
df.isnull().sum()

# .sum() to add the value ...

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

df.isna().sum()

# .sum() to add the value ...

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 Un-wanted columns from your dataset

Un-wanted Columns (ID, Notes, Agency, Status)

```
# Show columns Name
df.columns
```

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

Droping Columns below

```
df.drop(['Id','Notes','Agency','Status'], axis=1, inplace=True)

# .drop() is sued to delete/drop the rows or column
# axis =1 is used here to define that we are dropping Columns
```

Show the columns to verify that they are deleeted or not ?

```
# Show columns Name

df.columns
```

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

Get Over All Statstistic about the dataset

```
df.describe() # for Numerical data only


# .describe() returns the stastistic information
# By default it works on Numerical data
```

```
                  Id       TotalPay  TotalPayBenefits           Year
Notes
count  148654.000000  148654.000000     148654.000000  148654.000000
0.0
mean    74327.500000   74768.321972      93692.554811    2012.522643
NaN
std     42912.857795   50517.005274      62793.533483       1.117538
NaN
min         1.000000    -618.130000       -618.130000    2011.000000
NaN
25%     37164.250000   36168.995000      44065.650000    2012.000000
NaN
50%     74327.500000   71426.610000      92404.090000    2013.000000
NaN
75%    111490.750000  105839.135000     132876.450000    2014.000000
NaN
max    148654.000000  567595.430000     567595.430000    2014.000000
NaN
```

```
# for all data types we can use as below code

df.describe(include='all')
```

```
                   Id EmployeeName           JobTitle   BasePay
OvertimePay  \
count   148654.000000       148654             148654  148049.0
148654.0
unique            NaN       110811               2159  109900.0
66555.0
top               NaN    Kevin Lee  Transit Operator       0.0
0.0
freq              NaN           13               7036     875.0
66103.0
mean     74327.500000          NaN                NaN       NaN
NaN
std      42912.857795          NaN                NaN       NaN
NaN
min          1.000000          NaN                NaN       NaN
NaN
25%      37164.250000          NaN                NaN       NaN
```

```
NaN
50%      74327.500000          NaN                NaN        NaN
NaN
75%     111490.750000          NaN                NaN        NaN
NaN
max     148654.000000          NaN                NaN        NaN
NaN

        OtherPay  Benefits       TotalPay  TotalPayBenefits
Year  \
count   148654.0  112495.0  148654.000000     148654.000000
148654.000000
unique   84968.0   99635.0            NaN               NaN
NaN
top          0.0       0.0            NaN               NaN
NaN
freq     35218.0    1053.0            NaN               NaN
NaN
mean         NaN       NaN   74768.321972      93692.554811
2012.522643
std          NaN       NaN   50517.005274      62793.533483
1.117538
min          NaN       NaN    -618.130000       -618.130000
2011.000000
25%          NaN       NaN   36168.995000      44065.650000
2012.000000
50%          NaN       NaN   71426.610000      92404.090000
2013.000000
75%          NaN       NaN  105839.135000     132876.450000
2014.000000
max          NaN       NaN  567595.430000     567595.430000
2014.000000

        Notes         Agency Status
count     0.0         148654  38119
unique    NaN              1      2
top       NaN  San Francisco     FT
freq      NaN         148654  22334
mean      NaN            NaN    NaN
std       NaN            NaN    NaN
min       NaN            NaN    NaN
25%       NaN            NaN    NaN
50%       NaN            NaN    NaN
75%       NaN            NaN    NaN
max       NaN            NaN    NaN
```

Find the top 5 Employee Names

```
df['EmployeeName'].value_counts()

# .value_counts() returns the total No of count of each data appears.

# here we are counting in the Employee name column
```

```
EmployeeName
Kevin Lee                    13
Richard Lee                  11
Steven Lee                   11
William Wong                 11
Stanley Lee                   9
                             ..
Jeffrey Skover                1
Ken Dever                     1
Kevin Whitfield               1
Granville McCollough III      1
Joe Lopez                     1
Name: count, Length: 110811, dtype: int64
```

```
# below we are displaying the only top 5 Employee Names
df.EmployeeName.value_counts().head(5)
```

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

Show Qnique Job Titles avaialbe in your dataset

```
# Uniue() returns the unique value form the data

df.JobTitle.unique()
```

```
array(['GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY',
       'CAPTAIN III (POLICE DEPARTMENT)',
       'WIRE ROPE CABLE MAINTENANCE MECHANIC', ..., 'Conversion',
       'Cashier 3', 'Not provided'], dtype=object)
```

```
# nuniue() returns the total no of  unique value form the data

df.JobTitle.nunique()
```

```
2159
```

How Many Job Titles Contain 'CAPTAIN'

```
# Shpw the columns
df.columns
```

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

```
# Searching the CAPTAIN in JobTitle

df['JobTitle'].str.contains('CAPTAIN')

# .str.contains('') use dto search any string type text in the dataset
```

```
0         False
1          True
2          True
3         False
4         False
          ...
148649    False
148650    False
148651    False
148652    False
148653    False
Name: JobTitle, Length: 148654, dtype: bool
```

```
# Showing the Data of CAPTAIN JobTitle Only

df[df['JobTitle'].str.contains('CAPTAIN')]


# .str.contains('') use dto search any string type text in the dataset
```

```
          Id      EmployeeName                                  JobTitle
\
1          2      GARY JIMENEZ        CAPTAIN III (POLICE DEPARTMENT)

2          3    ALBERT PARDINI        CAPTAIN III (POLICE DEPARTMENT)

11        12  PATRICIA JACKSON        CAPTAIN III (POLICE DEPARTMENT)

17        18    SEBASTIAN WONG  CAPTAIN, EMERGENCYCY MEDICAL SERVICES

22        23     GEORGE GARCIA              CAPTAIN, FIRE SUPPRESSION

...      ...               ...                                     ...
```

```
8684    8685     JEANNE SEYLER              CAPTAIN, FIRE SUPPRESSION

10485  10486        JANE SMITH  CAPTAIN, EMERGENCYCY MEDICAL SERVICES

11198  11199     KATHRYN BROWN        CAPTAIN III (POLICE DEPARTMENT)

31297  31298   MARCO CARNIGLIA  CAPTAIN, EMERGENCYCY MEDICAL SERVICES

34124  34125     JOHN FORBES-3              CAPTAIN, FIRE SUPPRESSION


         BasePay OvertimePay   OtherPay Benefits   TotalPay
TotalPayBenefits  \
1      155966.02   245131.88  137811.38      NaN  538909.28
538909.28
2      212739.13   106088.18    16452.6      NaN  335279.91
335279.91
11       99722.0    87082.62   110804.3      NaN  297608.92
297608.92
17     140546.87   119397.26   18625.08      NaN  278569.21
278569.21
22     140546.88    93200.58   39955.25      NaN  273702.71
273702.71
...          ...         ...        ...      ...        ...
...
8684    95055.34         0.0    9197.14      NaN  104252.48
104252.48
10485    74592.0     1538.59   18804.68      NaN   94935.27
94935.27
11198    10684.5         0.0   81244.87      NaN   91929.37
91929.37
31297    9839.72         0.0    1203.77      NaN   11043.49
11043.49
34124        0.0      982.06    2277.34      NaN    3259.40
3259.40

       Year  Notes         Agency Status
1      2011    NaN  San Francisco    NaN
2      2011    NaN  San Francisco    NaN
11     2011    NaN  San Francisco    NaN
17     2011    NaN  San Francisco    NaN
22     2011    NaN  San Francisco    NaN
...     ...    ...            ...    ...
8684   2011    NaN  San Francisco    NaN
10485  2011    NaN  San Francisco    NaN
11198  2011    NaN  San Francisco    NaN
31297  2011    NaN  San Francisco    NaN
34124  2011    NaN  San Francisco    NaN
```

```
[141 rows x 13 columns]
```

```
len(df[df['JobTitle'].str.contains('CAPTAIN')])

# len() retuns the size
```

```
141
```

Display All the Employee Names from Fire Department

```
# Show the Columns
df.columns
```

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

```
df['JobTitle'].str.contains('fire', case=False)



# .str.contains('') use dto search any string type text in the dataset
# case = Fase to turn off the case sensitive search in string
```

```
0         False
1         False
2         False
3         False
4          True
          ...
148649    False
148650    False
148651    False
148652    False
148653    False
Name: JobTitle, Length: 148654, dtype: bool
```

```
df[df['JobTitle'].str.contains('fire', case=False)]

# .str.contains('') use dto search any string type text in the dataset
# case = Fase to turn off the case sensitive search in string
```

```
            Id        EmployeeName  \
4            5     PATRICK GARDNER
```

```
6            7           ALSON LEE
8            9      MICHAEL MORRIS
9           10  JOANNE HAYES-WHITE
10          11       ARTHUR KENNEY
...        ...                 ...
145956  145957    Kenneth C Farris
147556  147557       Edward A Dunn
148021  148022      Kari A Johnson
148209  148210        Sheryl K Lee
148554  148555     Lawrence F Gatt

                                                  JobTitle    BasePay  \
4           DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)   134401.6
6                     BATTALION CHIEF, (FIRE DEPARTMENT)   92492.01
8                     BATTALION CHIEF, (FIRE DEPARTMENT)  176932.64
9                CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)   285262.0
10      ASSISTANT CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)  194999.39
...                                                  ...        ...
145956                                       Firefighter       0.00
147556                                       Firefighter    1063.24
148021                                       Firefighter     688.71
148209                                       Firefighter     459.14
148554                            Fire Alarm Dispatcher      73.33

       OvertimePay   OtherPay Benefits   TotalPay  TotalPayBenefits
Year  \
4           9737.0  182234.59      NaN  326373.19         326373.19
2011
6          89062.9  134426.14      NaN  315981.05         315981.05
2011
8         86362.68   40132.23      NaN  303427.55         303427.55
2011
9              0.0   17115.73      NaN  302377.73         302377.73
2011
10        71344.88    33149.9      NaN  299494.17         299494.17
2011
...            ...        ...      ...        ...               ...
...
145956        0.00       0.00  4645.56       0.00           4645.56
2014
147556        0.00     132.90   385.66    1196.14           1581.80
2014
148021        0.00       0.00   143.39     688.71            832.10
2014
148209        0.00       0.00    95.59     459.14            554.73
2014
148554        0.00       0.00     0.73      73.33             74.06
2014

        Notes         Agency Status
```

```
4         NaN  San Francisco    NaN
6         NaN  San Francisco    NaN
8         NaN  San Francisco    NaN
9         NaN  San Francisco    NaN
10        NaN  San Francisco    NaN
...       ...            ...    ...
145956    NaN  San Francisco     PT
147556    NaN  San Francisco     PT
148021    NaN  San Francisco     PT
148209    NaN  San Francisco     PT
148554    NaN  San Francisco     PT

[5879 rows x 13 columns]
```

what are the top 5 most popular jobs ?

```
df['JobTitle'].value_counts()

# .value_counts()  returns the count of each uniue data in dataset
```

```
JobTitle
Transit Operator                                    7036
Special Nurse                                       4389
Registered Nurse                                    3736
Public Svc Aide-Public Works                        2518
Police Officer 3                                    2421
                                                    ...
CHIEF HOUSING INSPECTOR                                1
TRAFFIC SIGNAL OPERATOR                                1
COURT COMPUTER FACILITIES COORDINATOR                  1
AUTOMOTIVE BODY AND FENDER WORKER SUPERVISOR I         1
VICTIM & WITNESS TECHNICIAN                            1
Name: count, Length: 2159, dtype: int64
```

```
df['JobTitle'].value_counts().head()

# showing first 5 value with .head() method
```

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

What are the least five jobs ?

```
df['JobTitle'].value_counts().tail()

# .value_counts()  returns the count of each uniue data in dataset

# showing last 5 value with .tail() method
```

```
JobTitle
CHIEF HOUSING INSPECTOR                                1
TRAFFIC SIGNAL OPERATOR                                1
COURT COMPUTER FACILITIES COORDINATOR                  1
AUTOMOTIVE BODY AND FENDER WORKER SUPERVISOR I         1
VICTIM & WITNESS TECHNICIAN                            1
Name: count, dtype: int64
```

Find the Minimum, maximum & avg Base Pay of Employee ?

```
### Converting the BasePay datatype to Numeric dataType

df['BasePay'] = pd.to_numeric(df['BasePay'], errors='coerce')
```

Minimum BasePay

```
df['BasePay'].min()

# .min() returns the minimum value
```

```
-166.01
```

Avg BasePay

```
df['BasePay'].mean()

# .mean() returns the avg value
```

```
66325.4488404877
```

Maximum BasePay

```
df['BasePay'].max()

# .max() returns the maximum value
```

```
319275.01
```

All In One

```
df['BasePay'].describe()

# .describe() returns the Basic Stastistics
```

```
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
```

what is the highest amount of OvertimePay in the dataset ?

```
df.columns
```

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

```
# converting the float data type to Numeric datatype

df['OvertimePay']= pd.to_numeric(df['BasePay'])

# Showing Maximum Value of BasePay

df['BasePay'].max()
```

```
319275.01
```

Find the Highest Base Pay Employee Details ?

```
# filtering the data

df[df['BasePay']== df['BasePay'].max()]
```

```
          Id    EmployeeName          JobTitle    BasePay  OvertimePay
\
72925  72926  Gregory P Suhr  Chief of Police  319275.01    319275.01
```

```
       OtherPay  Benefits   TotalPay  TotalPayBenefits  Year  Notes  \
72925  20007.06  86533.21  339282.07         425815.28  2013    NaN

              Agency Status
72925  San Francisco    NaN
```

```
df['Benefits']
```

```
0                   NaN
1                   NaN
2                   NaN
3                   NaN
4                   NaN
               ...
148649             0.00
148650    Not Provided
148651    Not Provided
148652    Not Provided
148653             0.00
Name: Benefits, Length: 148654, dtype: object
```

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

```
df[df['TotalPayBenefits']== df['TotalPayBenefits'].min()]
```

```
       EmployeeName                      JobTitle  BasePay  OvertimePay
\
148653    Joe Lopez  Counselor, Log Cabin Ranch      0.0           0.0


       OtherPay Benefits  TotalPay  TotalPayBenefits  Year
148653  -618.13     0.00   -618.13           -618.13  2014
```

What is the average (mean) BasePay of all employees per year ? (2011 : 2014) ?

```
df['BasePay'].dtype

# .dtype returns the datatype

dtype('float64')
```

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


# .groupby() is used to group the data

# here we are grouping the Year column and finding the average BasePay
```

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

Replace 'Not Provided' in Employee Name to 'NaN

```
# S

df['EmployeeName']
```

```
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
```

```
df['EmployeeName'] = df['EmployeeName'].replace('Not provided',np.nan)


# .replace() is used to replace the value ()
# It takes 2 parameter, what to replace and with which to replace

# showing the EmpleName data

df['EmployeeName']
```

```
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
```

Find the Job Title of ALBERT PARDINI

```
# comparing data

df[df['EmployeeName'] == 'ALBERT PARDINI']['JobTitle']
```

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

HOW MUCH ALBERT PARDINI Make Benefits

```
# HOW MUCH ALBERT PARDINI Make Benefits
df[df['EmployeeName'] == 'ALBERT PARDINI']['TotalPayBenefits']
```

```
2    335279.91
Name: TotalPayBenefits, dtype: float64
```

Find the AVG Base Pay for All Employee per Year

```
df.columns
```

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

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

```
Year
2011    63595.956517
2012    65436.406857
2013    69630.030216
```

```
2014    66564.421924
Name: BasePay, dtype: float64
```

Find the Avg Base pay for Account Job Title of all Employee

```
df[df['JobTitle']=='ACCOUNTANT']['BasePay'].mean()
```

```
46643.172
```