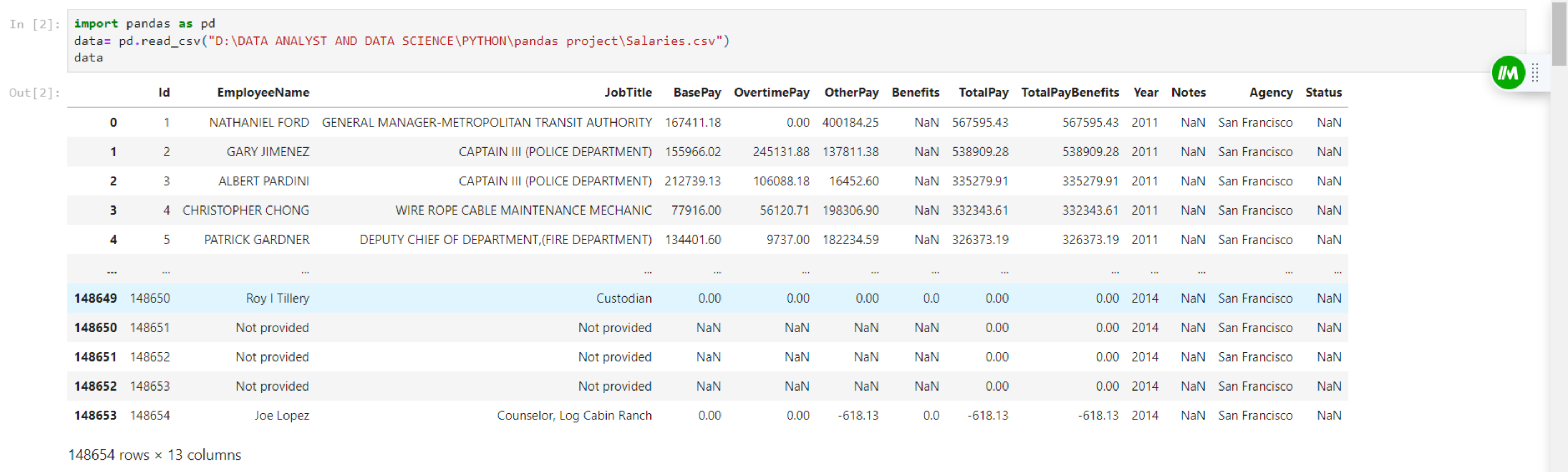
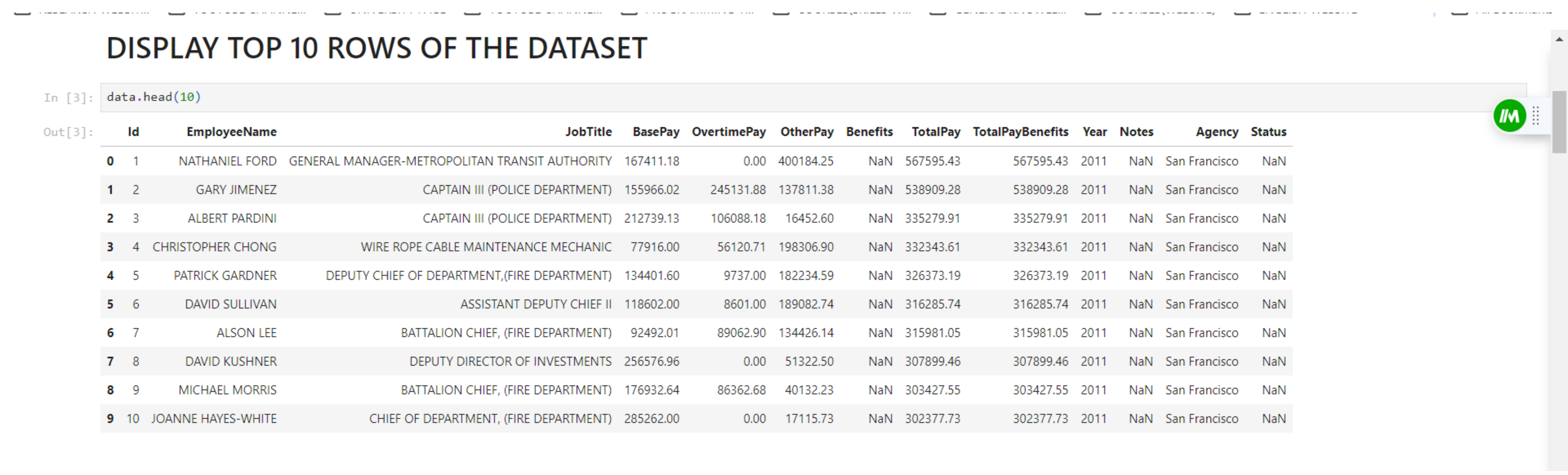


In [2]:
```python
import pandas as pd
data= pd.read_csv("D:\DATA ANALYST AND DATA SCIENCE\PYTHON\pandas project\Salaries.csv")
data
```

Out[2]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year	Notes	Agency	Status
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.00	400184.25	NaN	567595.43	567595.43	2011	NaN	San Francisco	NaN
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN	538909.28	538909.28	2011	NaN	San Francisco	NaN
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	106088.18	16452.60	NaN	335279.91	335279.91	2011	NaN	San Francisco	NaN
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.00	56120.71	198306.90	NaN	332343.61	332343.61	2011	NaN	San Francisco	NaN
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)	134401.60	9737.00	182234.59	NaN	326373.19	326373.19	2011	NaN	San Francisco	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...
148649	148650	Roy I Tillery	Custodian	0.00	0.00	0.00	0.0	0.00	0.00	2014	NaN	San Francisco	NaN
148650	148651	Not provided	Not provided	NaN	NaN	NaN	NaN	0.00	0.00	2014	NaN	San Francisco	NaN
148651	148652	Not provided	Not provided	NaN	NaN	NaN	NaN	0.00	0.00	2014	NaN	San Francisco	NaN
148652	148653	Not provided	Not provided	NaN	NaN	NaN	NaN	0.00	0.00	2014	NaN	San Francisco	NaN
148653	148654	Joe Lopez	Counselor, Log Cabin Ranch	0.00	0.00	-618.13	0.0	-618.13	-618.13	2014	NaN	San Francisco	NaN

148654 rows × 13 columns



DISPLAY TOP 10 ROWS OF THE DATASET

```
In [3]: data.head(10)
```

Out[3]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year	Notes	Agency	Status
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.00	400184.25	NaN	567595.43	567595.43	2011	NaN	San Francisco	NaN
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN	538909.28	538909.28	2011	NaN	San Francisco	NaN
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	106088.18	16452.60	NaN	335279.91	335279.91	2011	NaN	San Francisco	NaN
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.00	56120.71	198306.90	NaN	332343.61	332343.61	2011	NaN	San Francisco	NaN
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)	134401.60	9737.00	182234.59	NaN	326373.19	326373.19	2011	NaN	San Francisco	NaN
5	6	DAVID SULLIVAN	ASSISTANT DEPUTY CHIEF II	118602.00	8601.00	189082.74	NaN	316285.74	316285.74	2011	NaN	San Francisco	NaN
6	7	ALSON LEE	BATTALION CHIEF, (FIRE DEPARTMENT)	92492.01	89062.90	134426.14	NaN	315981.05	315981.05	2011	NaN	San Francisco	NaN
7	8	DAVID KUSHNER	DEPUTY DIRECTOR OF INVESTMENTS	256576.96	0.00	51322.50	NaN	307899.46	307899.46	2011	NaN	San Francisco	NaN
8	9	MICHAEL MORRIS	BATTALION CHIEF, (FIRE DEPARTMENT)	176932.64	86362.68	40132.23	NaN	303427.55	303427.55	2011	NaN	San Francisco	NaN
9	10	JOANNE HAYES-WHITE	CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	285262.00	0.00	17115.73	NaN	302377.73	302377.73	2011	NaN	San Francisco	NaN

CHECK THE LAST 10 ROWS OF THE DATASET

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

Out[5]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year	Notes	Agency	Status
148644	148645	Randy D Winn	Stationary Eng, Sewage Plant	0.0	0.0	0.00	0.0	0.00	0.00	2014	NaN	San Francisco	NaN
148645	148646	Carolyn A Wilson	Human Services Technician	0.0	0.0	0.00	0.0	0.00	0.00	2014	NaN	San Francisco	NaN
148646	148647	Not provided	Not provided	NaN	NaN	NaN	NaN	0.00	0.00	2014	NaN	San Francisco	NaN
148647	148648	Joann Anderson	Communications Dispatcher 2	0.0	0.0	0.00	0.0	0.00	0.00	2014	NaN	San Francisco	NaN
148648	148649	Leon Walker	Custodian	0.0	0.0	0.00	0.0	0.00	0.00	2014	NaN	San Francisco	NaN
148649	148650	Roy I Tillery	Custodian	0.0	0.0	0.00	0.0	0.00	0.00	2014	NaN	San Francisco	NaN
148650	148651	Not provided	Not provided	NaN	NaN	NaN	NaN	0.00	0.00	2014	NaN	San Francisco	NaN
148651	148652	Not provided	Not provided	NaN	NaN	NaN	NaN	0.00	0.00	2014	NaN	San Francisco	NaN
148652	148653	Not provided	Not provided	NaN	NaN	NaN	NaN	0.00	0.00	2014	NaN	San Francisco	NaN
148653	148654	Joe Lopez	Counselor, Log Cabin Ranch	0.0	0.0	-618.13	0.0	-618.13	-618.13	2014	NaN	San Francisco	NaN

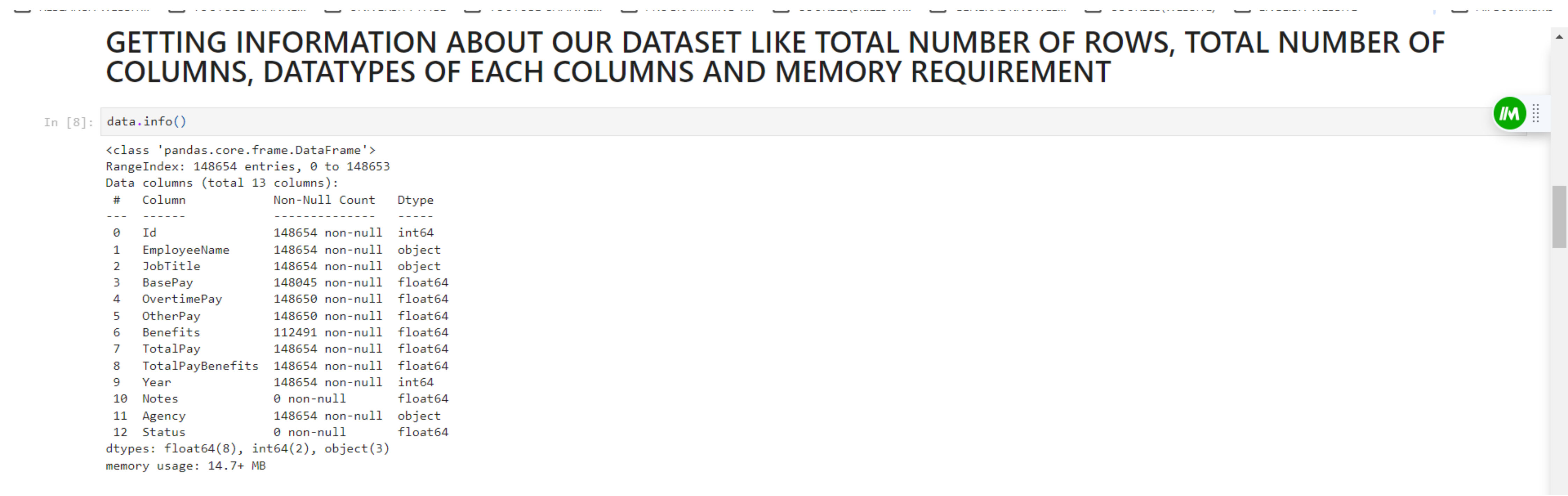
FIND SHAPE OF OUR DATASET(NUMBER OF ROWS AND NUMBER OF COLUMNS)

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

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

In [7]:
```
print("Number of Rows", data.shape[0])
print("Number of Columns", data.shape[1])
```

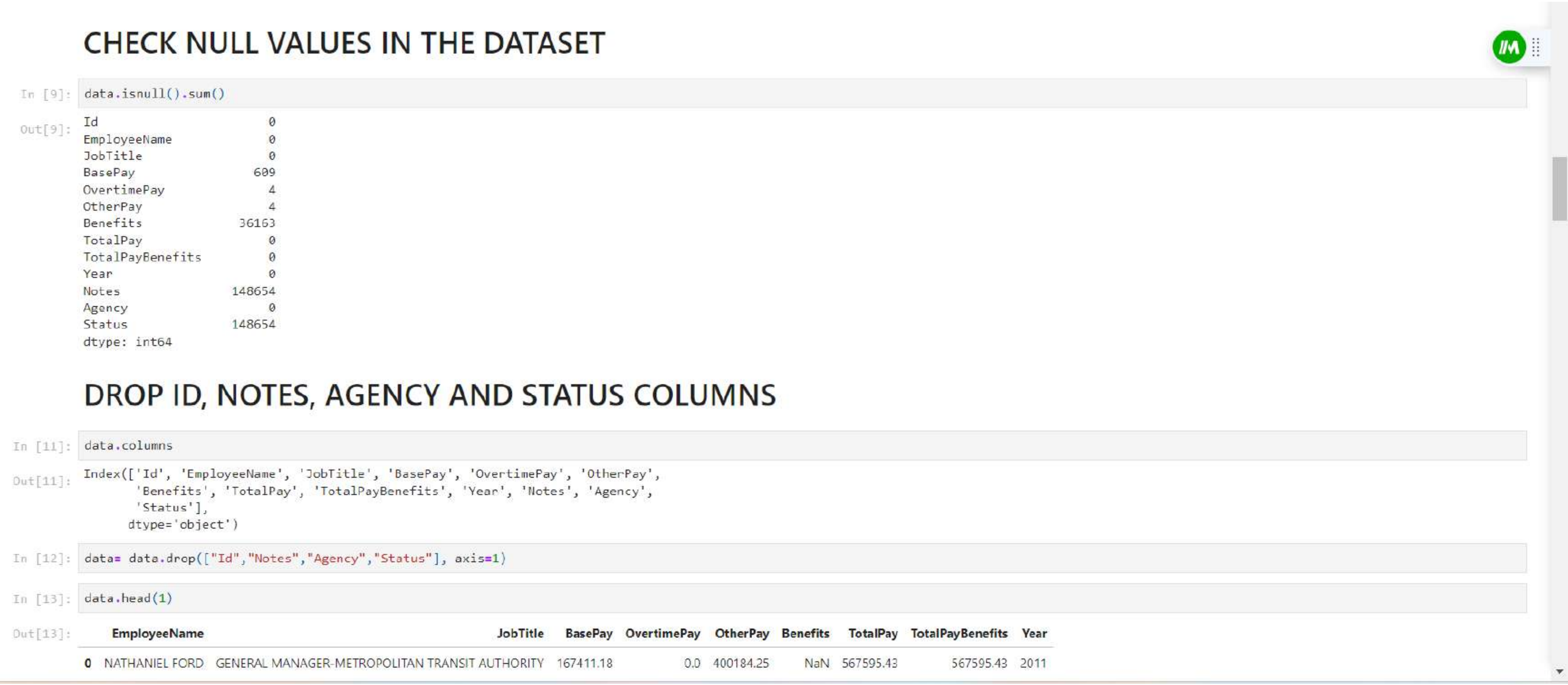
```
Number of Rows 148654
Number of Columns 13
```



GETTING INFORMATION ABOUT OUR DATASET LIKE TOTAL NUMBER OF ROWS, TOTAL NUMBER OF COLUMNS, DATATYPES OF EACH COLUMNS AND MEMORY REQUIREMENT

```
In [8]: 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
```



CHECK NULL VALUES IN THE DATASET

In [9]:
```
data.isnull().sum()
```

Out[9]:
```
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
```

DROP ID, NOTES, AGENCY AND STATUS COLUMNS

In [11]:
```
data.columns
```

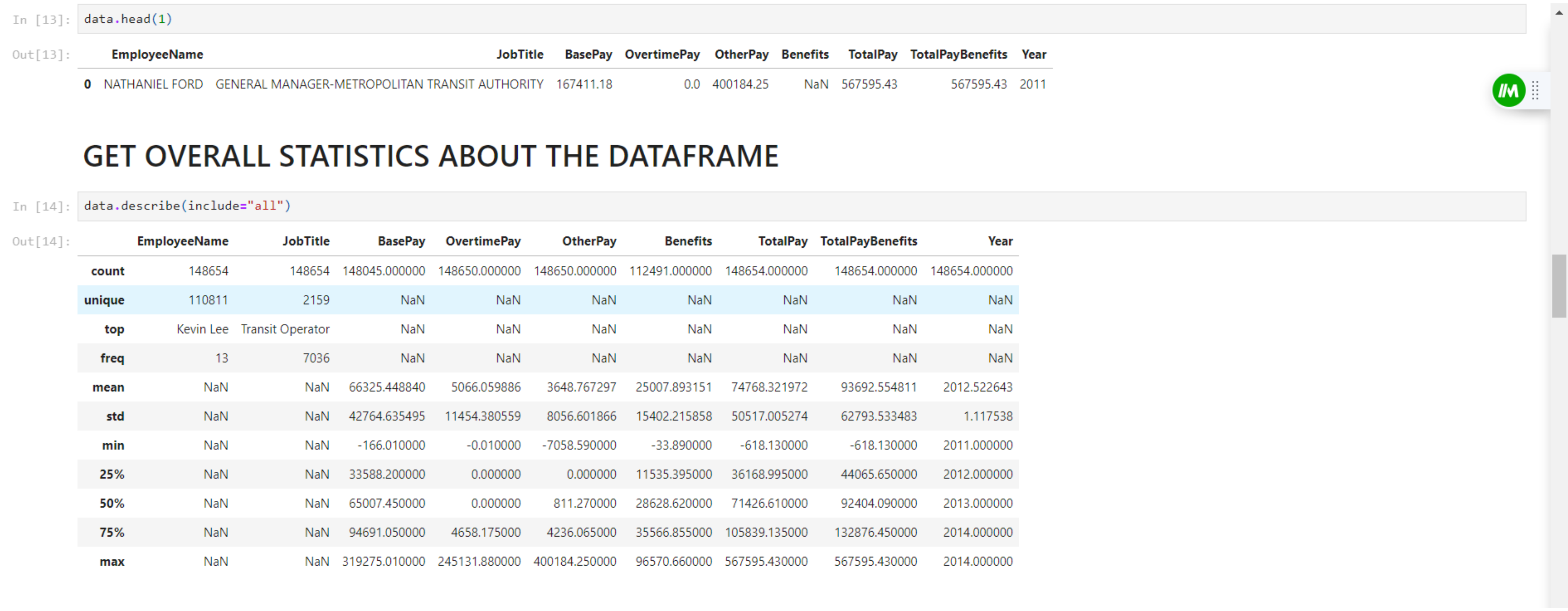
Out[11]:
```
Index(['Id', 'EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year', 'Notes', 'Agency',
       'Status'],
      dtype='object')
```

In [12]:
```
data= data.drop(["Id","Notes","Agency","Status"], axis=1)
```

In [13]:
```
data.head(1)
```

Out[13]:

	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



```
In [13]: data.head(1)
```

Out[13]:

	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

GET OVERALL STATISTICS ABOUT THE DATAFRAME

```
In [14]: data.describe(include="all")
```

Out[14]:

	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year
count	148654	148654	148045.000000	148650.000000	148650.000000	112491.000000	148654.000000	148654.000000	148654.000000
unique	110811	2159	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	Kevin Lee	Transit Operator	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freq	13	7036	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	NaN	66325.448840	5066.059886	3648.767297	25007.893151	74768.321972	93692.554811	2012.522643
std	NaN	NaN	42764.635495	11454.380559	8056.601866	15402.215858	50517.005274	62793.533483	1.117538
min	NaN	NaN	-166.010000	-0.010000	-7058.590000	-33.890000	-618.130000	-618.130000	2011.000000
25%	NaN	NaN	33588.200000	0.000000	0.000000	11535.395000	36168.995000	44065.650000	2012.000000
50%	NaN	NaN	65007.450000	0.000000	811.270000	28628.620000	71426.610000	92404.090000	2013.000000
75%	NaN	NaN	94691.050000	4658.175000	4236.065000	35566.855000	105839.135000	132876.450000	2014.000000
max	NaN	NaN	319275.010000	245131.880000	400184.250000	96570.660000	567595.430000	567595.430000	2014.000000

FIND OCCURRENCE OF THE EMPLOYEE NAMES(TOP 5)

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

```
Out[15]: Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
               'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
              dtype='object')
```

```
In [17]: data["EmployeeName"].value_counts().head()
```

```
Out[17]: Kevin Lee         13
         Richard Lee       11
         Steven Lee        11
         William Wong      11
         Stanley Lee        9
         Name: EmployeeName, dtype: int64
```

FIND THE NUMBER OF UNIQUE JOB TITLES

```
In [18]: data.columns
```

```
Out[18]: Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
               'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
              dtype='object')
```

```
In [19]: data["JobTitle"].nunique()
```

```
Out[19]: 2159
```

TOTAL NUMBER OF JOB TITLES CONTAIN CAPTAIN

In [20]:
```
data.columns
```

Out[20]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

In [24]:
```
len(data[data["JobTitle"].str.contains("CAPTAIN", case=False)])
```

Out[24]:
```
552
```

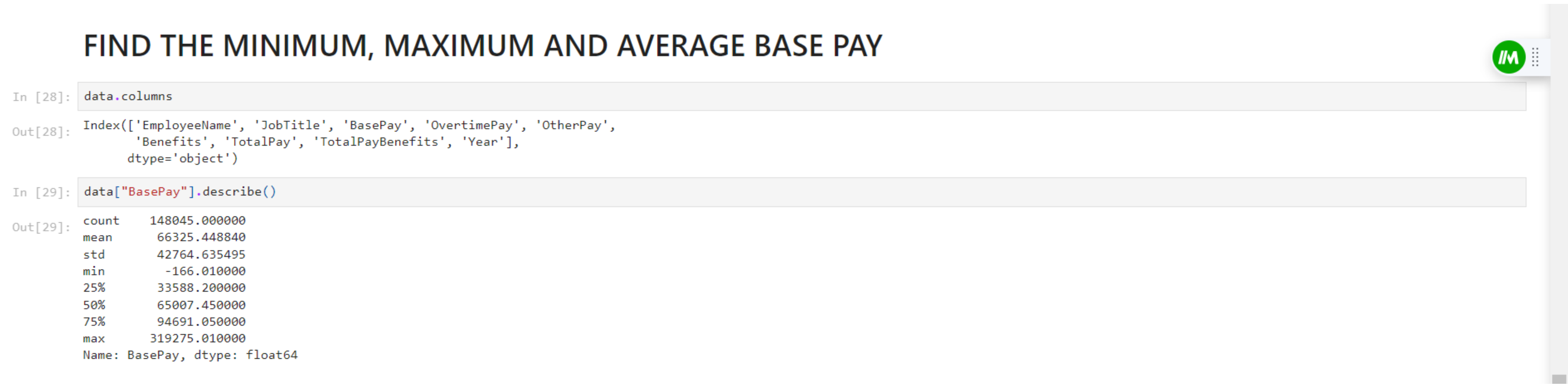
DISPLAY ALL THE EMPLOYEE NAMES FROM THE FIRE DEPARTMENT

In [25]:
```
data.columns
```

Out[25]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

In [26]:
```
data[data["JobTitle"].str.contains("fire", case=False)]["EmployeeName"]
```

Out[26]:
```
4            PATRICK GARDNER
6                  ALSON LEE
8             MICHAEL MORRIS
9          JOANNE HAYES-WHITE
10             ARTHUR KENNEY
                  ...
145956       Kenneth C Farris
147556          Edward A Dunn
148021         Kari A Johnson
148209           Sheryl K Lee
148554       Lawrence F Gatt
Name: EmployeeName, Length: 5879, dtype: object
```



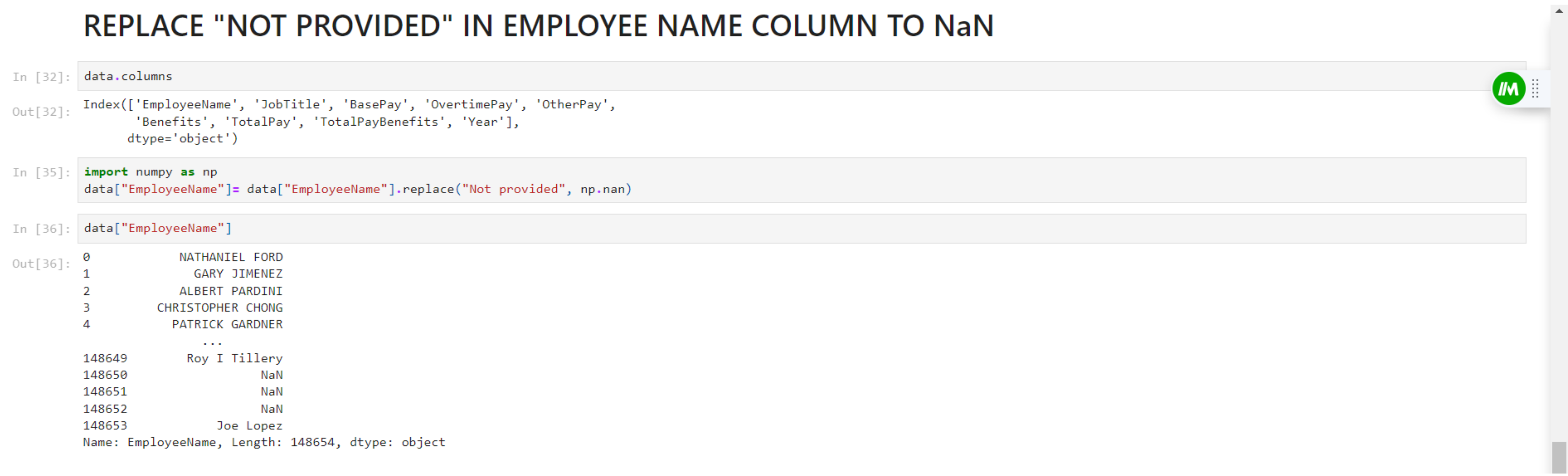
FIND THE MINIMUM, MAXIMUM AND AVERAGE BASE PAY

In [28]:
```
data.columns
```

Out[28]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

In [29]:
```
data["BasePay"].describe()
```

Out[29]:
```
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
```



REPLACE "NOT PROVIDED" IN EMPLOYEE NAME COLUMN TO NaN

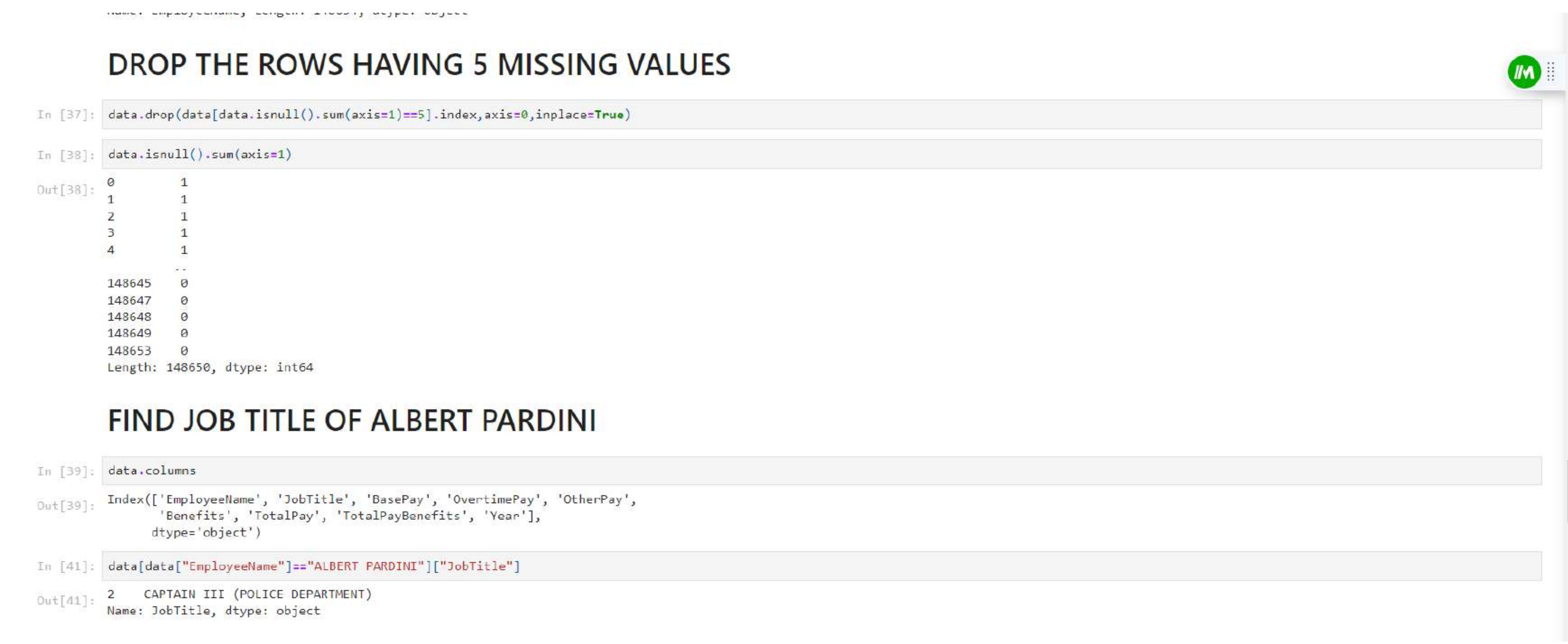
In [32]:
```
data.columns
```

Out[32]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

In [35]:
```
import numpy as np
data["EmployeeName"]= data["EmployeeName"].replace("Not provided", np.nan)
```

In [36]:
```
data["EmployeeName"]
```

Out[36]:
```
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
```



```
Name: EmployeeName, Length: 148654, dtype: object
```

DROP THE ROWS HAVING 5 MISSING VALUES

```
In [37]: data.drop(data[data.isnull().sum(axis=1)==5].index,axis=0,inplace=True)
```

```
In [38]: data.isnull().sum(axis=1)
```

```
Out[38]: 0         1
         1         1
         2         1
         3         1
         4         1
                  ..
         148645    0
         148647    0
         148648    0
         148649    0
         148653    0
         Length: 148650, dtype: int64
```

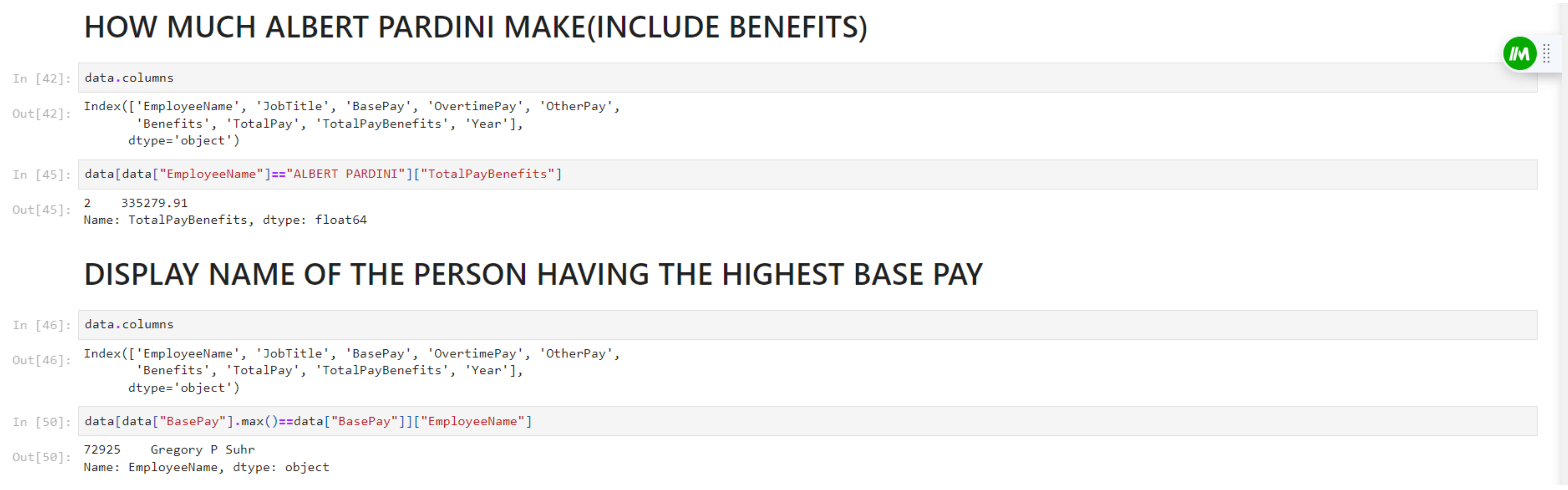
FIND JOB TITLE OF ALBERT PARDINI

```
In [39]: data.columns
```

```
Out[39]: Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
                'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
               dtype='object')
```

```
In [41]: data[data["EmployeeName"]=="ALBERT PARDINI"]["JobTitle"]
```

```
Out[41]: 2    CAPTAIN III (POLICE DEPARTMENT)
         Name: JobTitle, dtype: object
```



HOW MUCH ALBERT PARDINI MAKE(INCLUDE BENEFITS)

```
In [42]: data.columns
```

```
Out[42]: Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
               'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
              dtype='object')
```

```
In [45]: data[data["EmployeeName"]=="ALBERT PARDINI"]["TotalPayBenefits"]
```

```
Out[45]: 2    335279.91
         Name: TotalPayBenefits, dtype: float64
```

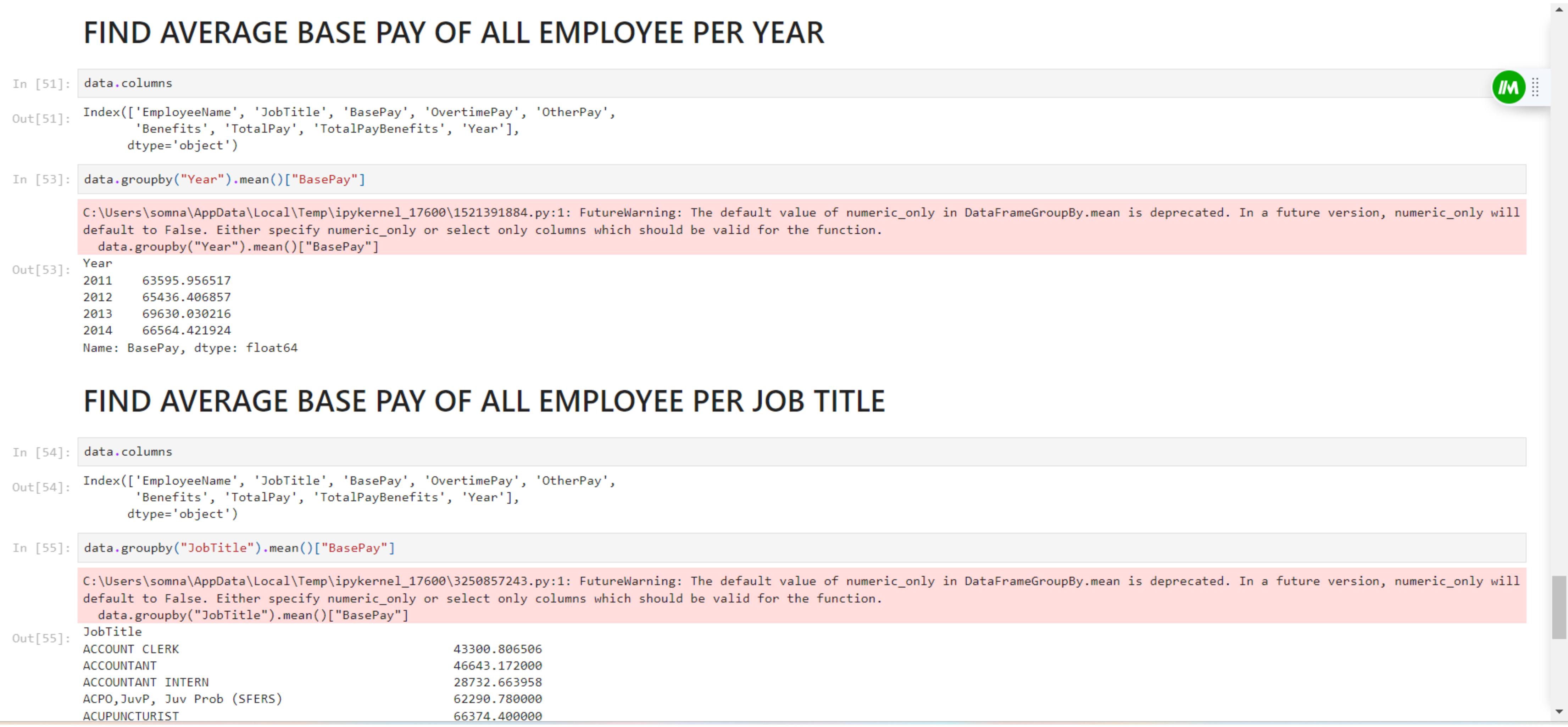
DISPLAY NAME OF THE PERSON HAVING THE HIGHEST BASE PAY

```
In [46]: data.columns
```

```
Out[46]: Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
               'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
              dtype='object')
```

```
In [50]: data[data["BasePay"].max()==data["BasePay"]]["EmployeeName"]
```

```
Out[50]: 72925    Gregory P Suhr
         Name: EmployeeName, dtype: object
```



FIND AVERAGE BASE PAY OF ALL EMPLOYEE PER YEAR

```
In [51]: data.columns

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

In [53]: data.groupby("Year").mean()["BasePay"]

C:\Users\somna\AppData\Local\Temp\ipykernel_17600\1521391884.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"]

Out[53]: Year
         2011    63595.956517
         2012    65436.406857
         2013    69630.030216
         2014    66564.421924
         Name: BasePay, dtype: float64
```

FIND AVERAGE BASE PAY OF ALL EMPLOYEE PER JOB TITLE

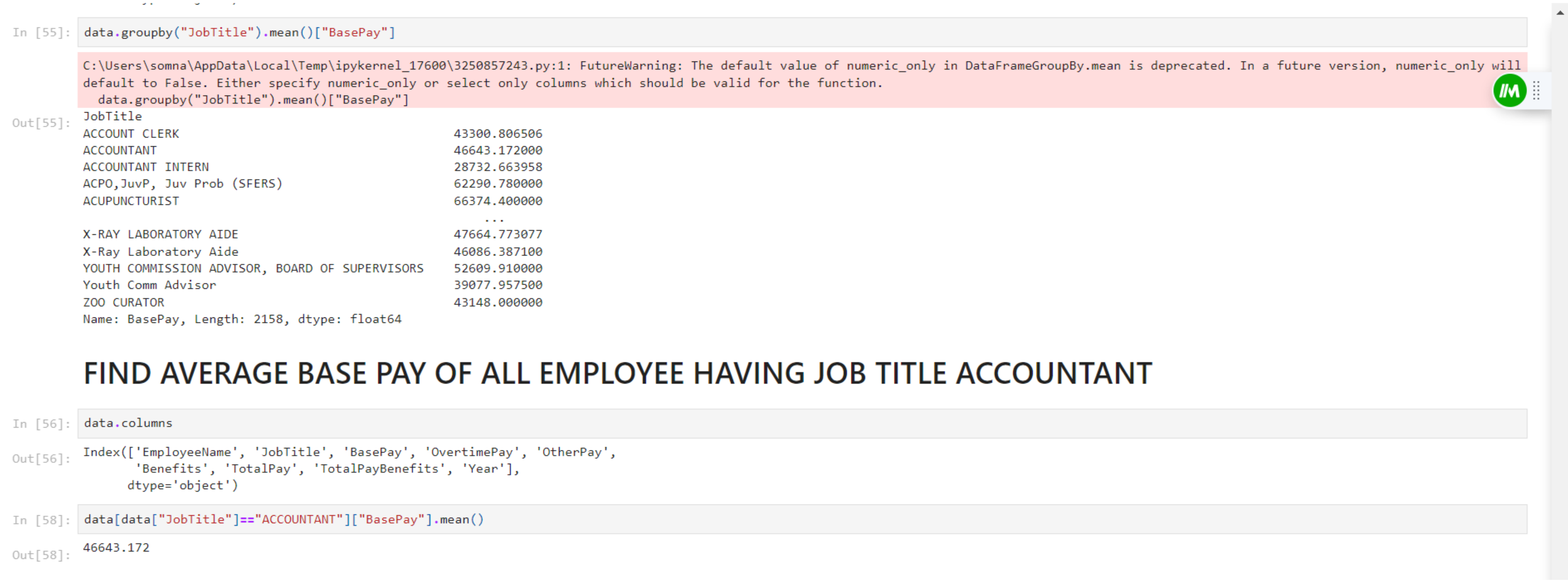
```
In [54]: data.columns

Out[54]: Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
                'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
               dtype='object')

In [55]: data.groupby("JobTitle").mean()["BasePay"]

C:\Users\somna\AppData\Local\Temp\ipykernel_17600\3250857243.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"]

Out[55]: JobTitle
         ACCOUNT CLERK                                     43300.806506
         ACCOUNTANT                                        46643.172000
         ACCOUNTANT INTERN                                 28732.663958
         ACPO,JuvP, Juv Prob (SFERS)                       62290.780000
         ACUPUNCTURIST                                     66374.400000
```



In [55]:
```
data.groupby("JobTitle").mean()["BasePay"]
```

```
C:\Users\somna\AppData\Local\Temp\ipykernel_17600\3250857243.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"]
```

Out[55]:
```
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: 2158, dtype: float64
```

FIND AVERAGE BASE PAY OF ALL EMPLOYEE HAVING JOB TITLE ACCOUNTANT

In [56]:
```
data.columns
```

Out[56]:
```
Index(['EmployeeName', 'JobTitle', 'BasePay', 'OvertimePay', 'OtherPay',
       'Benefits', 'TotalPay', 'TotalPayBenefits', 'Year'],
      dtype='object')
```

In [58]:
```
data[data["JobTitle"]=="ACCOUNTANT"]["BasePay"].mean()
```

Out[58]:
```
46643.172
```

```
Out[58]: 46643.172
```

FIND TOP 5 MOST COMMON JOBS

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

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

```
In [61]: data["JobTitle"].value_counts().head()
```

```
Out[61]: Transit Operator                7036
         Special Nurse                   4389
         Registered Nurse                3736
         Public Svc Aide-Public Works    2518
         Police Officer 3                2421
         Name: JobTitle, dtype: int64
```

```
In [ ]:
```