Import pandas

```
import numpy as np
import pandas as pd
```

Read the data from Salaries.csv and store it in a dataframe

```
df=pd.read_csv("/content/Salaries.csv")
df
```

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year	Notes	Agency
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.00	400184.25	NaN	567595.43	567595.43	2011	NaN	San Francisco
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN	538909.28	538909.28	2011	NaN	San Francisco
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	106088.18	16452.60	NaN	335279.91	335279.91	2011	NaN	San Francisco
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.00	56120.71	198306.90	NaN	332343.61	332343.61	2011	NaN	San Francisco
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
...	...	...	...	...	...	...	...	...	...	...	...	...
148649	148650	Roy I Tillery	Custodian	0.00	0.00	0.00	0.0	0.00	0.00	2014	NaN	San Francisco
148650	148651	Not provided	Not provided	NaN	NaN	NaN	NaN	0.00	0.00	2014	NaN	San Francisco

```
df
```

	U	V	W	X	Y	Z
A	82	79	40	88	32	38
B	97	90	66	71	41	17
C	72	26	92	41	69	13
D	15	51	98	27	39	44
E	90	22	74	60	20	43

Check if the dataframe is properly read or not using the head function

```
Start coding or generate with AI.
```

```
df.head(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

What columns exist in this dataframe?

```
print(df.columns)
```

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

How many rows does this dataframe have?

```
df.shape[0]
```

```
148654
```

```
df.shape[0]
```

```
5
```

Display the information about the dataframe using the info function. Which of these columns have missing values in them?

```
print(df.info())
df.isnull()
```

```
<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
None
```

```
            Id  EmployeeName  JobTitle  BasePay  OvertimePay  OtherPay  Benefits  TotalPay  TotalPayBenefits  Year  Notes  Agency  Status
     0   False         False     False    False        False     False      True     False             False False   True   False    True
     1   False         False     False    False        False     False      True     False             False False   True   False    True
     2   False         False     False    False        False     False      True     False             False False   True   False    True
     3   False         False     False    False        False     False      True     False             False False   True   False    True
     4   False         False     False    False        False     False      True     False             False False   True   False    True
   ...     ...           ...       ...      ...          ...       ...       ...       ...               ...   ...    ...     ...     ...
148649   False         False     False    False        False     False     False     False             False False   True   False    True
148650   False         False     False     True         True      True      True     False             False False   True   False    True
148651   False         False     False     True         True      True      True     False             False False   True   False    True
148652   False         False     False     True         True      True      True     False             False False   True   False    True
148653   False         False     False    False        False     False     False     False             False False   True   False    True

148654 rows × 13 columns
```

What is the total BasePay?

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

```
9819151073.590002
```

What is the highest amount of overtime pay?

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

```
245131.88
```

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

```
df[df['EmployeeName']=='Joseph Driscoll']['JobTitle'].iloc[0]
```

```
'Captain, Fire Suppression'
```

How much does JOSEPH DRISCOLL make (including benefits)?

```
df[df['EmployeeName']=='Joseph Driscoll']['TotalPayBenefits'].iloc[0]
```

```
331834.79
```

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

```
# Assuming 'Total Pay & Benefits' is the column that includes total compensation
highest_paid_person = df.loc[df['TotalPayBenefits'].idxmax()]

# Extracting the name of the highest paid person
highest_paid_name = highest_paid_person['EmployeeName']
print(f"The name of the highest paid person (including benefits) is: {highest_paid_name}")
```

```
The name of the highest paid person (including benefits) is: NATHANIEL FORD
```

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

```
# Group by year and calculate the mean BasePay for each year
average_basepay_per_year = df.groupby('Year')['BasePay'].mean()

print("Average BasePay per year from 2011 to 2014:")
print(average_basepay_per_year)
```

```
Average BasePay per year from 2011 to 2014:
Year
2011    63595.956517
2012    65436.406857
2013    69630.030216
2014    66564.421924
Name: BasePay, dtype: float64
```

Replace all the missing values in the Benefits column with 0

```
# Replace missing values in 'Benefits' column with 0
df['Benefits'] = df['Benefits'].fillna(0)

# Display the DataFrame after filling missing values
print("\nDataFrame after filling missing values:")
print(df)
```

```
DataFrame after filling missing values:
            Id       EmployeeName  \
0            1     NATHANIEL FORD
1            2       GARY JIMENEZ
2            3     ALBERT PARDINI
3            4  CHRISTOPHER CHONG
4            5    PATRICK GARDNER
...        ...                ...
148649  148650      Roy I Tillery
148650  148651       Not provided
148651  148652       Not provided
148652  148653       Not provided
148653  148654          Joe Lopez

                                             JobTitle    BasePay  \
0       GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY  167411.18
1                      CAPTAIN III (POLICE DEPARTMENT)  155966.02
2                      CAPTAIN III (POLICE DEPARTMENT)  212739.13
3                 WIRE ROPE CABLE MAINTENANCE MECHANIC   77916.00
4         DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)  134401.60
...                                               ...        ...
148649                                      Custodian       0.00
148650                                   Not provided        NaN
148651                                   Not provided        NaN
148652                                   Not provided        NaN
148653                      Counselor, Log Cabin Ranch       0.00

        OvertimePay   OtherPay  Benefits   TotalPay  TotalPayBenefits  Year  \
0              0.00  400184.25       0.0  567595.43         567595.43  2011
1         245131.88  137811.38       0.0  538909.28         538909.28  2011
2         106088.18   16452.60       0.0  335279.91         335279.91  2011
3          56120.71  198306.90       0.0  332343.61         332343.61  2011
4           9737.00  182234.59       0.0  326373.19         326373.19  2011
...             ...        ...       ...        ...               ...   ...
148649         0.00       0.00       0.0       0.00              0.00  2014
148650          NaN        NaN       0.0       0.00              0.00  2014
148651          NaN        NaN       0.0       0.00              0.00  2014
148652          NaN        NaN       0.0       0.00              0.00  2014
148653         0.00    -618.13       0.0    -618.13           -618.13  2014

        Notes         Agency  Status
0         NaN  San Francisco     NaN
1         NaN  San Francisco     NaN
```

```
2         NaN  San Francisco     NaN
3         NaN  San Francisco     NaN
4         NaN  San Francisco     NaN
...       ...            ...     ...
148649    NaN  San Francisco     NaN
148650    NaN  San Francisco     NaN
148651    NaN  San Francisco     NaN
148652    NaN  San Francisco     NaN
148653    NaN  San Francisco     NaN

[148654 rows x 13 columns]
```

How many unique job titles exist in the dataframe?

```
# Count the number of unique job titles
unique_job_titles_count = df['JobTitle'].nunique()

print(f"Number of unique job titles: {unique_job_titles_count}")
```

```
Number of unique job titles: 2159
```

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

```
import pandas as pd

# Assuming df is your DataFrame containing the data
# and it has columns 'Name' and 'Total Pay & Benefits'

# Find the row with the minimum 'Total Pay & Benefits'
lowest_paid_person = df.loc[df['TotalPayBenefits'].idxmin()]

# Extracting the name and total pay of the lowest paid person
lowest_paid_name = lowest_paid_person['EmployeeName']
lowest_paid_amount = lowest_paid_person['TotalPayBenefits']

print(f"The name of the lowest paid person (including benefits) is: {lowest_paid_name}")
print(f"Total Pay & Benefits: {lowest_paid_amount}")
```

```
The name of the lowest paid person (including benefits) is: Joe Lopez
Total Pay & Benefits: -618.13
```

What are the top 5 most common jobs?

```
t=df['JobTitle'].value_counts().head(5)
t
```

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

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

```
df_2013 = df[df['Year'] == 2013]

# Count occurrences of each job title
job_title_counts = df_2013['JobTitle'].value_counts()

# Count job titles with only one occurrence
job_titles_with_one_person = job_title_counts[job_title_counts == 1]

# Get the number of such job titles
num_job_titles_one_person = job_titles_with_one_person.shape[0]

print(f"Number of Job Titles represented by only one person in 2013: {num_job_titles_one_person}")
```

```
Number of Job Titles represented by only one person in 2013: 202
```

How many people have the word Chief in their job title?

Hint: Use lambda expression here

```
num_people_with_chief = df['JobTitle'].apply(lambda title: 'Chief' in title).sum()

print(f"Number of people with 'Chief' in their job title: {num_people_with_chief}")
```

```
Number of people with 'Chief' in their job title: 423
```