1. Import pandas

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

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

```
sal = pd.read_csv('Salaries.csv')
```

```
C:\Users\DELL\AppData\Local\Temp\ipykernel_23056\1748292466.py:1: DtypeWarning:
Columns (3,4,5,6,12) have mixed types. Specify dtype option on import or set
low_memory=False.
  sal = pd.read_csv('Salaries.csv')
```

3. Check the head of the DataFrame.

```
sal.head()
```

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Ben
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

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

```
sal.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
```

```
# Convert columns to numeric dtype
sal['BasePay'] = pd.to_numeric(sal['BasePay'], errors='coerce')
sal['OvertimePay'] = pd.to_numeric(sal['OvertimePay'], errors='coerce')
sal['OtherPay'] = pd.to_numeric(sal['OtherPay'], errors='coerce')
sal['Benefits'] = pd.to_numeric(sal['Benefits'], errors='coerce')
```

5. What is the average BasePay ?

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

```
66325.44884050643
```

6. What is the highest amount of OvertimePay in the dataset ?

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

```
245131.88
```

7. 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).

```
sal[sal['EmployeeName'] == 'JOSEPH DRISCOLL']['JobTitle']
```

```
24    CAPTAIN, FIRE SUPPRESSION
Name: JobTitle, dtype: object
```

8. How much does JOSEPH DRISCOLL make (including benefits)?

```
sal[sal['EmployeeName'] == 'JOSEPH DRISCOLL']['TotalPayBenefits']
```

```
24    270324.91
Name: TotalPayBenefits, dtype: float64
```

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

```
sal[sal['TotalPayBenefits'] == sal['TotalPayBenefits'].max()]['EmployeeName']
```

```
0    NATHANIEL FORD
Name: EmployeeName, dtype: object
```

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

```
sal[sal['TotalPayBenefits'] == sal['TotalPayBenefits'].min()]['EmployeeName']
```

```
148653    Joe Lopez
Name: EmployeeName, dtype: object
```

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

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

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

12. How many unique job titles are there?

```
sal['JobTitle'].nunique()
```

```
2159
```

13. What are the top 5 most common jobs?

```
sal['JobTitle'].value_counts().head(5)
```

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

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

```
sum(sal[sal['Year'] == 2013]['JobTitle'].value_counts() == 1)
```

```
202
```

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

```
(sal['JobTitle'].str.contains('Chief')).sum()
```

```
423
```

```
(sal['JobTitle'].str.contains('chief')).sum()
```

```
0
```

16. . Is there a correlation between length of the Job Title string and Salary?

```
sal['title_len'] = sal['JobTitle'].apply(len)
sal[['title_len', 'TotalPayBenefits']].corr()
```

	title_len	TotalPayBenefits
title_len	1.000000	-0.036878
TotalPayBenefits	-0.036878	1.000000

Missing value: NaN, None

```
.isnull()
.notnull()
.dropna()
.dropna(axis=1)
.dropna(thresh=2)
.fillna(value=5)
.fillna(value = df['Marks'].mean())
.fillna(method = 'ffill')
.fillna(method = 'bfill')
method = 'padfill'
.replace(to_replace = 'Not Provided','NaN')
```