QUESTIONS

1. Display Top 10 Rows of The Dataset
2. Check Last 10 Rows of The Dataset
3. Find Shape of Our Dataset (Number of Rows And Number of Columns)
4. Getting Information About Our Dataset Like Total Number Rows, Total Number of Columns, Datatypes of Each Column And Memory Requirement
5. Check Null Values In The Dataset
6. Drop ID, Notes, Agency, and Status Columns
7. Get Overall Statistics About The Dataframe
8. Find Occurrence of The Employee Names (Top 5)
9. Find The Number of Unique Job Titles
10. Total Number of Job Titles Contain Captain
11. Display All the Employee Names From Fire Department
12. Find Minimum, Maximum, and Average BasePay
13. Replace 'Not Provided' in EmployeeName' Column to NaN
14. Drop The Rows Having 5 Missing Values
15. Find Job Title of ALBERT PARDINI
16. How Much ALBERT PARDINI Make (Include Benefits)?
17. Display Name of The Person Having The Highest BasePay
18. Find Average BasePay of All Employee Per Year
19. Find Average BasePay of All Employee Per JobTitle
20. Find Average BasePay of Employee Having Job Title ACCOUNTANT
21. Find Top 5 Most Common Jobs

```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("kaggle/sf-salaries")

print("Path to dataset files:", path)
```

```
Path to dataset files: /root/.cache/kagglehub/datasets/kaggle/sf-salaries/versions/5
```

```
import pandas as pd
data = pd.read_csv(path + '/Salaries.csv')
```

```
<ipython-input-112-3656fd778a7f>:2: DtypeWarning: Columns (3,4,5,6,12) have mixed types. Specify dtype option on import or set low_memor
  data = pd.read_csv(path + '/Salaries.csv')
```

1. Display Top 10 Rows of The Dataset

```
data.head(10)
```

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year	Notes	Agency	Sta
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.0	400184.25	NaN	567595.43	567595.43	2011	NaN	San Francisco	N
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN	538909.28	538909.28	2011	NaN	San Francisco	N
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	106088.18	16452.6	NaN	335279.91	335279.91	2011	NaN	San Francisco	N
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.0	56120.71	198306.9	NaN	332343.61	332343.61	2011	NaN	San Francisco	N
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	134401.6	9737.0	182234.59	NaN	326373.19	326373.19	2011	NaN	San Francisco	N
5	6	DAVID SULLIVAN	ASSISTANT DEPUTY CHIEF II	118602.0	8601.0	189082.74	NaN	316285.74	316285.74	2011	NaN	San Francisco	N
6	7	ALSON LEE	BATTALION CHIEF, (FIRE DEPARTMENT)	92492.01	89062.9	134426.14	NaN	315981.05	315981.05	2011	NaN	San Francisco	N
7	8	DAVID KUSHNER	DEPUTY DIRECTOR OF INVESTMENTS	256576.96	0.0	51322.5	NaN	307899.46	307899.46	2011	NaN	San Francisco	N

2. Check Last 10 Rows of The Dataset

```
data.tail(10)
```

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

3. Find Shape of Our Dataset (Number of Rows And Number of Columns)

```
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           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
```

```
data.shape
```

```
(148654, 13)
```

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

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

4. Getting Information About Our Dataset Like Total Number Rows, Total Number of Columns, Datatypes of Each Column And Memory Requirement

```
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           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
```

5. Check Null Values In The Dataset

```
data.isnull().sum()
```

	0
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

6. Drop ID, Notes, Agency, and Status Columns

```
data.columns
```

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

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

```
data.columns
```

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

7. Get Overall Statistics About The Dataframe

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

	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year
count	148654	148654	148049.0	148654.0	148654.0	112495.0	148654.000000	148654.000000	148654.000000
unique	110811	2159	109900.0	66555.0	84968.0	99635.0	NaN	NaN	NaN
top	Kevin Lee	Transit Operator	0.0	0.0	0.0	0.0	NaN	NaN	NaN
freq	13	7036	875.0	66103.0	35218.0	1053.0	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	NaN	NaN	74768.321972	93692.554811	2012.522643
std	NaN	NaN	NaN	NaN	NaN	NaN	50517.005274	62793.533483	1.117538
min	NaN	NaN	NaN	NaN	NaN	NaN	-618.130000	-618.130000	2011.000000
25%	NaN	NaN	NaN	NaN	NaN	NaN	36168.995000	44065.650000	2012.000000
50%	NaN	NaN	NaN	NaN	NaN	NaN	71426.610000	92404.090000	2013.000000
75%	NaN	NaN	NaN	NaN	NaN	NaN	105839.135000	132876.450000	2014.000000
max	NaN	NaN	NaN	NaN	NaN	NaN	567595.430000	567595.430000	2014.000000

8. Find Occurrence of The Employee Names (Top 5)

```
data.columns
```

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

```
data['EmployeeName'].value_counts().head()
```

EmployeeName	count
Kevin Lee	13
William Wong	11
Richard Lee	11
Steven Lee	11
John Chan	9

9. Find The Number of Unique Job Titles

```
data.columns
```

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

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

```
2159
```

10. Total Number of Job Titles Contain Captain

```
data.columns
```

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

```
len(data[data['JobTitle'].str.contains('CAPTAIN',case=False)])
```

```
552
```

```
data[data['JobTitle'].str.contains('CAPTAIN',case=False)].count()
```

	0
EmployeeName	552
JobTitle	552
BasePay	551
OvertimePay	552
OtherPay	552
Benefits	411
TotalPay	552
TotalPayBenefits	552
Year	552

11. Display All the Employee Names From Fire Department

```
data.columns
```

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

```
data[data['JobTitle'].str.contains('FIRE',case=False)]
```

	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year
4	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)	134401.6	9737.0	182234.59	NaN	326373.19	326373.19	2011
6	ALSON LEE	BATTALION CHIEF, (FIRE DEPARTMENT)	92492.01	89062.9	134426.14	NaN	315981.05	315981.05	2011
8	MICHAEL MORRIS	BATTALION CHIEF, (FIRE DEPARTMENT)	176932.64	86362.68	40132.23	NaN	303427.55	303427.55	2011
9	JOANNE HAYES-WHITE	CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	285262.0	0.0	17115.73	NaN	302377.73	302377.73	2011
10	ARTHUR KENNEY	ASSISTANT CHIEF OF DEPARTMENT, (FIRE DEPARTMENT)	194999.39	71344.88	33149.9	NaN	299494.17	299494.17	2011
...	...	...	...	...	...	...	...	...	...
145956	Kenneth C Farris	Firefighter	0.00	0.00	0.00	4645.56	0.00	4645.56	2014
147556	Edward A Dunn	Firefighter	1063.24	0.00	132.90	385.66	1196.14	1581.80	2014
148021	Kari A Johnson	Firefighter	688.71	0.00	0.00	143.39	688.71	832.10	2014

```
data[data['JobTitle'].str.contains('FIRE',case=False)]['EmployeeName']
```

	EmployeeName
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

5879 rows × 1 columns

12. Find Minimum, Maximum, and Average BasePay

```
data.columns
```

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

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

	BasePay
count	148049.0
unique	109900.0
top	0.0
freq	875.0

13. Replace 'Not Provided' in EmployeeName' Column to NaN

```
import numpy as np

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

data['EmployeeName']
```

	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

148654 rows × 1 columns

14. Drop The Rows Having 5 Missing Values

```
data.columns
```

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

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

data.isnull().sum(axis=1)
```

	0
0	1
1	1
2	1
3	1
4	1
...	...
148649	0
148650	1
148651	1
148652	1
148653	0

148654 rows × 1 columns

15. Find Job Title of ALBERT PARDINI

```
data.columns
```

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

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

	JobTitle
2	CAPTAIN III (POLICE DEPARTMENT)

16. How Much ALBERT PARDINI Make (Include Benefits)?

```
data.columns
```

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

```
data[data['EmployeeName'] == 'ALBERT PARDINI']['TotalPayBenefits']
```

	TotalPayBenefits
2	335279.91

17. Display Name of The Person Having The Highest BasePay

```
data.columns
```

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

```
# Convert 'BasePay' column to numeric, handling errors
data['BasePay'] = pd.to_numeric(data['BasePay'], errors='coerce')

# Now find the person with the highest BasePay
data[data['BasePay'] == data['BasePay'].max()]
```

	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	TotalPayBenefits	Year
72925	Gregory P Suhr	Chief of Police	319275.01	0.0	20007.06	86533.21	339282.07	425815.28	2013

18. Find Average BasePay of All Employee Per Year

```
data.columns
```

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

```
# Convert 'BasePay' to numeric, forcing errors to NaN
data['BasePay'] = pd.to_numeric(data['BasePay'], errors='coerce')

# Now, find the person with the highest BasePay
highest_paid_employee = data.loc[data['BasePay'] == data['BasePay'].max(), 'EmployeeName']
print(highest_paid_employee)
```

```
72925    Gregory P Suhr
Name: EmployeeName, dtype: object
```

19. Find Average BasePay of All Employee Per JobTitle

```
# Ensure 'BasePay' is numeric to avoid errors
data['BasePay'] = pd.to_numeric(data['BasePay'], errors='coerce')

# Group by 'JobTitle' and calculate the average 'BasePay'
average_basepay_per_job = data.groupby('JobTitle')['BasePay'].mean()

# Display the result
print(average_basepay_per_job)
```

```
JobTitle
```