Lab1

Name: Daniel Mehta Student ID: n01753264

Read the Salaries.csv into a dataframe called df_data and use the head() method to check that you have read in the data correctly. Make sure you import pandas.

In [2]:

```
import pandas as pd
fp = 'Salaries.csv'

df_data = pd.read_csv(fp)

df_data.head()
```

Out[2]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	To
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.00	400184.25	NaN	567595.43	
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN	538909.28	
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	106088.18	16452.60	NaN	335279.91	
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.00	56120.71	198306.90	NaN	332343.61	
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)	134401.60	9737.00	182234.59	NaN	326373.19	

Use the dtypes attribute to view how each column is stored

In [4]:

```
df_data.dtypes
```

Out[4]:

```
Id                    int64
EmployeeName         object
JobTitle             object
BasePay             float64
OvertimePay         float64
OtherPay            float64
Benefits            float64
TotalPay            float64
TotalPayBenefits    float64
Year                  int64
Notes               float64
Agency               object
Status              float64
dtype: object
```

Slice the first two columns using .loc and store the result in a variable.

In [5]:

```
#Write you code here
result_1 = df_data.loc[:, ['Id','EmployeeName']]
result_1.head()
```

Out[5]:

	Id	EmployeeName
0	1	NATHANIEL FORD
1	2	GARY JIMENEZ
2	3	ALBERT PARDINI
3	4	CHRISTOPHER CHONG
4	5	PATRICK GARDNER

Slice the first two rows using .loc and store the result in a variable called result_2.

In [6]:

```
#Write you code here
result_2 = df_data.loc[0:1, :]
result_2.head()
```

Out[6]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	To
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.00	400184.25	NaN	567595.43	
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN	538909.28	

Slice the first four rows and the first five columns and store the result in a variable called result_3.

In [7]:

```
#Write you code here
result_3 = df_data.iloc[0:3,0:4]
result_3.head()
```

Out[7]:

	Id	EmployeeName	JobTitle	BasePay
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13

Slice rows 0,4,6 and select two columns randomly and store the result in variable called result_4.

In [8]:

```
#Write you code here
import random
rows = df_data.iloc[[0, 4, 6]]
random_columns = random.sample(list(df_data.columns), 2)
result_4 = rows[random_columns]
result_4.head()
```

Out[8]:

	Year	Agency
0	2011	San Francisco
4	2011	San Francisco
6	2011	San Francisco

Store the number rows in a variable called num_rows.

In [9]:

```
#Write you code here
num_rows = len(df_data.index)
num_rows
```

Out[9]:

```
148654
```

Print out the last row of the data to dataframe.

In [12]:

```
#Write you code here
df_data.tail(1)
```

Out[12]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay
148653	148654	Joe Lopez	Counselor, Log Cabin Ranch	0.0	0.0	-618.13	0.0	-618.13

In [ ]:

```
```

Compute the average and max TotalPay. Store the results in variables called avg_TotalPay and max_TotalPay

In [14]:

```
#Write your code here
avg_TotalPay = df_data.TotalPay.mean()
max_TotalPay = df_data.TotalPay.max()

print(f"Average: {avg_TotalPay}")
print(f"Max: {max_TotalPay}")
```

```
Average: 74768.32197169267
Max: 567595.43
```

Create a column called "final", which is BasePay*2.

In [16]:

```
#Write your code here
df_data["final"] = df_data.BasePay*2
df_data.head()
```

Out[16]:

	Id	EmployeeName	JobTitle	BasePay	OvertimePay	OtherPay	Benefits	TotalPay	To
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	0.00	400184.25	NaN	567595.43	
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	245131.88	137811.38	NaN	538909.28	
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	106088.18	16452.60	NaN	335279.91	
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.00	56120.71	198306.90	NaN	332343.61	
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)	134401.60	9737.00	182234.59	NaN	326373.19	

Use the drop() method to delete the column OvertimePay from the dataframe df_data.

In [20]:

```
#Write your code here
df_data.drop(["OvertimePay"], axis = 1, inplace = True)
df_data.head()
```

Out[20]:

	Id	EmployeeName	JobTitle	BasePay	OtherPay	Benefits	TotalPay	TotalPayBenefits
0	1	NATHANIEL FORD	GENERAL MANAGER-METROPOLITAN TRANSIT AUTHORITY	167411.18	400184.25	NaN	567595.43	567595.43
1	2	GARY JIMENEZ	CAPTAIN III (POLICE DEPARTMENT)	155966.02	137811.38	NaN	538909.28	538909.28
2	3	ALBERT PARDINI	CAPTAIN III (POLICE DEPARTMENT)	212739.13	16452.60	NaN	335279.91	335279.91
3	4	CHRISTOPHER CHONG	WIRE ROPE CABLE MAINTENANCE MECHANIC	77916.00	198306.90	NaN	332343.61	332343.61
4	5	PATRICK GARDNER	DEPUTY CHIEF OF DEPARTMENT,(FIRE DEPARTMENT)	134401.60	182234.59	NaN	326373.19	326373.19

In this set of practice exercises, we will be working with a demographic data regarding the passengers aboard the Titanic. Read in the data frame and use the head() method to check that it was read in correctly.

In [22]:

```python
import pandas as pd
#Write your code here
fp = 'Titanic.csv'

df_data = pd.read_csv(fp)

df_data.head()
```

Out[22]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

Use the rename method to change the column "Name" to "Passenger_Name" and the column "Ticket" to "Ticket_Num".

In [25]:

```
#Write your code here
df_data.rename(columns={"Name": "Passenger_Name", "Ticket": "Ticket_Num"}, inplace=True)
df_data.head()
```

Out[25]:

	PassengerId	Pclass	Passenger_Name	Sex	Age	SibSp	Parch	Ticket_Num	Fare	Cabin
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN

Select the name of passenger 896

In [35]:

```
#Write your code here
passenger_name = df_data.loc[df_data['PassengerId'] == 896, 'Passenger_Name'].iloc[0]
print(passenger_name)
```

```
Hirvonen, Mrs. Alexander (Helga E Lindqvist)
```

How many missing entries are there in the Age column?

In [37]:

```
#Write you code here
df_data.isnull().sum()["Age"]
```

Out[37]:
```
86
```

Compute the avg age of passengers ignoring the missing data.

In [40]:
```
#Write your code here
df_data.Age.mean(skipna=True)
```

Out[40]:
```
30.272590361445783
```

Using the fillna() method replace the missing values in the Age column with the mean.

In [44]:
```
#Write your code here
df_data['Age'] = df_data.Age.fillna(df_data.Age.mean())
df_data['Age']
```

Out[44]:
```
0      34.50000
1      47.00000
2      62.00000
3      27.00000
4      22.00000
         ...
413    30.27259
414    39.00000
415    38.50000
416    30.27259
417    30.27259
Name: Age, Length: 418, dtype: float64
```

In [ ]:
```
#Bonus: for students who wants to practice more
```

What is the average age of the 5 oldest passengers? The reset_index method will be helpful here.

In [47]:
```
#Write your code here
sorted_df = df_data.sort_values(by='Age', ascending=False).reset_index(drop=True)
oldest_passengers = sorted_df.head(5)
average_age = oldest_passengers['Age'].mean()
average_age
```

Out[47]:
```
67.0
```

In [ ]:
```

```