Jenit Harnesha

21052158

CSE 36

Test

In [2]:

```
import pandas as pd
df=pd.read_csv(r"C:\Users\KIIT\Downloads\Salary_Classification.csv")
df
```

Out[2]:

	Department	WorkedHours	Certification	YearsExperience	Salary
0	Development	2300	0	1.1	39343
1	Testing	2100	1	1.3	46205
2	Development	2104	2	1.5	37731
3	UX	1200	1	2.0	43525
4	Testing	1254	2	2.2	39891
5	UX	1236	1	2.9	56642
6	Development	1452	2	3.0	60150
7	Testing	1789	1	3.2	54445
8	UX	1645	1	3.2	64445
9	UX	1258	0	3.7	57189
10	Testing	1478	3	3.9	63218
11	Development	1257	2	4.0	55794
12	Development	1596	1	4.0	56957
13	Testing	1256	2	4.1	57081
14	UX	1489	3	4.5	61111
15	Development	1236	3	4.9	67938
16	Testing	2311	2	5.1	66029
17	UX	2245	3	5.3	83088
18	Development	2365	1	5.9	81363
19	Development	1500	3	6.0	93940
20	Testing	1456	2	6.8	91738
21	Testing	1760	1	7.1	98273
22	UX	2400	4	7.9	101302
23	Development	2148	3	8.2	113812
24	UX	1450	2	8.7	109431
25	UX	1000	4	9.0	105582
26	Testing	1540	3	9.5	116969
27	Development	1500	2	9.6	112635
28	Testing	3000	4	10.3	122391
29	UX	2100	3	10.5	121872

1. How do I select a specific column in a Pandas DataFrame?

In [4]:
```
selected_column = df['Salary']
selected_column
```

Out[4]:
```
0      39343
1      46205
2      37731
3      43525
4      39891
5      56642
6      60150
7      54445
8      64445
9      57189
10     63218
11     55794
12     56957
13     57081
14     61111
15     67938
16     66029
17     83088
18     81363
19     93940
20     91738
21     98273
22    101302
23    113812
24    109431
25    105582
26    116969
27    112635
28    122391
29    121872
Name: Salary, dtype: int64
```

2. How do I filter rows in a Pandas DataFrame based on a condition?

In [9]:
```
filtered_df = df[df['YearsExperience'] > 5]
filtered_df
```

Out[9]:

	Department	WorkedHours	Certification	YearsExperience	Salary
16	Testing	2311	2	5.1	66029
17	UX	2245	3	5.3	83088
18	Development	2365	1	5.9	81363
19	Development	1500	3	6.0	93940
20	Testing	1456	2	6.8	91738
21	Testing	1760	1	7.1	98273
22	UX	2400	4	7.9	101302
23	Development	2148	3	8.2	113812
24	UX	1450	2	8.7	109431
25	UX	1000	4	9.0	105582
26	Testing	1540	3	9.5	116969
27	Development	1500	2	9.6	112635
28	Testing	3000	4	10.3	122391
29	UX	2100	3	10.5	121872

3. How do I rename columns in a Pandas DataFrame?

In [10]:
```
df.rename(columns={'YearsExperience': 'Experience'}, inplace=True)
df
```

Out[10]:

	Department	WorkedHours	Certification	Experience	Salary
0	Development	2300	0	1.1	39343
1	Testing	2100	1	1.3	46205
2	Development	2104	2	1.5	37731
3	UX	1200	1	2.0	43525
4	Testing	1254	2	2.2	39891
5	UX	1236	1	2.9	56642
6	Development	1452	2	3.0	60150
7	Testing	1789	1	3.2	54445
8	UX	1645	1	3.2	64445
9	UX	1258	0	3.7	57189
10	Testing	1478	3	3.9	63218
11	Development	1257	2	4.0	55794
12	Development	1596	1	4.0	56957
13	Testing	1256	2	4.1	57081
14	UX	1489	3	4.5	61111
15	Development	1236	3	4.9	67938
16	Testing	2311	2	5.1	66029
17	UX	2245	3	5.3	83088
18	Development	2365	1	5.9	81363
19	Development	1500	3	6.0	93940
20	Testing	1456	2	6.8	91738
21	Testing	1760	1	7.1	98273
22	UX	2400	4	7.9	101302
23	Development	2148	3	8.2	113812
24	UX	1450	2	8.7	109431
25	UX	1000	4	9.0	105582
26	Testing	1540	3	9.5	116969
27	Development	1500	2	9.6	112635
28	Testing	3000	4	10.3	122391
29	UX	2100	3	10.5	121872

4. How do I sort a Pandas DataFrame by a specific column?

In [14]:

```
sorted_df = df.sort_values(by='Salary', ascending=True)
sorted_df
```

Out[14]:

	Department	WorkedHours	Certification	Experience	Salary
2	Development	2104	2	1.5	37731
0	Development	2300	0	1.1	39343
4	Testing	1254	2	2.2	39891
3	UX	1200	1	2.0	43525
1	Testing	2100	1	1.3	46205
7	Testing	1789	1	3.2	54445
11	Development	1257	2	4.0	55794
5	UX	1236	1	2.9	56642
12	Development	1596	1	4.0	56957
13	Testing	1256	2	4.1	57081
9	UX	1258	0	3.7	57189
6	Development	1452	2	3.0	60150
14	UX	1489	3	4.5	61111
10	Testing	1478	3	3.9	63218
8	UX	1645	1	3.2	64445
16	Testing	2311	2	5.1	66029
15	Development	1236	3	4.9	67938
18	Development	2365	1	5.9	81363
17	UX	2245	3	5.3	83088
20	Testing	1456	2	6.8	91738
19	Development	1500	3	6.0	93940
21	Testing	1760	1	7.1	98273
22	UX	2400	4	7.9	101302
25	UX	1000	4	9.0	105582
24	UX	1450	2	8.7	109431
27	Development	1500	2	9.6	112635
23	Development	2148	3	8.2	113812
26	Testing	1540	3	9.5	116969
29	UX	2100	3	10.5	121872
28	Testing	3000	4	10.3	122391

5. How do I drop duplicate rows in a Pandas DataFrame?

In [15]:
```
cleaned_df = df.drop_duplicates()
cleaned_df
```

Out[15]:

	Department	WorkedHours	Certification	Experience	Salary
0	Development	2300	0	1.1	39343
1	Testing	2100	1	1.3	46205
2	Development	2104	2	1.5	37731
3	UX	1200	1	2.0	43525
4	Testing	1254	2	2.2	39891
5	UX	1236	1	2.9	56642
6	Development	1452	2	3.0	60150
7	Testing	1789	1	3.2	54445
8	UX	1645	1	3.2	64445
9	UX	1258	0	3.7	57189
10	Testing	1478	3	3.9	63218
11	Development	1257	2	4.0	55794
12	Development	1596	1	4.0	56957
13	Testing	1256	2	4.1	57081
14	UX	1489	3	4.5	61111
15	Development	1236	3	4.9	67938
16	Testing	2311	2	5.1	66029
17	UX	2245	3	5.3	83088
18	Development	2365	1	5.9	81363
19	Development	1500	3	6.0	93940
20	Testing	1456	2	6.8	91738
21	Testing	1760	1	7.1	98273
22	UX	2400	4	7.9	101302
23	Development	2148	3	8.2	113812
24	UX	1450	2	8.7	109431
25	UX	1000	4	9.0	105582
26	Testing	1540	3	9.5	116969
27	Development	1500	2	9.6	112635
28	Testing	3000	4	10.3	122391
29	UX	2100	3	10.5	121872

6. How do I calculate summary statistics (mean, median, etc.) for a Pandas DataFrame?

In [17]:

```
mean = df['Salary'].mean()
median = df['Salary'].median()
print("Mean: ",mean)
print("Median: ",median)
```

```
Mean:  76003.0
Median:  65237.0
```

7. How do I add a new column to a Pandas DataFrame?

In [22]:
```
df['Gender'] = ['M','F','F','M','F','M','F','F','M','F','M','F','F','M'
df
```

Out[22]:

	Department	WorkedHours	Certification	Experience	Salary	Gender
0	Development	2300	0	1.1	39343	M
1	Testing	2100	1	1.3	46205	F
2	Development	2104	2	1.5	37731	F
3	UX	1200	1	2.0	43525	M
4	Testing	1254	2	2.2	39891	F
5	UX	1236	1	2.9	56642	M
6	Development	1452	2	3.0	60150	F
7	Testing	1789	1	3.2	54445	F
8	UX	1645	1	3.2	64445	M
9	UX	1258	0	3.7	57189	F
10	Testing	1478	3	3.9	63218	M
11	Development	1257	2	4.0	55794	F
12	Development	1596	1	4.0	56957	F
13	Testing	1256	2	4.1	57081	M
14	UX	1489	3	4.5	61111	F
15	Development	1236	3	4.9	67938	M
16	Testing	2311	2	5.1	66029	F
17	UX	2245	3	5.3	83088	F
18	Development	2365	1	5.9	81363	M
19	Development	1500	3	6.0	93940	F
20	Testing	1456	2	6.8	91738	M
21	Testing	1760	1	7.1	98273	F
22	UX	2400	4	7.9	101302	F
23	Development	2148	3	8.2	113812	M
24	UX	1450	2	8.7	109431	F
25	UX	1000	4	9.0	105582	M
26	Testing	1540	3	9.5	116969	F
27	Development	1500	2	9.6	112635	F
28	Testing	3000	4	10.3	122391	M
29	UX	2100	3	10.5	121872	F

8. How do I group a Pandas DataFrame by one or multiple columns?

In [24]:
```
grouped_df = df.groupby(['Certification', 'WorkedHours'])
grouped_df
```

Out[24]:
```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000001FF60AF
EEB0>
```

9.How do I perform left, right, or outer join on two Pandas DataFrames?

In [ ]:
```
df1=pd.read_csv(r"file path")
df2=pd.read_csv(r"file path")
merged_df_left = pd.merge(df1, df2, how='left', on='CommonColumn')
merged_df_right = pd.merge(df1, df2, how='right', on='CommonColumn')
merged_df_outer = pd.merge(df1, df2, how='outer', on='CommonColumn')
print(merged_df_left)
print(merged_df_right)
print(merged_df_outer)
```

10. How do I handle missing data in a Pandas DataFrame?

In [28]:
```
cleaned_df = df.dropna()
filled_df = df.fillna(0)
df['Salary'].fillna(0, inplace=True)
df
```

Out[28]:

	Department	WorkedHours	Certification	Experience	Salary	Gender
0	Development	2300	0	1.1	39343	M
1	Testing	2100	1	1.3	46205	F
2	Development	2104	2	1.5	37731	F
3	UX	1200	1	2.0	43525	M
4	Testing	1254	2	2.2	39891	F
5	UX	1236	1	2.9	56642	M
6	Development	1452	2	3.0	60150	F
7	Testing	1789	1	3.2	54445	F
8	UX	1645	1	3.2	64445	M
9	UX	1258	0	3.7	57189	F
10	Testing	1478	3	3.9	63218	M
11	Development	1257	2	4.0	55794	F
12	Development	1596	1	4.0	56957	F
13	Testing	1256	2	4.1	57081	M
14	UX	1489	3	4.5	61111	F
15	Development	1236	3	4.9	67938	M
16	Testing	2311	2	5.1	66029	F
17	UX	2245	3	5.3	83088	F
18	Development	2365	1	5.9	81363	M
19	Development	1500	3	6.0	93940	F
20	Testing	1456	2	6.8	91738	M
21	Testing	1760	1	7.1	98273	F
22	UX	2400	4	7.9	101302	F
23	Development	2148	3	8.2	113812	M
24	UX	1450	2	8.7	109431	F
25	UX	1000	4	9.0	105582	M
26	Testing	1540	3	9.5	116969	F
27	Development	1500	2	9.6	112635	F
28	Testing	3000	4	10.3	122391	M
29	UX	2100	3	10.5	121872	F

11.How do I convert a Pandas DataFrame to a NumPy array?

In [31]:
```
numpy_array = df.to_numpy()
numpy_array
```

Out[31]:
```
array([['Development', 2300, 0, 1.1, 39343, 'M'],
       ['Testing', 2100, 1, 1.3, 46205, 'F'],
       ['Development', 2104, 2, 1.5, 37731, 'F'],
       ['UX', 1200, 1, 2.0, 43525, 'M'],
       ['Testing', 1254, 2, 2.2, 39891, 'F'],
       ['UX', 1236, 1, 2.9, 56642, 'M'],
       ['Development', 1452, 2, 3.0, 60150, 'F'],
       ['Testing', 1789, 1, 3.2, 54445, 'F'],
       ['UX', 1645, 1, 3.2, 64445, 'M'],
       ['UX', 1258, 0, 3.7, 57189, 'F'],
       ['Testing', 1478, 3, 3.9, 63218, 'M'],
       ['Development', 1257, 2, 4.0, 55794, 'F'],
       ['Development', 1596, 1, 4.0, 56957, 'F'],
       ['Testing', 1256, 2, 4.1, 57081, 'M'],
       ['UX', 1489, 3, 4.5, 61111, 'F'],
       ['Development', 1236, 3, 4.9, 67938, 'M'],
       ['Testing', 2311, 2, 5.1, 66029, 'F'],
       ['UX', 2245, 3, 5.3, 83088, 'F'],
       ['Development', 2365, 1, 5.9, 81363, 'M'],
       ['Development', 1500, 3, 6.0, 93940, 'F'],
       ['Testing', 1456, 2, 6.8, 91738, 'M'],
       ['Testing', 1760, 1, 7.1, 98273, 'F'],
       ['UX', 2400, 4, 7.9, 101302, 'F'],
       ['Development', 2148, 3, 8.2, 113812, 'M'],
       ['UX', 1450, 2, 8.7, 109431, 'F'],
       ['UX', 1000, 4, 9.0, 105582, 'M'],
       ['Testing', 1540, 3, 9.5, 116969, 'F'],
       ['Development', 1500, 2, 9.6, 112635, 'F'],
       ['Testing', 3000, 4, 10.3, 122391, 'M'],
       ['UX', 2100, 3, 10.5, 121872, 'F']], dtype=object)
```

12.How do I merge two Pandas' DataFrames on a specific column?

In [ ]:
```
df1=pd.read_csv(r"file path")
df2=pd.read_csv(r"file path")
merged_df = pd.merge(df1, df2, on='specific_column')
merged_df
```

13.How do I pivot a Pandas DataFrame?

In [35]:
```
pivot_df = df.pivot(index='Certification', columns='Salary')
pivot_df
```

Out[35]:

Salary	37731	39343	39891	43525	46205	54445	55794	56
Certification								
0	NaN	Development	NaN	NaN	NaN	NaN	NaN	N
1	NaN	NaN	NaN	UX	Testing	Testing	NaN	
2	Development	NaN	Testing	NaN	NaN	NaN	Development	N
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

5 rows × 120 columns

14.How do I read a CSV file into Pandas DataFrame?

In [36]:
```
df=pd.read_csv(r"C:\Users\KIIT\Downloads\Salary_Classification.csv")
df
```

Out[36]:
```
      Department  WorkedHours  Certification  YearExperience  Salary
 0   Development         2300              0             1.1   39343
 1       Testing         2100              1             1.3   46205
 2   Development         2104              2             1.5   37731
 3            UX         1200              1             2.0   43525
 4       Testing         1254              2             2.2   39891
 5            UX         1236              1             2.9   56642
 6   Development         1452              2             3.0   60150
 7       Testing         1789              1             3.2   54445
 8            UX         1645              1             3.2   64445
 9            UX         1258              0             3.7   57189
10       Testing         1478              3             3.9   63218
11   Development         1257              2             4.0   55794
12   Development         1596              1             4.0   56957
13       Testing         1256              2             4.1   57081
14            UX         1489              3             4.5   61111
15   Development         1236              3             4.9   67938
16       Testing         2311              2             5.1   66029
17            UX         2245              3             5.3   83088
18   Development         2365              1             5.9   81363
19   Development         1500              3             6.0   93940
20       Testing         1456              2             6.8   91738
21       Testing         1760              1             7.1   98273
22            UX         2400              4             7.9  101302
23   Development         2148              3             8.2  113812
24            UX         1450              2             8.7  109431
25            UX         1000              4             9.0  105582
26       Testing         1540              3             9.5  116969
27   Development         1500              2             9.6  112635
28       Testing         3000              4            10.3  122391
29            UX         2100              3            10.5  121872
```

15. How do I write a Pandas DataFrame to a CSV file?

In [ ]:
```
df.to_csv(r"C:\Users\KIIT\Desktop\TTL\pandas_dataframe.csv", index=Fals
```

16. How do I convert a Pandas DataFrame to a JSON file?

In [ ]:
```
df.to_json(r"C:\Users\KIIT\Desktop\TTL\pandas_dataframe.json", orient='
```

17. How do I apply a function to a column in a Pandas DataFrame?

In [40]:
```
def func(x):
    mean=df['Salary'].mean()
    print(mean)

df['Mean'] = df['Salary'].apply(func)
```

```
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
76003.0
```

18. How do I drop a column from a Pandas DataFrame?

In [49]:
```
df.drop(columns=['Mean'], inplace=True)
df
```

Out[49]:

	Department	WorkedHours	YearExperience	Salary
0	Development	2300	1.1	39343.0
1	Testing	2100	1.3	46205.0
2	Development	2104	1.5	37731.0
3	UX	1200	2.0	43525.0
4	Testing	1254	2.2	39891.0
5	UX	1236	2.9	56642.0
6	Development	1452	3.0	60150.0
7	Testing	1789	3.2	54445.0
8	UX	1645	3.2	64445.0
9	UX	1258	3.7	57189.0
10	Testing	1478	3.9	63218.0
11	Development	1257	4.0	55794.0
12	Development	1596	4.0	56957.0
13	Testing	1256	4.1	57081.0
14	UX	1489	4.5	61111.0
15	Development	1236	4.9	67938.0
16	Testing	2311	5.1	66029.0
17	UX	2245	5.3	83088.0
18	Development	2365	5.9	81363.0
19	Development	1500	6.0	93940.0
20	Testing	1456	6.8	91738.0
21	Testing	1760	7.1	98273.0
22	UX	2400	7.9	101302.0
23	Development	2148	8.2	113812.0
24	UX	1450	8.7	109431.0
25	UX	1000	9.0	105582.0
26	Testing	1540	9.5	116969.0
27	Development	1500	9.6	112635.0
28	Testing	3000	10.3	122391.0
29	UX	2100	10.5	121872.0

19. How do I change the datatype of a column in a Pandas DataFrame?

In [47]:
```
df['Salary'] = df['Salary'].astype(float)
df
```

Out[47]:

	Department	WorkedHours	YearExperience	Salary	Mean
0	Development	2300	1.1	39343.0	None
1	Testing	2100	1.3	46205.0	None
2	Development	2104	1.5	37731.0	None
3	UX	1200	2.0	43525.0	None
4	Testing	1254	2.2	39891.0	None
5	UX	1236	2.9	56642.0	None
6	Development	1452	3.0	60150.0	None
7	Testing	1789	3.2	54445.0	None
8	UX	1645	3.2	64445.0	None
9	UX	1258	3.7	57189.0	None
10	Testing	1478	3.9	63218.0	None
11	Development	1257	4.0	55794.0	None
12	Development	1596	4.0	56957.0	None
13	Testing	1256	4.1	57081.0	None
14	UX	1489	4.5	61111.0	None
15	Development	1236	4.9	67938.0	None
16	Testing	2311	5.1	66029.0	None
17	UX	2245	5.3	83088.0	None
18	Development	2365	5.9	81363.0	None
19	Development	1500	6.0	93940.0	None
20	Testing	1456	6.8	91738.0	None
21	Testing	1760	7.1	98273.0	None
22	UX	2400	7.9	101302.0	None
23	Development	2148	8.2	113812.0	None
24	UX	1450	8.7	109431.0	None
25	UX	1000	9.0	105582.0	None
26	Testing	1540	9.5	116969.0	None
27	Development	1500	9.6	112635.0	None
28	Testing	3000	10.3	122391.0	None
29	UX	2100	10.5	121872.0	None

20. How do I find the unique values in a column of a Pandas DataFrame?

In [52]:
```
unique_values = df['Department'].unique()
unique_values
```

Out[52]:
```
array(['Development', 'Testing', 'UX'], dtype=object)
```

21. How do I select the first n rows from a Pandas DataFrame?

In [54]:
```
n = 5
first_n_rows = df.head(n)
first_n_rows
```

Out[54]:

	Department	WorkedHours	YearExperience	Salary
0	Development	2300	1.1	39343.0
1	Testing	2100	1.3	46205.0
2	Development	2104	1.5	37731.0
3	UX	1200	2.0	43525.0
4	Testing	1254	2.2	39891.0

22. How do I reset the index of a Pandas DataFrame?

In [55]:
```python
reset_index_df = df.reset_index(drop=True)
reset_index_df
```

Out[55]:

	Department	WorkedHours	YearExperience	Salary
0	Development	2300	1.1	39343.0
1	Testing	2100	1.3	46205.0
2	Development	2104	1.5	37731.0
3	UX	1200	2.0	43525.0
4	Testing	1254	2.2	39891.0
5	UX	1236	2.9	56642.0
6	Development	1452	3.0	60150.0
7	Testing	1789	3.2	54445.0
8	UX	1645	3.2	64445.0
9	UX	1258	3.7	57189.0
10	Testing	1478	3.9	63218.0
11	Development	1257	4.0	55794.0
12	Development	1596	4.0	56957.0
13	Testing	1256	4.1	57081.0
14	UX	1489	4.5	61111.0
15	Development	1236	4.9	67938.0
16	Testing	2311	5.1	66029.0
17	UX	2245	5.3	83088.0
18	Development	2365	5.9	81363.0
19	Development	1500	6.0	93940.0
20	Testing	1456	6.8	91738.0
21	Testing	1760	7.1	98273.0
22	UX	2400	7.9	101302.0
23	Development	2148	8.2	113812.0
24	UX	1450	8.7	109431.0
25	UX	1000	9.0	105582.0
26	Testing	1540	9.5	116969.0
27	Development	1500	9.6	112635.0
28	Testing	3000	10.3	122391.0
29	UX	2100	10.5	121872.0

23. How do I slice a Pandas DataFrame by row?

In [56]:
```
sliced_df = df.iloc[2:5]
sliced_df
```

Out[56]:

	Department	WorkedHours	YearExperience	Salary
2	Development	2104	1.5	37731.0
3	UX	1200	2.0	43525.0
4	Testing	1254	2.2	39891.0

24.How do I select the last n rows from a Pandas DataFrame?

In [57]:
```
last_n_rows = df.tail(n)
last_n_rows
```

Out[57]:

	Department	WorkedHours	YearExperience	Salary
25	UX	1000	9.0	105582.0
26	Testing	1540	9.5	116969.0
27	Development	1500	9.6	112635.0
28	Testing	3000	10.3	122391.0
29	UX	2100	10.5	121872.0

25. How do I select a random sample of rows from a Pandas DataFrame?

In [58]:
```
random_sample = df.sample(n=3)
random_sample
```

Out[58]:

	Department	WorkedHours	YearExperience	Salary
29	UX	2100	10.5	121872.0
7	Testing	1789	3.2	54445.0
21	Testing	1760	7.1	98273.0

26. How do I create a pivot table in Pandas?

In [60]:
```
pivot_table = pd.pivot_table(df, index='Department', values='Salary', a
pivot_table
```

Out[60]:

	Salary
Department	
Development	71966.3
Testing	75624.0
UX	80418.7

27. How do I concatenate two Pandas DataFrames?

In [62]:
```
df1 = df.head(3)
df2 = df.tail(3)
concatenated_df = pd.concat([df1, df2])
concatenated_df
```

Out[62]:

	Department	WorkedHours	YearExperience	Salary
0	Development	2300	1.1	39343.0
1	Testing	2100	1.3	46205.0
2	Development	2104	1.5	37731.0
27	Development	1500	9.6	112635.0
28	Testing	3000	10.3	122391.0
29	UX	2100	10.5	121872.0

28. How do I create a copy of a Pandas DataFrame?

In [63]:
```
copied_df = df.copy()
copied_df
```

Out[63]:

	Department	WorkedHours	YearExperience	Salary
0	Development	2300	1.1	39343.0
1	Testing	2100	1.3	46205.0
2	Development	2104	1.5	37731.0
3	UX	1200	2.0	43525.0
4	Testing	1254	2.2	39891.0
5	UX	1236	2.9	56642.0
6	Development	1452	3.0	60150.0
7	Testing	1789	3.2	54445.0
8	UX	1645	3.2	64445.0
9	UX	1258	3.7	57189.0
10	Testing	1478	3.9	63218.0
11	Development	1257	4.0	55794.0
12	Development	1596	4.0	56957.0
13	Testing	1256	4.1	57081.0
14	UX	1489	4.5	61111.0
15	Development	1236	4.9	67938.0
16	Testing	2311	5.1	66029.0
17	UX	2245	5.3	83088.0
18	Development	2365	5.9	81363.0
19	Development	1500	6.0	93940.0
20	Testing	1456	6.8	91738.0
21	Testing	1760	7.1	98273.0
22	UX	2400	7.9	101302.0
23	Development	2148	8.2	113812.0
24	UX	1450	8.7	109431.0
25	UX	1000	9.0	105582.0
26	Testing	1540	9.5	116969.0
27	Development	1500	9.6	112635.0
28	Testing	3000	10.3	122391.0
29	UX	2100	10.5	121872.0

29. 29.How do I drop rows with NaN values in a Pandas DataFrame?

In [64]:
```
df_without_nan = df.dropna()
df_without_nan
```

Out[64]:

	Department	WorkedHours	YearExperience	Salary
0	Development	2300	1.1	39343.0
1	Testing	2100	1.3	46205.0
2	Development	2104	1.5	37731.0
3	UX	1200	2.0	43525.0
4	Testing	1254	2.2	39891.0
5	UX	1236	2.9	56642.0
6	Development	1452	3.0	60150.0
7	Testing	1789	3.2	54445.0
8	UX	1645	3.2	64445.0
9	UX	1258	3.7	57189.0
10	Testing	1478	3.9	63218.0
11	Development	1257	4.0	55794.0
12	Development	1596	4.0	56957.0
13	Testing	1256	4.1	57081.0
14	UX	1489	4.5	61111.0
15	Development	1236	4.9	67938.0
16	Testing	2311	5.1	66029.0
17	UX	2245	5.3	83088.0
18	Development	2365	5.9	81363.0
19	Development	1500	6.0	93940.0
20	Testing	1456	6.8	91738.0
21	Testing	1760	7.1	98273.0
22	UX	2400	7.9	101302.0
23	Development	2148	8.2	113812.0
24	UX	1450	8.7	109431.0
25	UX	1000	9.0	105582.0
26	Testing	1540	9.5	116969.0
27	Development	1500	9.6	112635.0
28	Testing	3000	10.3	122391.0
29	UX	2100	10.5	121872.0

30. How do I fill in missing values in a Pandas DataFrame?

In [65]:
```
filled_df = df.fillna(value=0)
filled_df
```

Out[65]:

	Department	WorkedHours	YearExperience	Salary
0	Development	2300	1.1	39343.0
1	Testing	2100	1.3	46205.0
2	Development	2104	1.5	37731.0
3	UX	1200	2.0	43525.0
4	Testing	1254	2.2	39891.0
5	UX	1236	2.9	56642.0
6	Development	1452	3.0	60150.0
7	Testing	1789	3.2	54445.0
8	UX	1645	3.2	64445.0
9	UX	1258	3.7	57189.0
10	Testing	1478	3.9	63218.0
11	Development	1257	4.0	55794.0
12	Development	1596	4.0	56957.0
13	Testing	1256	4.1	57081.0
14	UX	1489	4.5	61111.0
15	Development	1236	4.9	67938.0
16	Testing	2311	5.1	66029.0
17	UX	2245	5.3	83088.0
18	Development	2365	5.9	81363.0
19	Development	1500	6.0	93940.0
20	Testing	1456	6.8	91738.0
21	Testing	1760	7.1	98273.0
22	UX	2400	7.9	101302.0
23	Development	2148	8.2	113812.0
24	UX	1450	8.7	109431.0
25	UX	1000	9.0	105582.0
26	Testing	1540	9.5	116969.0
27	Development	1500	9.6	112635.0
28	Testing	3000	10.3	122391.0
29	UX	2100	10.5	121872.0

31. How do I perform a shift operation on a column in a Pandas DataFrame?

In [66]:
```
df['ShiftedColumn'] = df['Salary'].shift(1)
df
```

Out[66]:

	Department	WorkedHours	YearExperience	Salary	ShiftedColumn
0	Development	2300	1.1	39343.0	NaN
1	Testing	2100	1.3	46205.0	39343.0
2	Development	2104	1.5	37731.0	46205.0
3	UX	1200	2.0	43525.0	37731.0
4	Testing	1254	2.2	39891.0	43525.0
5	UX	1236	2.9	56642.0	39891.0
6	Development	1452	3.0	60150.0	56642.0
7	Testing	1789	3.2	54445.0	60150.0
8	UX	1645	3.2	64445.0	54445.0
9	UX	1258	3.7	57189.0	64445.0
10	Testing	1478	3.9	63218.0	57189.0
11	Development	1257	4.0	55794.0	63218.0
12	Development	1596	4.0	56957.0	55794.0
13	Testing	1256	4.1	57081.0	56957.0
14	UX	1489	4.5	61111.0	57081.0
15	Development	1236	4.9	67938.0	61111.0
16	Testing	2311	5.1	66029.0	67938.0
17	UX	2245	5.3	83088.0	66029.0
18	Development	2365	5.9	81363.0	83088.0
19	Development	1500	6.0	93940.0	81363.0
20	Testing	1456	6.8	91738.0	93940.0
21	Testing	1760	7.1	98273.0	91738.0
22	UX	2400	7.9	101302.0	98273.0
23	Development	2148	8.2	113812.0	101302.0
24	UX	1450	8.7	109431.0	113812.0
25	UX	1000	9.0	105582.0	109431.0
26	Testing	1540	9.5	116969.0	105582.0
27	Development	1500	9.6	112635.0	116969.0
28	Testing	3000	10.3	122391.0	112635.0
29	UX	2100	10.5	121872.0	122391.0

32. How do I perform a cumulative product operation on a column in a Pandas DataFrame?

In [67]:
```
df['CumulativeProduct'] = df['Salary'].cumprod()
df
```

Out[67]:
```
     Department  WorkedHours  YearExperience    Salary  ShiftedColumn  CumulativePr
 0  Development         2300             1.1   39343.0            NaN       3.93430
 1      Testing         2100             1.3   46205.0        39343.0       1.81784
 2  Development         2104             1.5   37731.0        46205.0       6.85890
 3           UX         1200             2.0   43525.0        37731.0       2.98533
 4      Testing         1254             2.2   39891.0        43525.0       1.19088
 5           UX         1236             2.9   56642.0        39891.0       6.74539
 6  Development         1452             3.0   60150.0        56642.0       4.05735
 7      Testing         1789             3.2   54445.0        60150.0       2.20902
 8           UX         1645             3.2   64445.0        54445.0       1.42360
 9           UX         1258             3.7   57189.0        64445.0       8.14146
10      Testing         1478             3.9   63218.0        57189.0       5.14687
11  Development         1257             4.0   55794.0        63218.0       2.87164
12  Development         1596             4.0   56957.0        55794.0       1.63560
13      Testing         1256             4.1   57081.0        56957.0       9.33618
14           UX         1489             4.5   61111.0        57081.0       5.70543
15  Development         1236             4.9   67938.0        61111.0       3.87615
16      Testing         2311             5.1   66029.0        67938.0       2.55938
17           UX         2245             5.3   83088.0        66029.0       2.12654
18  Development         2365             5.9   81363.0        83088.0       1.73022
19  Development         1500             6.0   93940.0        81363.0       1.62536
20      Testing         1456             6.8   91738.0        93940.0      1.491081
21      Testing         1760             7.1   98273.0        91738.0      1.465330
22           UX         2400             7.9  101302.0        98273.0      1.484409
23  Development         2148             8.2  113812.0       101302.0      1.689435
24           UX         1450             8.7  109431.0       113812.0      1.848766
25           UX         1000             9.0  105582.0       109431.0      1.951964
26      Testing         1540             9.5  116969.0       105582.0      2.283193
27  Development         1500             9.6  112635.0       116969.0      2.571674
28      Testing         3000            10.3  122391.0       112635.0      3.147498
29           UX         2100            10.5  121872.0       122391.0      3.835919
```

33. How do I perform a cumulative maximum operation on a column in a Pandas DataFrame?

In [68]:
```python
df['CumulativeMax'] = df['Salary'].cummax()
df
```

Out[68]:

	Department	WorkedHours	YearExperience	Salary	ShiftedColumn	CumulativePr
0	Development	2300	1.1	39343.0	NaN	3.934300
1	Testing	2100	1.3	46205.0	39343.0	1.81784
2	Development	2104	1.5	37731.0	46205.0	6.85890
3	UX	1200	2.0	43525.0	37731.0	2.98533
4	Testing	1254	2.2	39891.0	43525.0	1.19088
5	UX	1236	2.9	56642.0	39891.0	6.74539
6	Development	1452	3.0	60150.0	56642.0	4.05735
7	Testing	1789	3.2	54445.0	60150.0	2.20902
8	UX	1645	3.2	64445.0	54445.0	1.42360
9	UX	1258	3.7	57189.0	64445.0	8.14146
10	Testing	1478	3.9	63218.0	57189.0	5.14687
11	Development	1257	4.0	55794.0	63218.0	2.87164
12	Development	1596	4.0	56957.0	55794.0	1.63560
13	Testing	1256	4.1	57081.0	56957.0	9.33618
14	UX	1489	4.5	61111.0	57081.0	5.70543
15	Development	1236	4.9	67938.0	61111.0	3.87615
16	Testing	2311	5.1	66029.0	67938.0	2.55938
17	UX	2245	5.3	83088.0	66029.0	2.12654
18	Development	2365	5.9	81363.0	83088.0	1.73022
19	Development	1500	6.0	93940.0	81363.0	1.62536
20	Testing	1456	6.8	91738.0	93940.0	1.491081
21	Testing	1760	7.1	98273.0	91738.0	1.465330
22	UX	2400	7.9	101302.0	98273.0	1.484409
23	Development	2148	8.2	113812.0	101302.0	1.689435
24	UX	1450	8.7	109431.0	113812.0	1.848766
25	UX	1000	9.0	105582.0	109431.0	1.951964
26	Testing	1540	9.5	116969.0	105582.0	2.283193
27	Development	1500	9.6	112635.0	116969.0	2.571674
28	Testing	3000	10.3	122391.0	112635.0	3.147498
29	UX	2100	10.5	121872.0	122391.0	3.835919

34. How do I calculate the moving average of a column in a Pandas DataFrame?

In [69]:
```
window_size = 3
df['MovingAverage'] = df['Salary'].rolling(window=window_size).mean()
df
```

Out[69]:

	Department	WorkedHours	YearExperience	Salary	ShiftedColumn	CumulativePr
0	Development	2300	1.1	39343.0	NaN	3.93430
1	Testing	2100	1.3	46205.0	39343.0	1.81784
2	Development	2104	1.5	37731.0	46205.0	6.85890
3	UX	1200	2.0	43525.0	37731.0	2.98533
4	Testing	1254	2.2	39891.0	43525.0	1.19088
5	UX	1236	2.9	56642.0	39891.0	6.74539
6	Development	1452	3.0	60150.0	56642.0	4.05735
7	Testing	1789	3.2	54445.0	60150.0	2.20902
8	UX	1645	3.2	64445.0	54445.0	1.42360
9	UX	1258	3.7	57189.0	64445.0	8.14146
10	Testing	1478	3.9	63218.0	57189.0	5.14687
11	Development	1257	4.0	55794.0	63218.0	2.87164
12	Development	1596	4.0	56957.0	55794.0	1.63560
13	Testing	1256	4.1	57081.0	56957.0	9.33618
14	UX	1489	4.5	61111.0	57081.0	5.70543
15	Development	1236	4.9	67938.0	61111.0	3.87615
16	Testing	2311	5.1	66029.0	67938.0	2.55938
17	UX	2245	5.3	83088.0	66029.0	2.12654
18	Development	2365	5.9	81363.0	83088.0	1.73022
19	Development	1500	6.0	93940.0	81363.0	1.62536
20	Testing	1456	6.8	91738.0	93940.0	1.491081
21	Testing	1760	7.1	98273.0	91738.0	1.465330
22	UX	2400	7.9	101302.0	98273.0	1.484409
23	Development	2148	8.2	113812.0	101302.0	1.689435
24	UX	1450	8.7	109431.0	113812.0	1.848766
25	UX	1000	9.0	105582.0	109431.0	1.951964
26	Testing	1540	9.5	116969.0	105582.0	2.283193
27	Development	1500	9.6	112635.0	116969.0	2.571674
28	Testing	3000	10.3	122391.0	112635.0	3.147498
29	UX	2100	10.5	121872.0	122391.0	3.835919

35. How do I calculate the exponential moving average of a column in a Pandas DataFrame?

In [70]:
```
span = 3
df['ExponentialMovingAverage'] = df['Salary'].ewm(span=span, adjust=Fal
df
```

Out[70]:

	Department	WorkedHours	YearExperience	Salary	ShiftedColumn	CumulativePr
0	Development	2300	1.1	39343.0	NaN	3.93430
1	Testing	2100	1.3	46205.0	39343.0	1.81784
2	Development	2104	1.5	37731.0	46205.0	6.85890
3	UX	1200	2.0	43525.0	37731.0	2.98533
4	Testing	1254	2.2	39891.0	43525.0	1.19088
5	UX	1236	2.9	56642.0	39891.0	6.74539
6	Development	1452	3.0	60150.0	56642.0	4.05735
7	Testing	1789	3.2	54445.0	60150.0	2.20902
8	UX	1645	3.2	64445.0	54445.0	1.42360
9	UX	1258	3.7	57189.0	64445.0	8.14146
10	Testing	1478	3.9	63218.0	57189.0	5.14687
11	Development	1257	4.0	55794.0	63218.0	2.87164
12	Development	1596	4.0	56957.0	55794.0	1.63560
13	Testing	1256	4.1	57081.0	56957.0	9.33618
14	UX	1489	4.5	61111.0	57081.0	5.70543
15	Development	1236	4.9	67938.0	61111.0	3.87615
16	Testing	2311	5.1	66029.0	67938.0	2.55938
17	UX	2245	5.3	83088.0	66029.0	2.12654
18	Development	2365	5.9	81363.0	83088.0	1.73022
19	Development	1500	6.0	93940.0	81363.0	1.62536
20	Testing	1456	6.8	91738.0	93940.0	1.491081
21	Testing	1760	7.1	98273.0	91738.0	1.465330
22	UX	2400	7.9	101302.0	98273.0	1.484409
23	Development	2148	8.2	113812.0	101302.0	1.689435
24	UX	1450	8.7	109431.0	113812.0	1.848766
25	UX	1000	9.0	105582.0	109431.0	1.951964
26	Testing	1540	9.5	116969.0	105582.0	2.283193
27	Development	1500	9.6	112635.0	116969.0	2.571674
28	Testing	3000	10.3	122391.0	112635.0	3.147498
29	UX	2100	10.5	121872.0	122391.0	3.835919

36. How do I calculate the cumulative minimum operation on a column in a Pandas DataFrame?

In [71]:
```
df['CumulativeMin'] = df['Salary'].cummin()
df
```

Out[71]:

	Department	WorkedHours	YearExperience	Salary	ShiftedColumn	CumulativePr
0	Development	2300	1.1	39343.0	NaN	3.93430
1	Testing	2100	1.3	46205.0	39343.0	1.81784
2	Development	2104	1.5	37731.0	46205.0	6.85890
3	UX	1200	2.0	43525.0	37731.0	2.98533
4	Testing	1254	2.2	39891.0	43525.0	1.19088
5	UX	1236	2.9	56642.0	39891.0	6.74539
6	Development	1452	3.0	60150.0	56642.0	4.05735
7	Testing	1789	3.2	54445.0	60150.0	2.20902
8	UX	1645	3.2	64445.0	54445.0	1.42360
9	UX	1258	3.7	57189.0	64445.0	8.14146
10	Testing	1478	3.9	63218.0	57189.0	5.14687
11	Development	1257	4.0	55794.0	63218.0	2.87164
12	Development	1596	4.0	56957.0	55794.0	1.63560
13	Testing	1256	4.1	57081.0	56957.0	9.33618
14	UX	1489	4.5	61111.0	57081.0	5.70543
15	Development	1236	4.9	67938.0	61111.0	3.87615
16	Testing	2311	5.1	66029.0	67938.0	2.55938
17	UX	2245	5.3	83088.0	66029.0	2.12654
18	Development	2365	5.9	81363.0	83088.0	1.73022
19	Development	1500	6.0	93940.0	81363.0	1.62536
20	Testing	1456	6.8	91738.0	93940.0	1.491081
21	Testing	1760	7.1	98273.0	91738.0	1.465330
22	UX	2400	7.9	101302.0	98273.0	1.484409
23	Development	2148	8.2	113812.0	101302.0	1.689435
24	UX	1450	8.7	109431.0	113812.0	1.848766
25	UX	1000	9.0	105582.0	109431.0	1.951964
26	Testing	1540	9.5	116969.0	105582.0	2.283193
27	Development	1500	9.6	112635.0	116969.0	2.571674
28	Testing	3000	10.3	122391.0	112635.0	3.147498
29	UX	2100	10.5	121872.0	122391.0	3.835919

In [ ]:
```

```