In [19]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import heapq
```

In [31]:
```
excelfilepath = 'data preprocessing/Employees.xlsx'
df = pd.read_excel( excelfilepath ,'Employees')
df_new = pd.read_excel(excelfilepath ,'NewEmployees')
```

In [33]:
```
df
```

Out[33]:

	EmpID	Salary	Role
0	1	2000.0	Senior
1	1	2000.0	Senior
2	3	NaN	Senior
3	4	2900.0	Senior
4	5	3200.0	Senior
5	6	3500.0	Senior
6	7	3800.0	Senior
7	8	4100.0	Senior
8	9	4400.0	Senior
9	10	4700.0	TL
10	11	5000.0	TL
11	12	5300.0	TL
12	13	5600.0	TL
13	14	5900.0	TL
14	15	6200.0	TL
15	16	6500.0	TL
16	17	6800.0	TL
17	18	7100.0	TL
18	19	7400.0	TL
19	20	7700.0	TL
20	21	8000.0	Manager
21	22	8300.0	Manager
22	23	8600.0	Manager
23	24	8900.0	Manager
24	25	9200.0	Manager
25	26	9500.0	Manager

	EmpID	Salary	Role
26	27	9800.0	Manager
27	28	10100.0	Manager
28	29	10400.0	Manager
29	30	20000.0	SMO
30	31	23000.0	SMO
31	32	26000.0	SMO
32	33	29000.0	SMO
33	34	32000.0	SMO
34	35	35000.0	SMO
35	36	38000.0	SMO
36	37	41000.0	SMO
37	38	44000.0	SMO

In [32]:
```
df_new
```

Out[32]:

	EmpID	Salary	Role
0	201	2000	Senior
1	202	2300	Senior
2	203	2600	Senior
3	204	2900	Senior
4	205	3200	Senior
5	206	3500	Senior
6	207	3800	Senior
7	208	4100	Senior
8	209	4400	Senior
9	210	4700	TL
10	211	5000	TL
11	212	5300	TL
12	213	5600	TL
13	214	5900	TL
14	215	6200	TL
15	216	6500	TL
16	217	6800	TL
17	218	7100	TL

	EmpID	Salary	Role
18	219	7400	TL
19	220	7700	TL
20	221	8000	Manager
21	222	8300	Manager
22	223	8600	Manager
23	224	8900	Manager
24	225	9200	Manager
25	226	9500	Manager
26	227	9800	Manager
27	228	10100	Manager
28	229	10400	Manager
29	230	20000	SMO
30	231	23000	SMO
31	232	26000	SMO
32	233	29000	SMO
33	234	32000	SMO
34	235	35000	SMO
35	236	38000	SMO
36	237	41000	SMO
37	238	44000	SMO

In [34]:
```
df.describe()
```

Out[34]:

	EmpID	Salary
count	38.000000	37.000000
mean	19.473684	12564.864865
std	11.156714	12023.130860
min	1.000000	2000.000000
25%	10.250000	5000.000000
50%	19.500000	7700.000000
75%	28.750000	10400.000000
max	38.000000	44000.000000

In [71]:
```
# to display the nullable values and try to solve this problem
df[df.Salary.isnull()]
```

```
# if i want to fill the missing values we use :
df.Salary.fillna(df.Salary.mean(), inplace = True)
```

```
C:\Users\Lenovo\anaconda3\lib\site-packages\pandas\core\series.py:4517: SettingWithCopyW
arning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  return super().fillna(
```

In [42]:
```
# to display the dublicated values and try to remove it
df[df.EmpID.duplicated()]
# if i want to drop the dublicates values we use :
df = df.drop_duplicates(keep = 'last')
```

In [44]:
```
# Getting the most largest numbers in order to validate with upper salary limits
print(heapq.nlargest(5 , list(df.Salary)))
# Getting the most smallest numbers in order to validate with upper salary limits
print(heapq.nsmallest(5 , list(df.Salary)))
```

```
[44000.0, 41000.0, 38000.0, 35000.0, 32000.0]
[2000.0, 2900.0, 3200.0, 3500.0, 3800.0]
```

In [72]:
```
df
```

Out[72]:

	EmpID	Salary	Role
1	1	2000.000000	Senior
2	3	12564.864865	Senior
3	4	2900.000000	Senior
4	5	3200.000000	Senior
5	6	3500.000000	Senior
6	7	3800.000000	Senior
7	8	4100.000000	Senior
8	9	4400.000000	Senior
9	10	4700.000000	TL
10	11	5000.000000	TL
11	12	5300.000000	TL
12	13	5600.000000	TL
13	14	5900.000000	TL
14	15	6200.000000	TL
15	16	6500.000000	TL
16	17	6800.000000	TL
17	18	7100.000000	TL
18	19	7400.000000	TL
19	20	7700.000000	TL

	EmpID	Salary	Role
20	21	8000.000000	Manager
21	22	8300.000000	Manager
22	23	8600.000000	Manager
23	24	8900.000000	Manager
24	25	9200.000000	Manager
25	26	9500.000000	Manager
26	27	9800.000000	Manager
27	28	10100.000000	Manager
28	29	10400.000000	Manager
29	30	20000.000000	SMO
30	31	23000.000000	SMO
31	32	26000.000000	SMO
32	33	29000.000000	SMO
33	34	32000.000000	SMO
34	35	35000.000000	SMO
35	36	38000.000000	SMO
36	37	41000.000000	SMO
37	38	44000.000000	SMO

In [45]:
```
 # validate all employees that below the quarter of average
df[df.Salary < df.Salary.mean() * .25]
```

Out[45]:

	EmpID	Salary	Role
1	1	2000.0	Senior
3	4	2900.0	Senior
4	5	3200.0	Senior

In [46]:
```
frames = [df , df_new]
all_records = pd.concat(frames)
all_records
```

Out[46]:

	EmpID	Salary	Role
1	1	2000.000000	Senior
2	3	12564.864865	Senior
3	4	2900.000000	Senior
4	5	3200.000000	Senior
5	6	3500.000000	Senior

	EmpID	Salary	Role
...	...	...	...
33	234	32000.000000	SMO
34	235	35000.000000	SMO
35	236	38000.000000	SMO
36	237	41000.000000	SMO
37	238	44000.000000	SMO

75 rows × 3 columns

In [53]:
```
# Adding new columns
all_records['SalaryWithTax'] = all_records['Salary'] * 1.22
all_records
```

Out[53]:

	EmpID	Salary	Role	SalaryWithTax
1	1	2000.000000	Senior	2440.000000
2	3	12564.864865	Senior	15329.135135
3	4	2900.000000	Senior	3538.000000
4	5	3200.000000	Senior	3904.000000
5	6	3500.000000	Senior	4270.000000
...	...	...	...	...
33	234	32000.000000	SMO	39040.000000
34	235	35000.000000	SMO	42700.000000
35	236	38000.000000	SMO	46360.000000
36	237	41000.000000	SMO	50020.000000
37	238	44000.000000	SMO	53680.000000

75 rows × 4 columns

In [59]:
```
# Removing columns
del all_records['SalaryWithTax']
```

In [60]:
```
all_records
```

Out[60]:

	EmpID	Salary	Role
1	1	2000.000000	Senior
2	3	12564.864865	Senior
3	4	2900.000000	Senior
4	5	3200.000000	Senior
5	6	3500.000000	Senior

	EmpID	Salary	Role
...	...	...	...
33	234	32000.000000	SMO
34	235	35000.000000	SMO
35	236	38000.000000	SMO
36	237	41000.000000	SMO
37	238	44000.000000	SMO

75 rows × 3 columns

In [63]:
```
# pivot table with aggregation
pd.pivot_table(all_records , values = ['Salary'] , columns = ['Role'] , aggfunc = 'sum'
```

Out[63]:

Role	Manager	SMO	Senior	TL
Salary	165600.0	576000.0	65264.864865	136400.0

In [70]:
```
# Saving dataframe to external entity

import os
FilePath = os.getcwd() + '\\CotenatedStudents.xlsx'
writer = pd.ExcelWriter(FilePath)
all_records.to_excel(writer , sheet_name = 'ConcatenatedSheet')
writer.save()
FilePath
```

Out[70]:
```
'C:\\Users\\Lenovo\\CotenatedStudents.xlsx'
```