

```
In [5]: 1 import pandas as pd
2 dict={
3     'Ename':['roshan','amar','ashwini','lohith','mohan','pramod'],
4     'category':['regular','adhoc','regular','adhoc','contract','regular'],
5     'department':['cs','cs','ec','mc','cs','ec'],
6     'experience':['10','15','20','14','none','8'],
7     'salary':['50000','15000','30000','15000','10000','40000']
8 }
9 df=pd.DataFrame(dict)
10 print(df)
11
```

	Ename	category	department	experience	salary
0	roshan	regular	cs	10	50000
1	amar	adhoc	cs	15	15000
2	ashwini	regular	ec	20	30000
3	lohith	adhoc	mc	14	15000
4	mohan	contract	cs	none	10000
5	pramod	regular	ec	8	40000

```
In [6]: 1 pivot_table_department = df.pivot_table(values='salary', index='Ename',
2 print(pivot_table_department )
3
```

department	cs	ec	mc
Ename			
amar	15000.0	NaN	NaN
ashwini	NaN	30000.0	NaN
lohith	NaN	NaN	15000.0
mohan	10000.0	NaN	NaN
pramod	NaN	40000.0	NaN
roshan	50000.0	NaN	NaN

```
In [7]: 1 new_data = {
2     'Ename' : 'newemploye',
3     'category' : 'regular',
4     'department' : 'cs',
5     'exproience' : 20,
6     'salary' : 40000
7 }
8
9 df = df.append(new_data, ignore_index=True)
10 print(df)
```

	Ename	category	department	experience	salary	exproience
0	roshan	regular	cs	10	50000	NaN
1	amar	adhoc	cs	15	15000	NaN
2	ashwini	regular	ec	20	30000	NaN
3	lohith	adhoc	mc	14	15000	NaN
4	mohan	contract	cs	none	10000	NaN
5	pramod	regular	ec	8	40000	NaN
6	newemploye	regular	cs	NaN	40000	20.0

C:\Users\Anusha V\AppData\Local\Temp\ipykernel_7628\3758197510.py:9: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

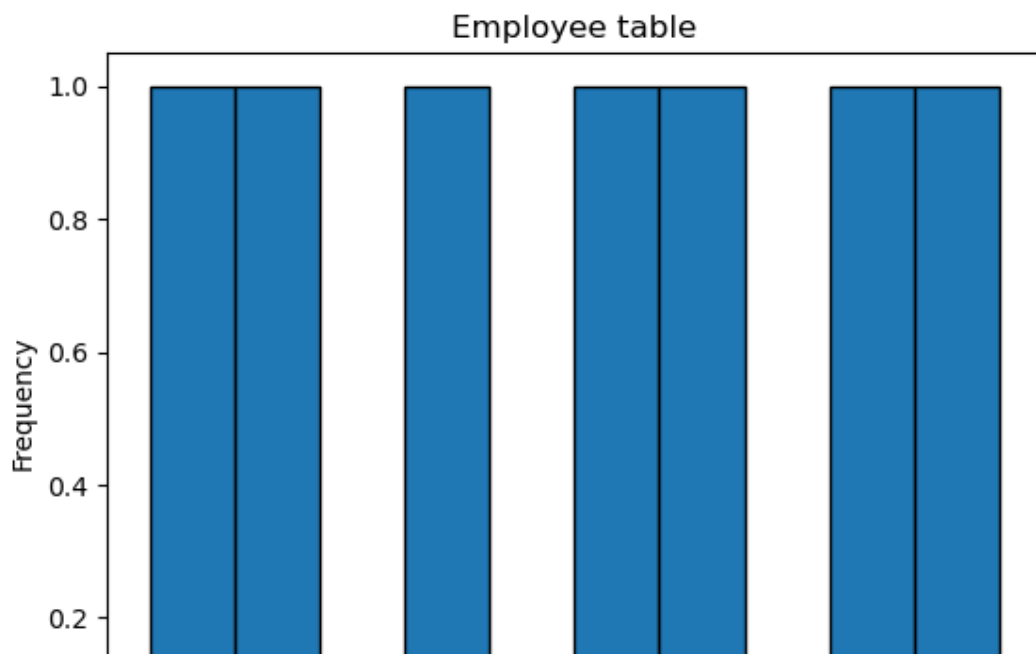
```
df = df.append(new_data, ignore_index=True)
```

```
In [8]: 1 cs_df = df[df['department'] == 'cs']  
2 cs_df
```

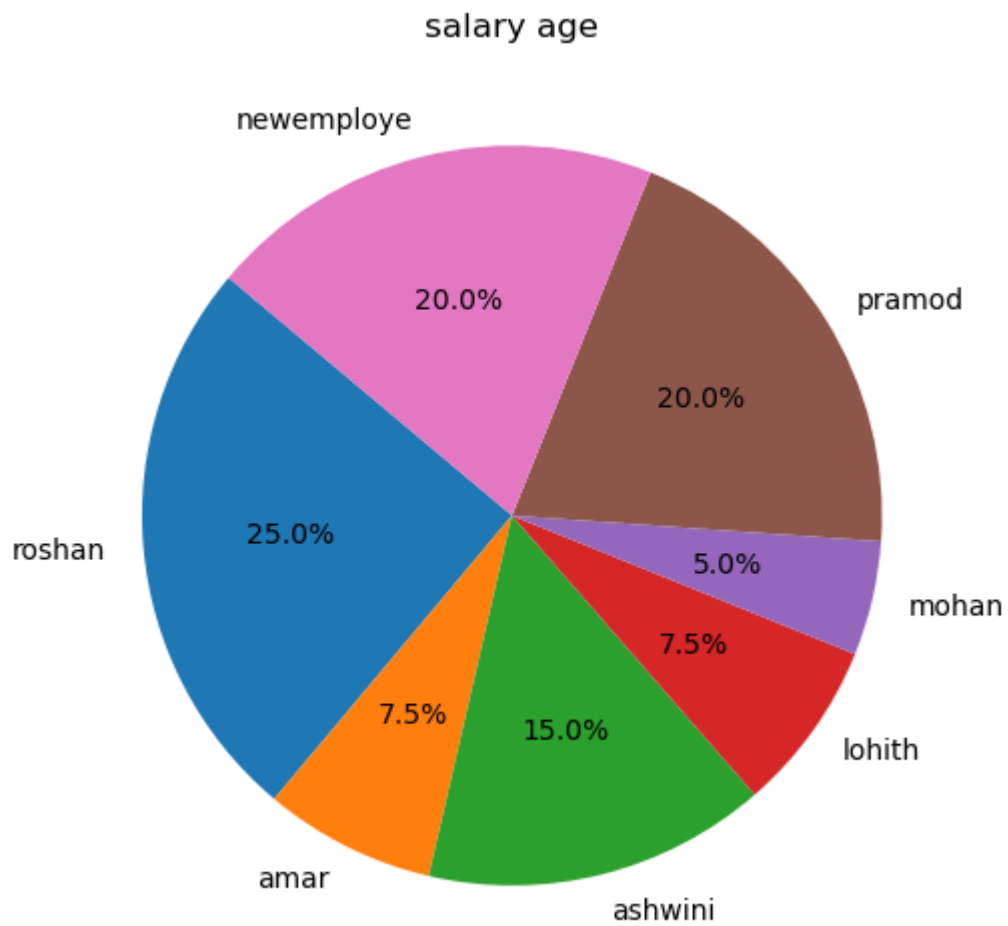
```
Out[8]:
```

	Ename	category	department	experience	salary	expreience
0	roshan	regular	cs	10	50000	NaN
1	amar	adhoc	cs	15	15000	NaN
4	mohan	contract	cs	none	10000	NaN
6	newemploye	regular	cs	NaN	40000	20.0

```
In [9]: 1 import matplotlib.pyplot as plt  
2 plt.hist(df['Ename'], bins=10, edgecolor='k')  
3 plt.xlabel('salary')  
4 plt.ylabel('Frequency')  
5 plt.title('Employee table')  
6 plt.show()  
7
```



```
In [10]: 1 import matplotlib.pyplot as plt
2
3 subject = 'salary' # Change to the subject you want to visualize
4 plt.figure(figsize=(6, 6))
5 plt.pie(df[subject], labels=df['Ename'], autopct='%1.1f%%', startangle=
6 plt.title(f'{subject} age')
7 plt.show()
```



```
In [11]: 1 sort_df = df.sort_values(by='salary',ascending=True)
          2 print(sort_df)
```

```
-----
-
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_7628\3214428358.py in <module>
----> 1 sort_df = df.sort_values(by='salary',ascending=True)
      2 print(sort_df)

~\anaconda3\lib\site-packages\pandas\util\_decorators.py in wrapper(*args,
**kwargs)
    309             stacklevel=stacklevel,
    310         )
--> 311         return func(*args, **kwargs)
    312
    313     return wrapper

~\anaconda3\lib\site-packages\pandas\core\frame.py in sort_values(self, by,
axis, ascending, inplace, kind, na_position, ignore_index, key)
    6331         ascending = ascending[0]
    6332
-> 6333         indexer = nargsort(
    6334             k, kind=kind, ascending=ascending, na_position=na_
position, key=key
    6335         )

~\anaconda3\lib\site-packages\pandas\core\sorting.py in nargsort(items, ki
nd, ascending, na_position, key, mask)
    415         non_nans = non_nans[::-1]
    416         non_nan_idx = non_nan_idx[::-1]
--> 417         indexer = non_nan_idx[non_nans.argsort(kind=kind)]
    418         if not ascending:
    419             indexer = indexer[::-1]

TypeError: '<' not supported between instances of 'int' and 'str'
```

```
In [31]: 1 df.groupby('salary').Ename.max()
```

```
Out[31]: salary
40000    newemploye
10000      mohan
15000    lohith
30000    ashwini
40000    pramod
50000    roshan
Name: Ename, dtype: object
```

In [33]: 1 df.groupby('salary').Ename.max()

Out[33]:

	Ename	category	department	experience	salary	expreience
0	roshan	regular	cs	10	50000	NaN
1	amar	adhoc	cs	15	15000	NaN
4	mohan	contract	cs	none	10000	NaN
6	newemploye	regular	cs	NaN	40000	20.0

In [34]: 1 pivot_table = df.pivot_table(index='Ename', aggfunc='mean')
2 print(pivot_table)
3

	expreience
Ename	
newemploye	20.0

In [35]: 1

```
-----
---
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_8912\802983524.py in <module>
----> 1 plt.plot(df['Ename'], df['salary'], marker='o', linestyle='-', color='b')
      2 plt.xlabel('Student Name')
      3 plt.ylabel('Maths Score')
      4 plt.title('Maths Scores by Student')
      5 plt.xticks(rotation=45)

~\anaconda3\lib\site-packages\matplotlib\pyplot.py in plot(scalex, scaley, data, *args, **kwargs)
    2767 @_copy_docstring_and_deprecators(Axes.plot)
    2768 def plot(*args, scalex=True, scaley=True, data=None, **kwargs):
-> 2769     return gca().plot(
    2770         *args, scalex=scalex, scaley=scaley,
    2771         **({"data": data} if data is not None else {}), **kwargs
```

In [8]: 1 data = [(1,2),(5,6),(3,7),(2,8)]
2 sorted_df = sorted(data, key=lambda x: x[1])
3 sorted_df

	Ename	category	department	experience	salary
0	roshan	regular	cs	10	50000
1	amar	adhoc	cs	15	15000
2	ashwini	regular	ec	20	30000
3	lohith	adhoc	mc	14	15000
4	mohan	contract	cs	none	10000
5	pramod	regular	ec	8	40000

```
In [9]: 1 sort_df = df.sort_values(by='salary',ascending=True)
        2 print(sort_df)
```

	Ename	category	department	experience	salary
4	mohan	contract	cs	none	10000
1	amar	adhoc	cs	15	15000
3	lohith	adhoc	mc	14	15000
2	ashwini	regular	ec	20	30000
5	pramod	regular	ec	8	40000
0	roshan	regular	cs	10	50000

```
In [11]: 1 aggregation_methods = {'salary':['mean','sum','max','min']}
        2 result = df.agg(aggregation_methods)
        3 print(result)
```

	salary
mean	83333583338333359097994280960.0
sum	500001500030000150001000040000
max	50000
min	10000

```
In [12]: 1 df.sort_index(axis=1,ascending=30)
        2
```

```
Out[12]:
```

	Ename	category	department	experience	salary
0	roshan	regular	cs	10	50000
1	amar	adhoc	cs	15	15000
2	ashwini	regular	ec	20	30000
3	lohith	adhoc	mc	14	15000
4	mohan	contract	cs	none	10000
5	pramod	regular	ec	8	40000

```
In [6]: 1 data = [(1,5),(2,3),(3,8),(4,1)]
        2 sorted_df = sorted(data, key=lambda x: x[1])
        3 print(sorted_df)
```

```
[(4, 1), (2, 3), (1, 5), (3, 8)]
```

```
In [4]: 1 data = [(1,2),(5,6),(3,7),(2,8)]
        2 sorted_df = sorted(data, key=lambda x: x[1])
        3 sorted_df
```

```
Out[4]: [(1, 2), (5, 6), (3, 7), (2, 8)]
```

```
In [8]: 1 from functools import reduce
        2 def add(x, y):
        3     return x+y
        4 numbers = (1,2,3,4,5)
        5 result = reduce(add, numbers)
        6 result
```

```
Out[8]: 15
```

```
In [10]: 1 number = (1,2,3,4,5)
2 def square(x):
3     return x ** 2
4 squared_number = map(square, number)
5 squared_number_list = list(squared_number)
6 print(squared_number_list)
```

```
[1, 4, 9, 16, 25]
```

```
In [12]: 1 number = (1,2,3,4,5)
2 def is_even(x):
3     return x % 2 == 0
4 even_number = filter(is_even, number)
5 even_number_list = list(even_number)
6 print(even_number_list)
```

```
[2, 4]
```

```
In [53]: 1 import pandas as pd
2 dict={
3     'Ename':['roshan','amar','ashwini','lohith','mohan','pramod'],
4     'category':['regular','adhoc','regular','adhoc','contract','regular'],
5     'department':['cs','cs','ec','mc','cs','ec'],
6     'experience':['10','15','20','14','none','8'],
7     'salary':['50000','15000','30000','15000','10000','40000']
8 }
9 df=pd.DataFrame(dict)
10 print(df)
11
```

	Ename	category	department	experience	salary
0	roshan	regular	cs	10	50000
1	amar	adhoc	cs	15	15000
2	ashwini	regular	ec	20	30000
3	lohith	adhoc	mc	14	15000
4	mohan	contract	cs	none	10000
5	pramod	regular	ec	8	40000

```
In [54]: 1 pivot_tabel = df.pivot_table(index='salary', aggfunc='mean')
2 print(pivot_table)
```

	salary
Ename	
amar	15000.0
ashwini	30000.0
lohith	15000.0
mohan	10000.0
pramod	40000.0
roshan	50000.0

C:\Users\Anusha V\AppData\Local\Temp\ipykernel_8372\1714949334.py:1: FutureWarning: Dropping invalid columns in DataFrameGroupBy.mean is deprecated. In a future version, a TypeError will be raised. Before calling .mean, select only columns which should be valid for the function.

```
pivot_tabel = df.pivot_table(index='salary', aggfunc='mean')
```

```
In [39]: 1 pivot_table = df.pivot_table(index='Ename', aggfunc='mean')
2         print(pivot_table)
```

```
          salary
Ename
amar      15000.0
ashwini   30000.0
lohith    15000.0
mohan     10000.0
pramod    40000.0
roshan    50000.0
```

C:\Users\Anusha V\AppData\Local\Temp\ipykernel_8372\1519672256.py:1: FutureWarning: Dropping invalid columns in DataFrameGroupBy.mean is deprecated. In a future version, a TypeError will be raised. Before calling .mean, select only columns which should be valid for the function.

```
pivot_table = df.pivot_table(index='Ename', aggfunc='mean')
```

```
In [47]: 1 new_data = {
2         'Ename': 'NewStudent',
3         'category': 'regular',
4         'department': 'CS',
5         'experience': 20,
6         'salary': 15000
7     }
8     df = df.append(new_data, ignore_index=True)
9     df
```

C:\Users\Anusha V\AppData\Local\Temp\ipykernel_8372\1262638907.py:8: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.

```
df = df.append(new_data, ignore_index=True)
```

```
Out[47]:
```

	Ename	category	department	experience	salary
0	roshan	regular	cs	10	50000
1	amar	adhoc	cs	15	15000
2	ashwini	regular	ec	20	30000
3	lohith	adhoc	mc	14	15000
4	mohan	contract	cs	none	10000
5	pramod	regular	ec	8	40000
6	NewStudent	regular	CS	20	15000
7	NewStudent	regular	CS	20	15000


```
In [48]: 1 sorted_df = df[df['experience'] > 10].sort_value(by='experience', ascend
2 sorted_df
```

```
-----
-
TypeError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_8372\3815716362.py in <module>
----> 1 sorted_df = df[df['experience'] > 10].sort_value(by='experience', ascending=False)
      2 sorted_df

~\anaconda3\lib\site-packages\pandas\core\ops\common.py in new_method(self, other)
      68         other = item_from_zerodim(other)
      69
----> 70         return method(self, other)
      71
      72     return new_method

~\anaconda3\lib\site-packages\pandas\core\arraylike.py in __gt__(self, other)
      54     @unpack_zerodim_and_defer("__gt__")
      55     def __gt__(self, other):
----> 56         return self._cmp_method(other, operator.gt)
      57
      58     @unpack_zerodim_and_defer("__ge__")

~\anaconda3\lib\site-packages\pandas\core\series.py in _cmp_method(self, other, op)
      5621
      5622         with np.errstate(all="ignore"):
-> 5623             res_values = ops.comparison_op(lvalues, rvalues, op)
      5624
      5625         return self._construct_result(res_values, name=res_name)

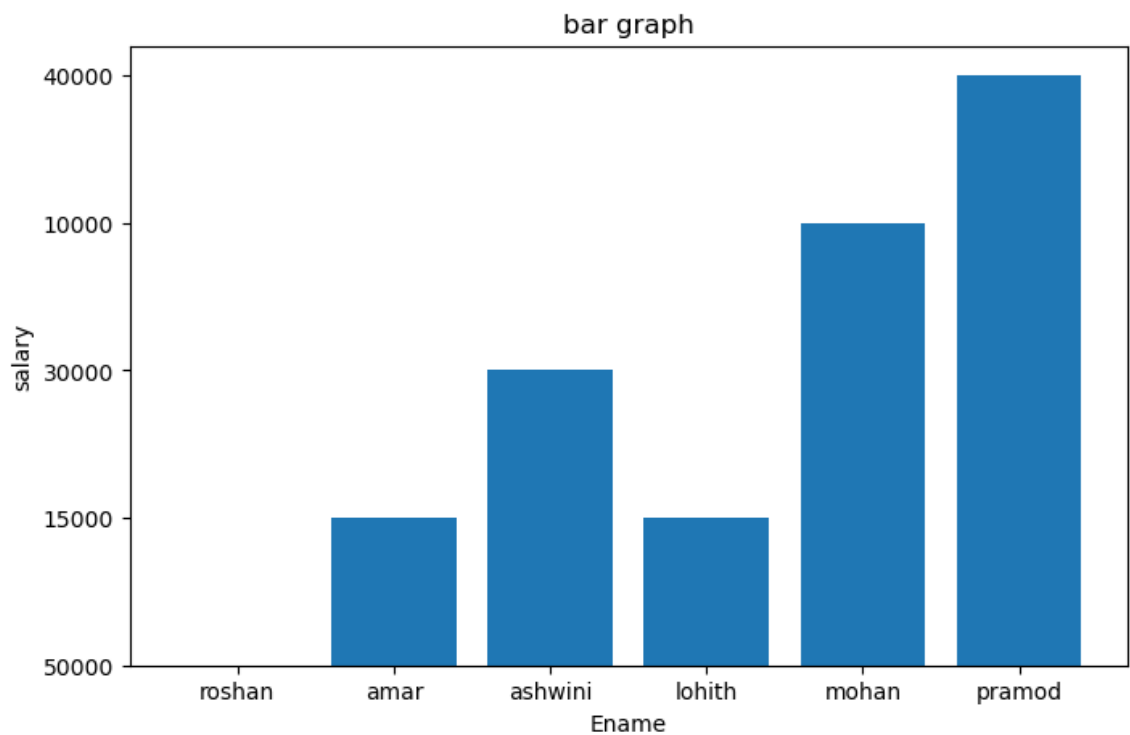
~\anaconda3\lib\site-packages\pandas\core\ops\array_ops.py in comparison_op(left, right, op)
      281
      282     elif is_object_dtype(lvalues.dtype) or isinstance(rvalues, str):
-> 283         res_values = comp_method_OBJECT_ARRAY(op, lvalues, rvalues)
      284
      285     else:

~\anaconda3\lib\site-packages\pandas\core\ops\array_ops.py in comp_method_OBJECT_ARRAY(op, x, y)
      71     result = libops.vec_compare(x.ravel(), y.ravel(), op)
      72     else:
----> 73         result = libops.scalar_compare(x.ravel(), y, op)
      74     return result.reshape(x.shape)
      75

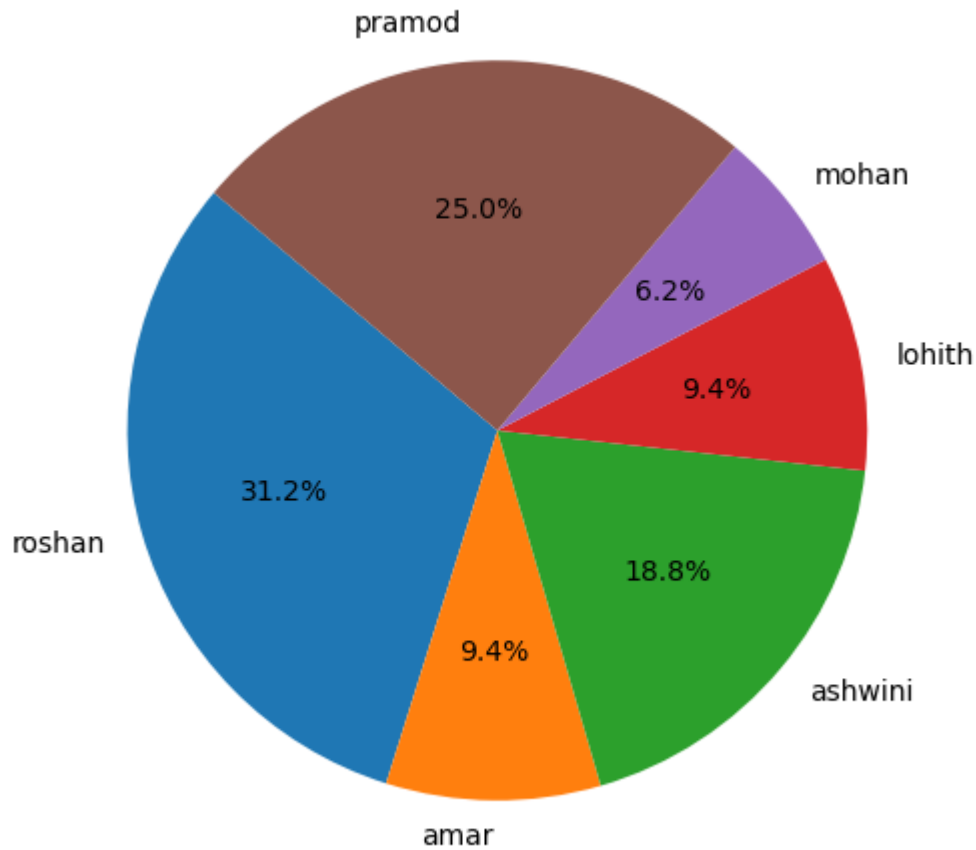
~\anaconda3\lib\site-packages\pandas\_libs\ops.pyx in pandas._libs.ops.scalar_compare()

TypeError: '>' not supported between instances of 'str' and 'int'
```

```
In [55]: 1 import matplotlib.pyplot as plt
2 a=df['salary']
3 b=df['Ename']
4 plt.figure(figsize=(8, 5))
5 plt.bar(b,a)
6 plt.xlabel('Ename')
7 plt.ylabel('salary')
8 plt.title('bar graph')
9 plt.show()
```



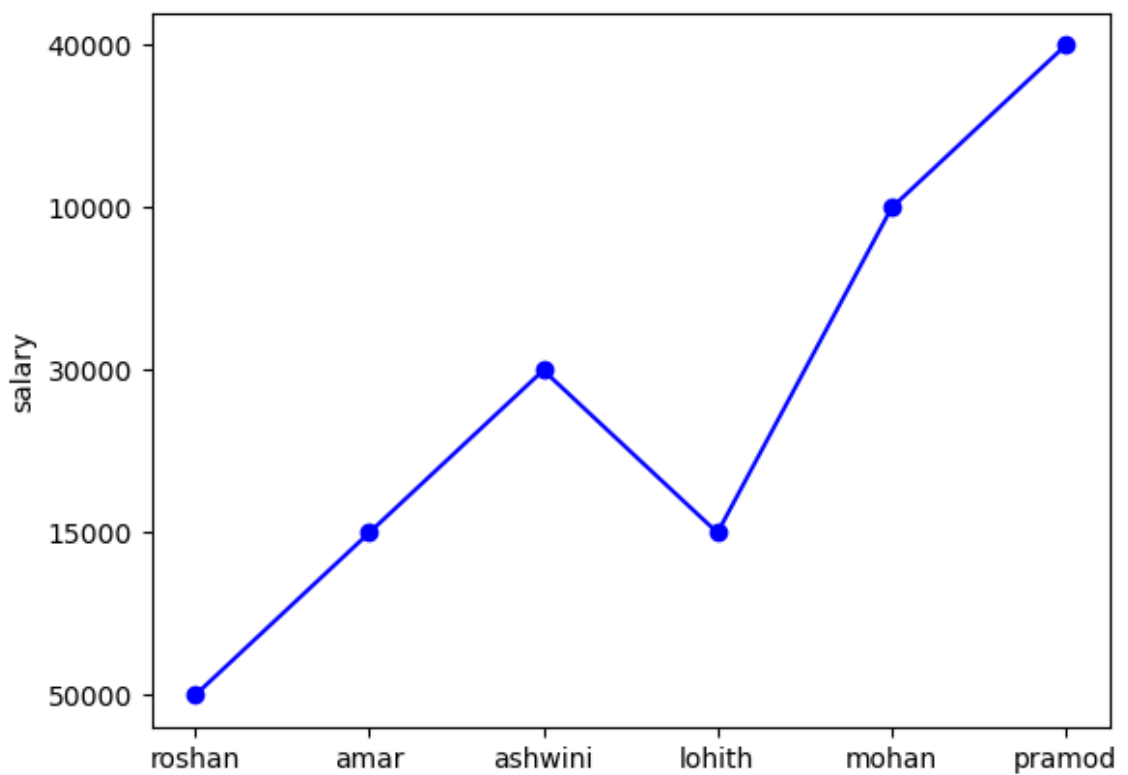
```
In [61]: 1 import matplotlib.pyplot as plt
2 subject = 'salary'
3 plt.figure(figsize=(6,6))
4 plt.pie(df[subject], labels=df['Ename'], autopct = '%1.1f%%', startangle
5 plt.title = ('pie chart')
6 plt.show()
```



```
In [73]: 1 plt.plot(df['Ename'], df['salary'], marker='o',linestyle='-',color='b')
          2 plt.ylabel('salary')
          3 plt.title('line graph')
          4 plt.show()
```

```
-
TypeError                                Traceback (most recent call las
t)
~\AppData\Local\Temp\ipykernel_8372\873038540.py in <module>
      1 plt.plot(df['Ename'], df['salary'], marker='o',linestyle='-',color
='b')
      2 plt.ylabel('salary')
----> 3 plt.title('line graph')
      4 plt.show()
```

TypeError: 'str' object is not callable



```
In [69]: 1 plt.plot(df['Ename'], df['salary'], marker='o', linestyle='-', color='b')
2 plt.xlabel('Eame')
3 plt.ylabel('Frequence')
4 plt.title('line graph')
5 plt.show()
```

TypeError Traceback (most recent call last)

~\AppData\Local\Temp\ipykernel_8372\3704409374.py in <module>

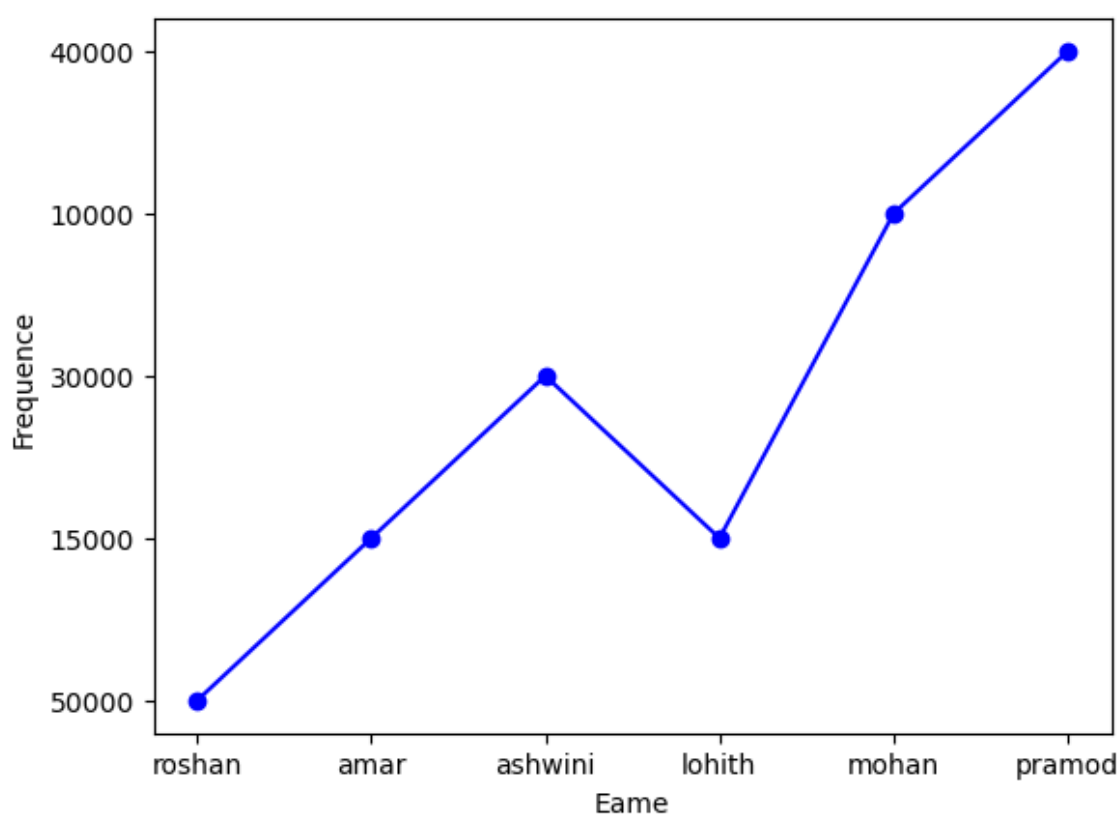
2 plt.xlabel('Eame')

3 plt.ylabel('Frequence')

----> 4 plt.title('line graph')

5 plt.show()

TypeError: 'str' object is not callable



out liers

```
In [ ]: 1
```