


Tên: Nguyễn Trọng Nghĩa MSSV: 207CT40473

1. Tạo một DataFrame từ dữ liệu trên

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eva', 'Frank', 'Grance', 'Hannad', 'Ivan', 'Jack', 'Kelly', 'Liam', 'Mona', 'Nina', 'Oscar'],
    'Age': [25, 30, 35, 28, 22, 45, 34, 31, 27, 29, 33, 40, 26, 32, 36],
    'Salary': ['50000', '60000', '70000', '55000', '52000', '80000', '72000', '68000', '61000', '59000', '63000', '77000', '53000', '66000', '75000'],
}
df = pd.DataFrame(data)
```


```
print(df)
```



	Name	Age	Salary
0	Alice	25	50000
1	Bob	30	60000
2	Charlie	35	70000
3	David	28	55000
4	Eva	22	52000
5	Frank	45	80000
6	Grance	34	72000
7	Hannad	31	68000
8	Ivan	27	61000
9	Jack	29	59000
10	Kelly	33	63000
11	Liam	40	77000
12	Mona	26	53000
13	Nina	32	66000
14	Oscar	36	75000

2. Hiển thị thông tin về DataFrame vừa tạo

```
print(df.head(15))
```




	Name	Age	Salary
0	Alice	25	50000
1	Bob	30	60000
2	Charlie	35	70000
3	David	28	55000
4	Eva	22	52000
5	Frank	45	80000
6	Grance	34	72000
7	Hannad	31	68000
8	Ivan	27	61000
9	Jack	29	59000
10	Kelly	33	63000
11	Liam	40	77000
12	Mona	26	53000
13	Nina	32	66000
14	Oscar	36	75000

3. Lọc các hàng trong DataFrame có 'Age' lớn hơn 28

```
df_3 = df[df['Age'] > 28]
```

```
print(df_3)
```



	Name	Age	Salary
1	Bob	30	60000
2	Charlie	35	70000
5	Frank	45	80000
6	Grance	34	72000
7	Hannad	31	68000
9	Jack	29	59000
10	Kelly	33	63000
11	Liam	40	77000
13	Nina	32	66000

```
14 Oscar 36 75000
```

4. Tính giá trị trung bình của cột 'Salary'.

```
df_4 = df['Salary'].mean()
```

```
print(f"Giá trị trung bình của cột Salary", df_4)
```

```
→ Giá trị trung bình của cột Salary 3.333373333800004e+73
```

5. Nhóm dữ liệu theo cột 'Age' và tính tổng 'Salary' cho mỗi nhóm

```
df_5 = df.groupby('Age')['Salary'].sum().reset_index()
```

```
print(df_5)
```

```
→
```

	Age	Salary
0	22	52000
1	25	50000
2	26	53000
3	27	61000
4	28	55000
5	29	59000
6	30	60000
7	31	68000
8	32	66000
9	33	63000
10	34	72000
11	35	70000
12	36	75000
13	40	77000
14	45	80000

6. Sắp xếp DataFrame theo cột 'Salary' giảm dần

```
df_6 = df.sort_values(by=['Salary'],ascending=False)
```

```
print(df_6)
```

```
→
```

	Name	Age	Salary
5	Frank	45	80000
11	Liam	40	77000
14	Oscar	36	75000
6	Grance	34	72000
2	Charlie	35	70000
7	Hannad	31	68000
13	Nina	32	66000
10	Kelly	33	63000
8	Ivan	27	61000
1	Bob	30	60000
9	Jack	29	59000
3	David	28	55000
12	Mona	26	53000
4	Eva	22	52000
0	Alice	25	50000

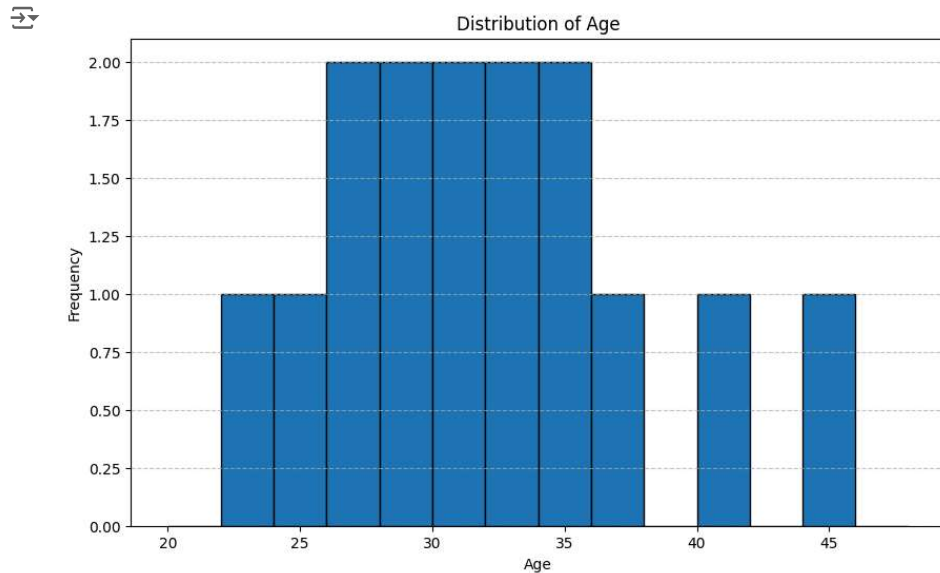
7. Vẽ biểu đồ cột cho cột 'Age'

```
import matplotlib.pyplot as plt
import pandas as pd

df = pd.DataFrame(data)

plt.figure(figsize=(10, 6))
plt.hist(df['Age'], bins=range(20, 50, 2), edgecolor='black')
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.grid(axis='y', linestyle='--', alpha=0.7)

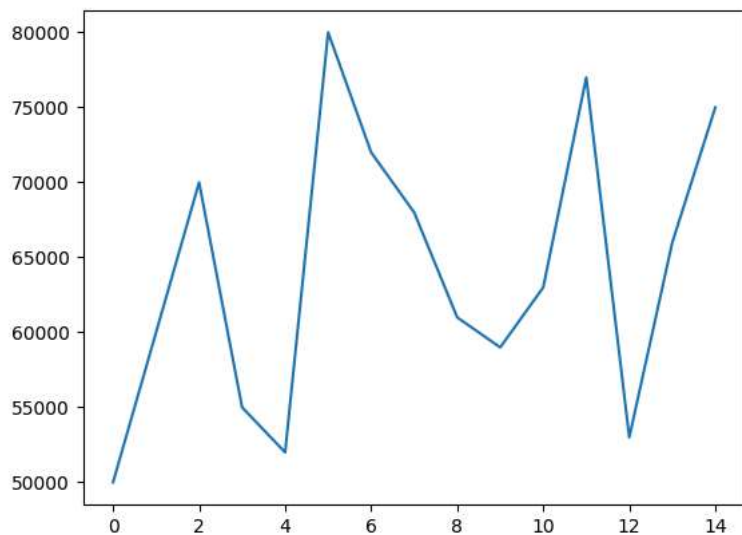
plt.show()
```



8. Vẽ biểu đồ đường cho cột 'Salary'

```
print(df['Salary'].dtype)
df['Salary'] = pd.to_numeric(df['Salary'])
print(df['Salary'].dtype)
df['Salary'].plot.line()
```

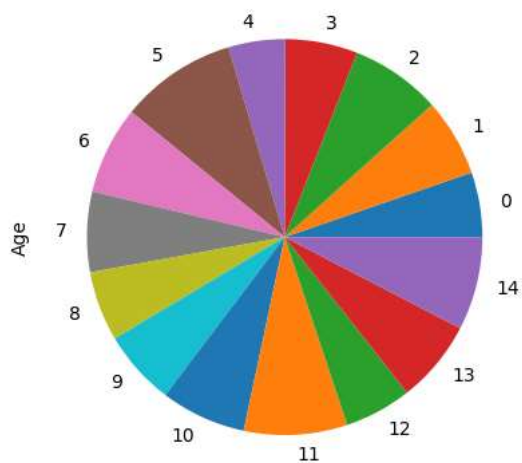
```
object
int64
<AxesSubplot:>
```



9. Vẽ biểu đồ tròn cho cột 'Age'

```
df['Age'].plot.pie()
```

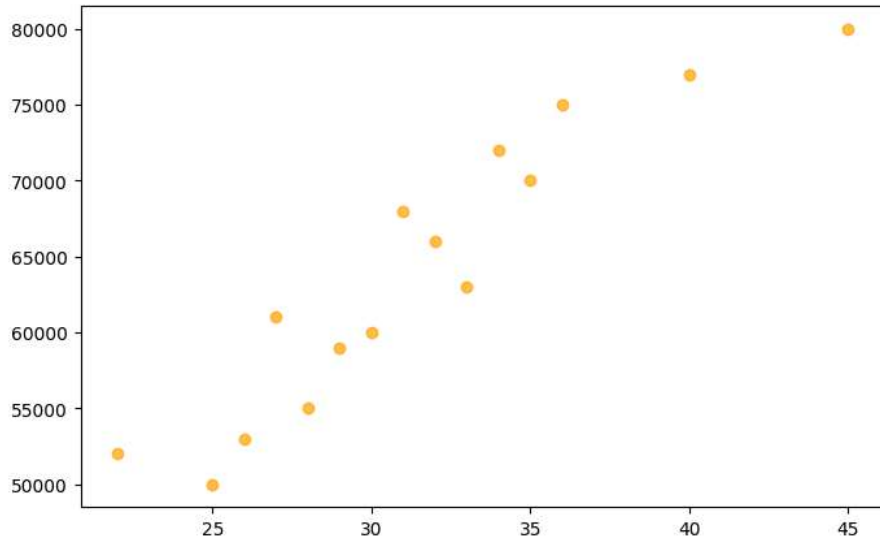
```
<AxesSubplot:ylabel='Age'>
```



10. Vẽ biểu đồ phân tán cho 'Age' và 'Salary'

```
plt.figure(figsize=(8, 5))
plt.scatter(df['Age'], df['Salary'], color='orange', marker='o', alpha=0.7)
```


 <matplotlib.collections.PathCollection at 0x1abad7b79d0>



11. Kiểm tra xem có giá trị NaN nào trong DataFrame không

```
nan_values = df.isna().any().any()

if nan_values:
    print("DataFrame có giá trị NaN.")
else:
    print("DataFrame sẽ không bao giờ có NaN.")
```


 DataFrame sẽ không bao giờ có NaN.

12. Thay thế các giá trị của cột 'Age' lớn hơn 30 bằng giá trị trung bình của cột đó

```
df_12 = df['Age'].mean()

df.loc[df['Age'] > 30, 'Age'] = df_12
```


```
print(df)
```



	Name	Age	Salary
0	Alice	25.000000	50000
1	Bob	30.000000	60000
2	Charlie	31.533333	70000
3	David	28.000000	55000
4	Eva	22.000000	52000
5	Frank	31.533333	80000
6	Grance	31.533333	72000
7	Hannad	31.533333	68000
8	Ivan	27.000000	61000
9	Jack	29.000000	59000
10	Kelly	31.533333	63000
11	Liam	31.533333	77000
12	Mona	26.000000	53000
13	Nina	31.533333	66000
14	Oscar	31.533333	75000

13. Chuẩn hóa(normalize) cột 'Age' về khoảng giá trị từ 0 đến 1

```
df['Age_normalized'] = (df['Age'] - df['Age'].min()) / (df['Age'].max() - df['Age'].min())
print(df)
```




	Name	Age	Salary	Age_normalized	Salary_pct_change
0	Alice	25	50000	0.130435	NaN
1	Bob	30	60000	0.347826	20.000000
2	Charlie	35	70000	0.565217	16.666667

3	David	28	55000	0.260870	-21.428571
4	Eva	22	52000	0.000000	-5.454545
5	Frank	45	80000	1.000000	53.846154
6	Grance	34	72000	0.521739	-10.000000
7	Hannad	31	68000	0.391304	-5.555556
8	Ivan	27	61000	0.217391	-10.294118
9	Jack	29	59000	0.304348	-3.278689
10	Kelly	33	63000	0.478261	6.779661
11	Liam	40	77000	0.782609	22.222222
12	Mona	26	53000	0.173913	-31.168831
13	Nina	32	66000	0.434783	24.528302
14	Oscar	36	75000	0.608696	13.636364

14. Tạo một cột mới 'Age_group' phân loại tuổi thành 'Young', 'Middle-aged' và 'Old' dựa trên giá trị của cột 'Age'


```
def age_group(age):
    if age < 30:
        return 'Young'
    elif age < 60:
        return 'Middle-aged'
    else:
        return 'Old'
df['Age_group'] = df['Age'].apply(age_group)
df = df.drop('Age', axis=1)
print(df)
```



	Name	Salary	Age_group
0	Alice	50000	Young
1	Bob	60000	Middle-aged
2	Charlie	70000	Middle-aged
3	David	55000	Young
4	Eva	52000	Young
5	Frank	80000	Middle-aged
6	Grance	72000	Middle-aged
7	Hannad	68000	Middle-aged
8	Ivan	61000	Young
9	Jack	59000	Young
10	Kelly	63000	Middle-aged
11	Liam	77000	Middle-aged
12	Mona	53000	Young
13	Nina	66000	Middle-aged
14	Oscar	75000	Middle-aged

15. Tính toán tỷ lệ phần trăm thay đổi(percentage change) của cột 'Salary'

```
df['Salary_pct_change'] = df['Salary'].pct_change() * 100
print(df[['Name', 'Salary', 'Salary_pct_change']])
```



	Name	Salary	Salary_pct_change
0	Alice	50000	NaN
1	Bob	60000	20.000000
2	Charlie	70000	16.666667
3	David	55000	-21.428571
4	Eva	52000	-5.454545
5	Frank	80000	53.846154
6	Grance	72000	-10.000000