

**Đinh Thạch Bảo - 207CT27605**

## ▼ câu 1

# @title câu 1

import pandas as pd

```
data = {
    "Name": ["Alice", "Bob", "Charlie", "David", "Eva", "Frank", "Grace", "Hannah", "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   Grace   34   72000
7  Hannah   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
```

## ▼ câu 2

# @title câu 2

```
print("Mô tả thống kê của DataFrame vừa tạo:")
print(df.describe())
```

```
print("Thông tin vừa tạo DataFrame:")
df.info()
```

```
↗ Mô tả thống kê của DataFrame vừa tạo:
      Age      Salary
count  15.000000    15.000000
mean   31.533333   64066.666667
std     5.974549    9482.816540
min    22.000000    50000.000000
25%    27.500000    57000.000000
50%    31.000000    63000.000000
75%    34.500000    71000.000000
max    45.000000    80000.000000
Thông tin vừa tạo DataFrame:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  -
 0   Name    15 non-null      object
 1   Age     15 non-null      int64
 2   Salary  15 non-null      int64
dtypes: int64(2), object(1)
memory usage: 488.0+ bytes
```

## ▼ câu 3

```
# @title câu 3
```

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

```
print(filtered_df)
```

```

1      Name  Age  Salary
2    Charlie  35   70000
5      Frank  45   80000
6      Grace  34   72000
7    Hannah  31   68000
9       Jack  29   59000
10     Kelly  33   63000
11      Liam  40   77000
13      Nina  32   66000
14     Oscar  36   75000

```

#### ▼ câu 4

```
# @title câu 4
```

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

```
print("Giá trị tb của cột Salary:", average_salary)
```

```

Giá trị tb của cột Salary: 64066.666666666664

```

#### ▼ câu 5

```
# @title câu 5
```

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

```
print(grouped_df)
```

```

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

```

#### ▼ câu 6

```
# @title câu 6
```

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

```
print(sorted_df)
```

```

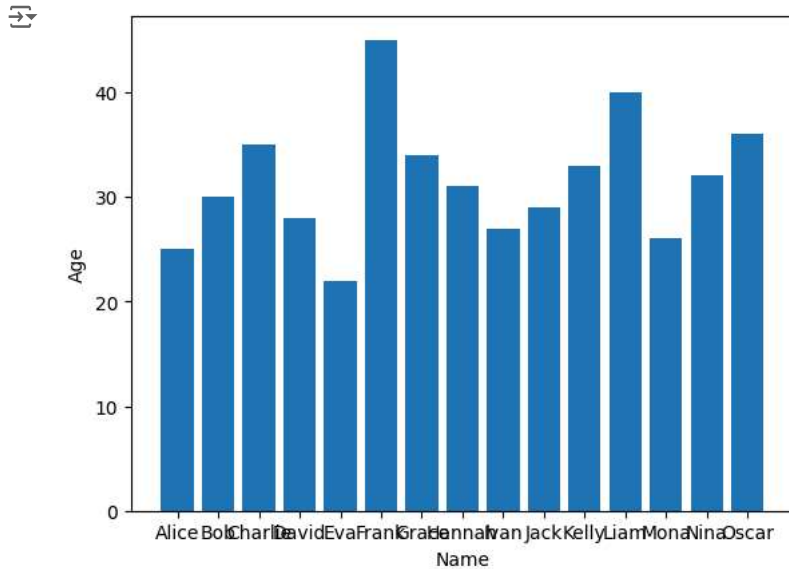
Name  Age  Salary
5    Frank  45   80000
11     Liam  40   77000
14     Oscar  36   75000
6      Grace  34   72000
2    Charlie  35   70000
7    Hannah  31   68000
13     Nina  32   66000
10     Kelly  33   63000
8       Ivan  27   61000
1       Bob  30   60000
9       Jack  29   59000
3      David 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

```

## ▼ câu 7

```
# @title câu 7
import matplotlib.pyplot as plt

plt.bar(df['Name'], df['Age'])
plt.xlabel('Name')
plt.ylabel('Age')
plt.show()
```



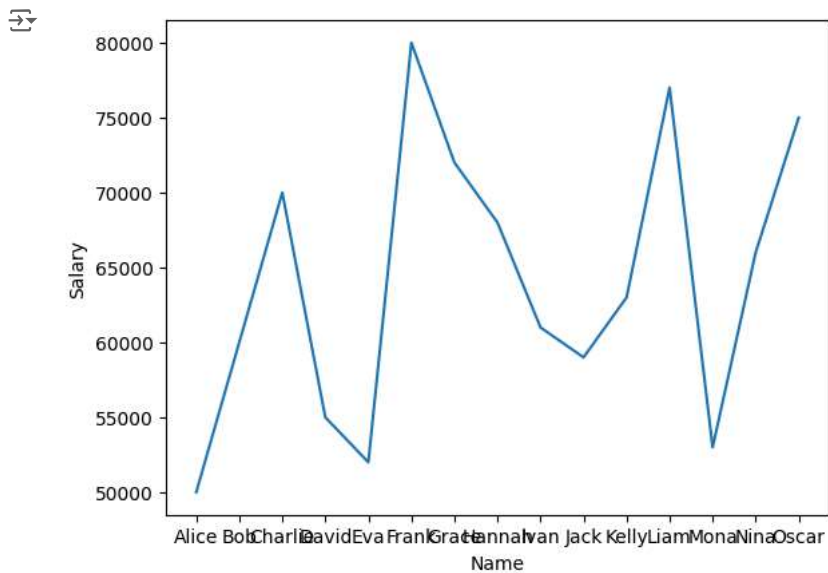
Start coding or [generate](#) with AI.

## ▼ câu 8

```
# @title câu 8

plt.plot(df['Name'], df['Salary'])
plt.xlabel('Name')
plt.ylabel('Salary')

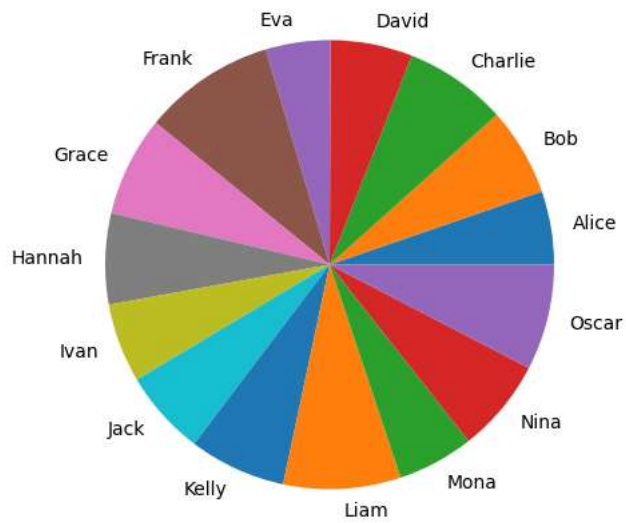
plt.show()
```



## ▼ câu 9

# @title câu 9

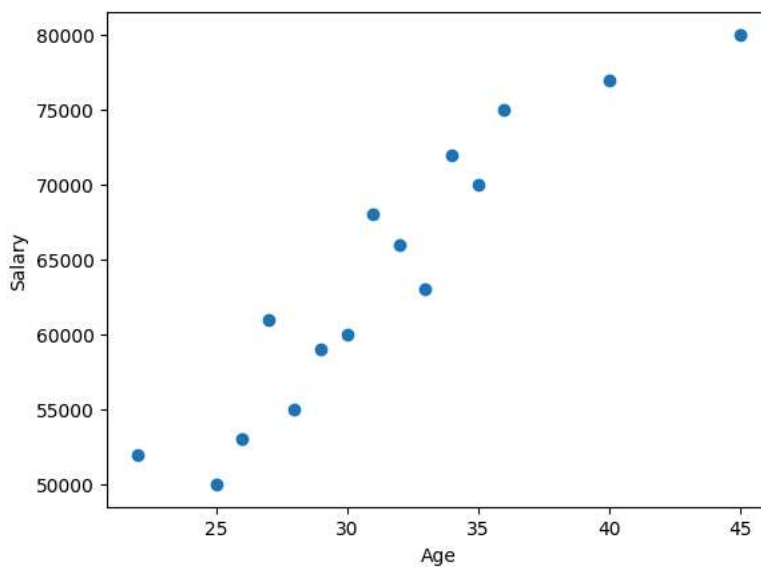
```
plt.pie(df['Age'], labels=df['Name'])
plt.axis('equal')
plt.show()
```



## ▼ câu 10

# @title câu 10


```
plt.scatter(df['Age'], df['Salary'])
plt.xlabel('Age')
plt.ylabel('Salary')
plt.show()
```



## ▼ câu 11

```
# @title câu 11
nan_exists = df.isnull().any().any()


if nan_exists:
    print("co NaN trong DataFrame.")
else:
    print("Ko co NaN trong DataFrame.")
```

 co NaN trong DataFrame.

## ▼ câu 12

```
# @title câu 12
mean_age = df['Age'].mean()
df['Age'] = df['Age'].apply(lambda x: mean_age if x > 30 else x)

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	Grace	31.533333	72000
7	Hannah	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

## ▼ câu 13

```
# @title câu 13

df['normalized'] = (df['Age'] - df['Age'].min()) / (df['Age'].max() - df['Age'].min())

print(df[['Name', 'Age', 'normalized']])
```



	Name	Age	normalized
0	Alice	25	0.130435
1	Bob	30	0.347826
2	Charlie	35	0.565217
3	David	28	0.260870
4	Eva	22	0.000000
5	Frank	45	1.000000
6	Grace	34	0.521739
7	Hannah	31	0.391304
8	Ivan	27	0.217391
9	Jack	29	0.304348
10	Kelly	33	0.478261
11	Liam	40	0.782609
12	Mona	26	0.173913
13	Nina	32	0.434783
14	Oscar	36	0.608696

## ▼ câu 14

```
# @title câu 14
def classify_age(age):
    if age < 35:
        return 'tre'
    elif age < 45:
        return 'trung nien'
    else:
        return 'gia'

# Tạo cột 'age_group' dựa trên cột 'Age'
df['Nhom tuoi'] = df['Age'].apply(lambda x: classify_age(x))

print(df[['Name', 'Age', 'Nhom tuoi']])
```

```

↩
      Name Age  Nhom tuoi
0    Alice 25      tre
1     Bob 30      tre
2  Charlie 35  trung nien
3    David 28      tre
4     Eva 22      tre
5    Frank 45      gia
6   Grace 34      tre
7  Hannah 31      tre
8     Ivan 27      tre
9     Jack 29      tre
10    Kelly 33      tre
11    Liam 40  trung nien
12    Mona 26      tre
13    Nina 32      tre
14   Oscar 36  trung nien

```

## ▼ câu 15

```

# @title câu 15
df['Salary_change_percent'] = ((df['Salary'] - df['Salary'].shift(1)) / df['Salary'].shift(1)) * 100

print(df[['Salary', 'Salary_change_percent']])

```

```

↩
      Salary  Salary_change_percent
0    50000             NaN
1    60000          20.000000
2    70000          16.666667
3    55000         -21.428571
4    52000          -5.454545
5    80000          53.846154
6    72000         -10.000000
7    68000          -5.555556
8    61000         -10.294118
9    59000          -3.278689
10   63000           6.779661
11   77000          22.222222
12   53000         -31.168831
13   66000          24.528302
14   75000          13.636364

```

## ▼ câu 16

```

# @title câu 16
df = df.drop_duplicates(subset=['Name'], keep='first')

```

```
print(df)
```

```

↩
      Name Age  Salary  Age_normalized  normalized  age_group  Nhom tuoi \
0    Alice 25   50000      0.130435    0.130435      tre      tre
1     Bob 30   60000      0.347826    0.347826      tre      tre
2  Charlie 35   70000      0.565217    0.565217  trung nien  trung nien
3    David 28   55000      0.260870    0.260870      tre      tre
4     Eva 22   52000      0.000000    0.000000      tre      tre
5    Frank 45   80000      1.000000    1.000000      gia      gia
6   Grace 34   72000      0.521739    0.521739      tre      tre
7  Hannah 31   68000      0.391304    0.391304      tre      tre
8     Ivan 27   61000      0.217391    0.217391      tre      tre
9     Jack 29   59000      0.304348    0.304348      tre      tre
10    Kelly 33   63000      0.478261    0.478261      tre      tre
11    Liam 40   77000      0.782609    0.782609  trung nien  trung nien
12    Mona 26   53000      0.173913    0.173913      tre      tre
13    Nina 32   66000      0.434783    0.434783      tre      tre
14   Oscar 36   75000      0.608696    0.608696  trung nien  trung nien

      Salary_change_percent
0             NaN
1          20.000000
2          16.666667
3         -21.428571
4          -5.454545
5          53.846154
6         -10.000000
7          -5.555556
8         -10.294118
9          -3.278689
10           6.779661
11          22.222222

```

12	-31.168831
13	24.528302
14	13.636364