baikiemtraso1

June 27, 2024

[34]:
```
import pandas as pd
import numpy as np
```

[35]:
```
#Câu 1: Tạo dataframe
names = np.array(['Alice', 'Bob', 'Charlie', 'David', 'Eva', 'Frank', 'Grace',␣
 ↪'Hannah', 'Ivan', 'Jack', 'Kelly', 'Liam', 'Mona', 'Nina', 'Oscar'])
ages = np.array([25, 30, 35, 28, 22, 45, 34, 31, 27, 29, 33, 40, 26, 32, 36])
salaries = np.array([50000, 60000, 70000,␣
 ↪55000,52000,80000,72000,68000,61000,59000,63000,77000,53000,66000,75000])
data = pd.DataFrame({'name': names,'age': ages,'salary': salaries})
```

[36]:
```
#Câu 2: Hiển thị thông tin về DataFrame
print(data)
```

```
       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
```

[37]:
```
#Câu 3: Lọc các hàng có giá trị lớn hơn 28 trong cột salary
data[data['age'] >= 28]
data.groupby('name')['name'].agg('count')
```

[37]:
```
name
Alice      1
```

```
Bob        1
Charlie    1
David      1
Eva        1
Frank      1
Grace      1
Hannah     1
Ivan       1
Jack       1
Kelly      1
Liam       1
Mona       1
Nina       1
Oscar      1
Name: name, dtype: int64
```

[38]:
```
#Câu 4: Tính giá trị tb của cột salary
mean_salary = np.mean(data['salary'].values)
print(mean_salary)
```

```
64066.666666666664
```

[39]:
```
#Câu 5: Nhóm dữ liệu theo cột age và tính tổng salary cho mỗi nhóm
data.groupby('age')['salary'].sum().reset_index()
print(data)
```

```
      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
```

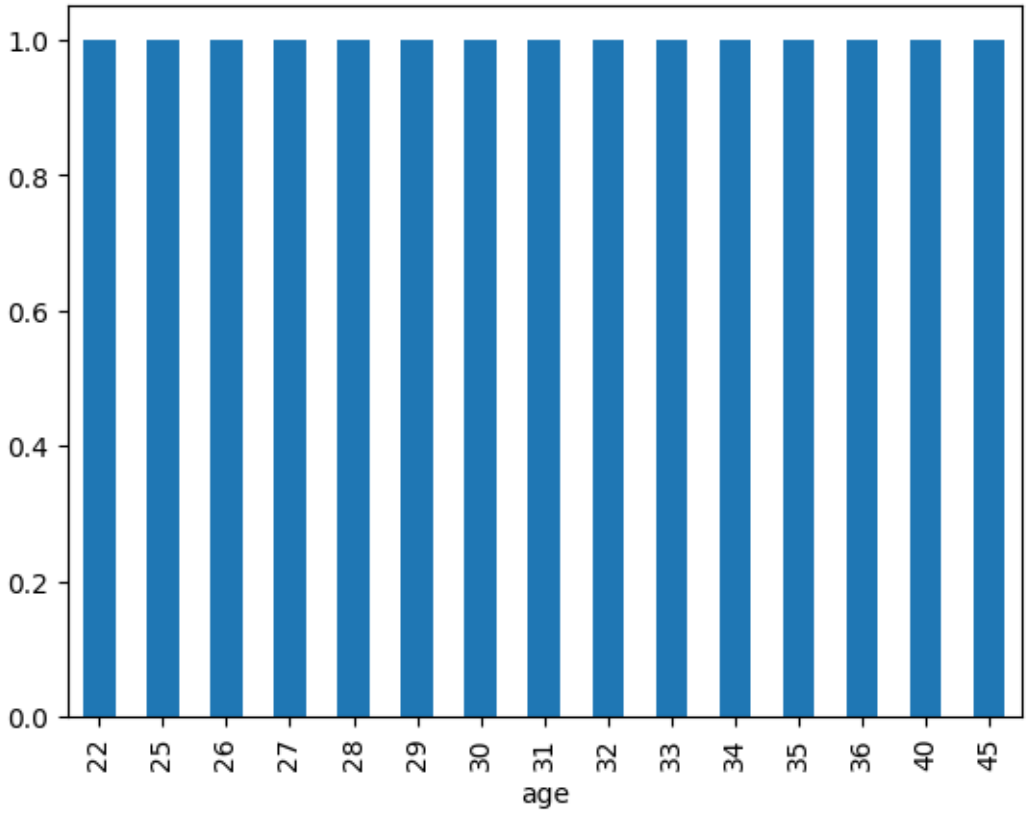
[40]:
```
#Câu 6: sắp xếp dataframe theo cột salary giảm dần
data.sort_values(by="salary", ascending=False)
print(data)
```

```
      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
```

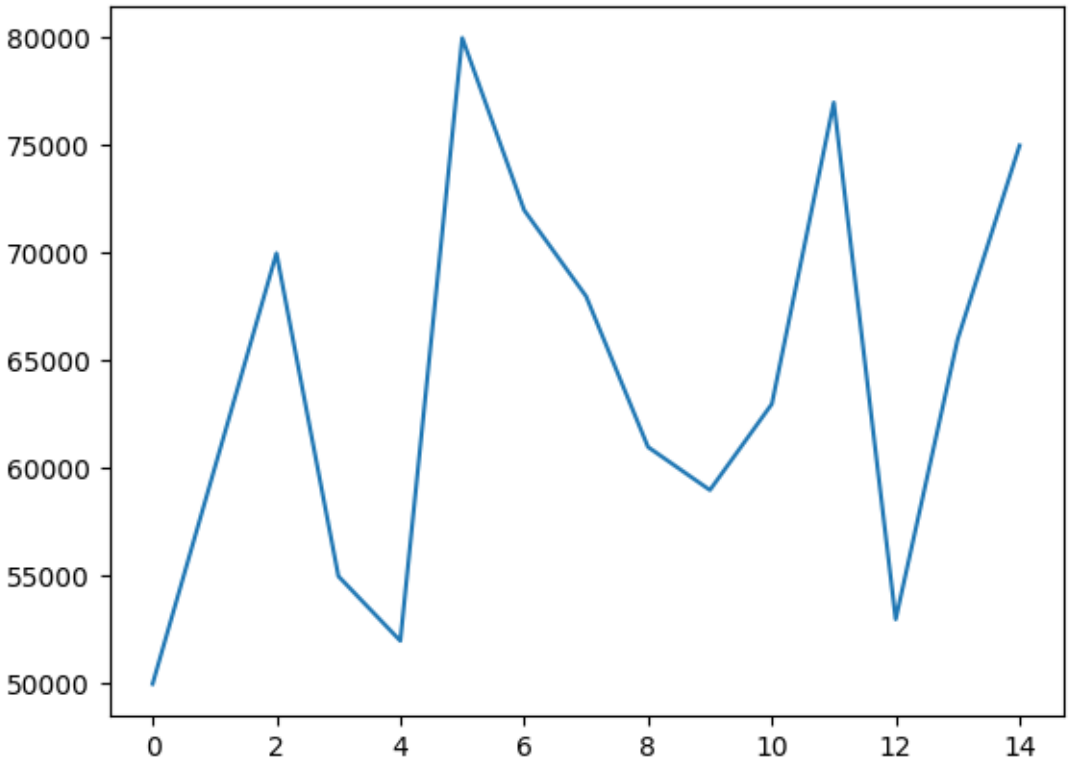
[41]:
```
#Câu 7: Vẽ biểu đồ cho cột age
cau7 =data.groupby('age')['age'].agg('count')
cau7.plot.bar()
```

[41]: <AxesSubplot:xlabel='age'>



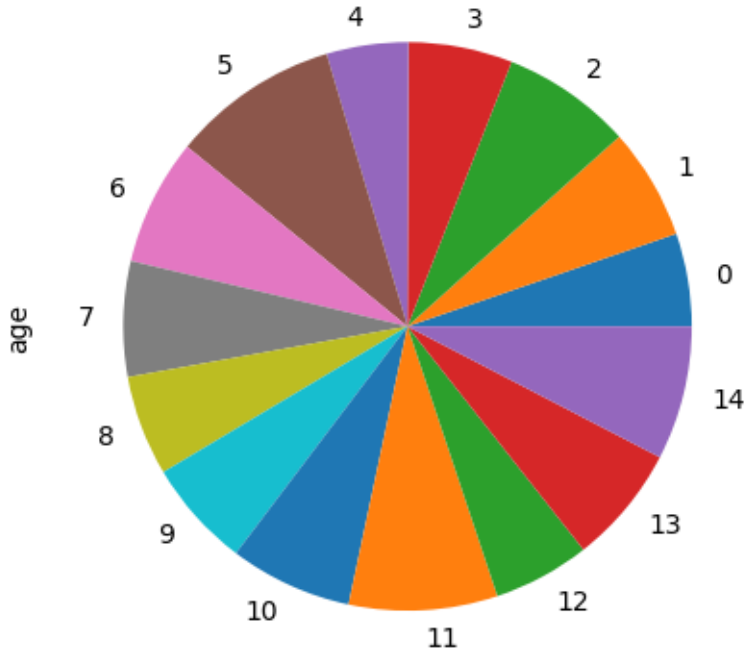
[42]:
```
#Câu 8: vẽ biểu đồ đường cho cột salary
data['salary'].plot()
```

[42]: <AxesSubplot:>



[43]:
```
#Câu 9: vẽ biểu đồ tròn cho cột age
data['age'].plot.pie()
```

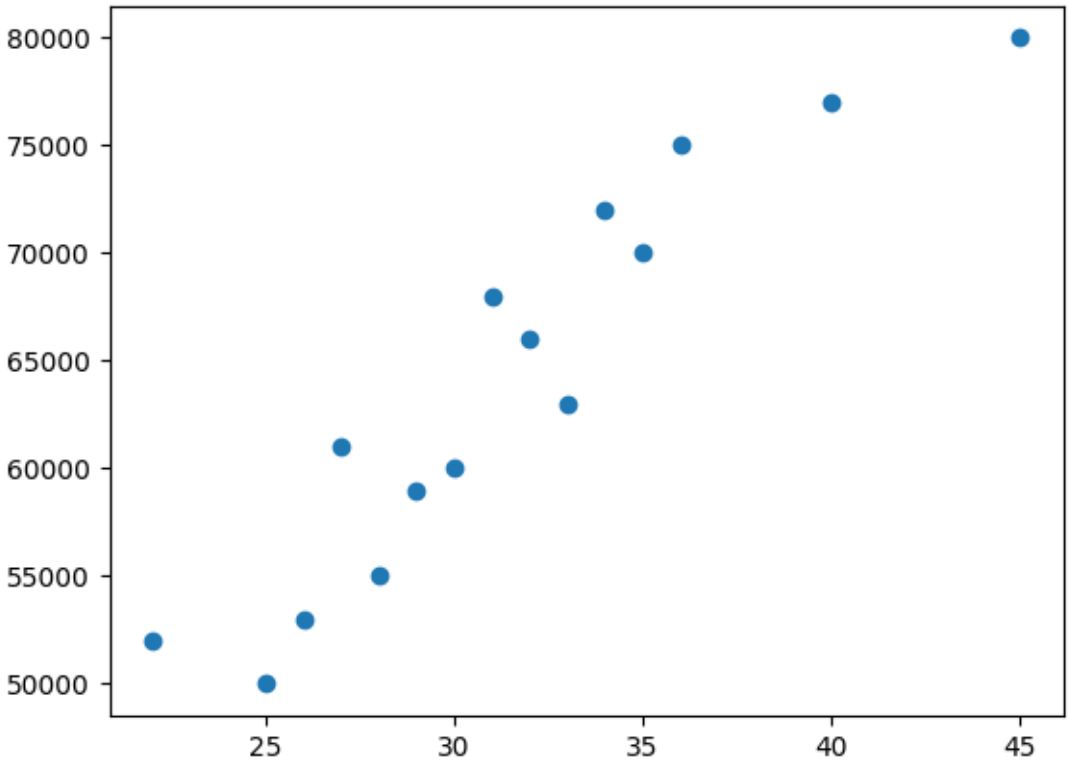
[43]: <AxesSubplot:ylabel='age'>



[44]:
```
#Câu 10: Vẽ biểu đồ phân tán cho age và salary
import matplotlib.pyplot as plt
correlation_dh1_t1 = data[['age', 'salary']].corr()
print(correlation_dh1_t1)
plt.scatter(data['age'], data['salary'])
```

```
              age    salary
age      1.000000  0.937327
salary   0.937327  1.000000
```

[44]: <matplotlib.collections.PathCollection at 0x292fd65df00>



[45]:
```
#Câu 11: Kiểm tra xem có giá trị NaN nào trong dataframe không
data.isnull().sum()
```

[45]:
```
name      0
age       0
salary    0
dtype: int64
```

[46]:
```
#Câu 12: thay thế các giá trị ở cột age lớn hơn 30 bằng giá trị tb của cột đó
tuoitb = data['age'].mean()

data.loc[data['age'] > 30, 'age'] = tuoitb

print(data)
```

```
      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
```

[47]:
```
#Câu 13: Chuẩn hóa cột age về khảong giá trị từ 0 đến 1
data['age_chuanhoa'] = (data['age'] - data['age'].min()) / (data['age'].max() -␣
 ↪data['age'].min())

print(data)
```

```
      name        age  salary  age_chuanhoa
0    Alice  25.000000   50000      0.314685
1      Bob  30.000000   60000      0.839161
2  Charlie  31.533333   70000      1.000000
3    David  28.000000   55000      0.629371
4      Eva  22.000000   52000      0.000000
5    Frank  31.533333   80000      1.000000
6    Grace  31.533333   72000      1.000000
7   Hannah  31.533333   68000      1.000000
8     Ivan  27.000000   61000      0.524476
9     Jack  29.000000   59000      0.734266
10   Kelly  31.533333   63000      1.000000
11    Liam  31.533333   77000      1.000000
12    Mona  26.000000   53000      0.419580
13    Nina  31.533333   66000      1.000000
14   Oscar  31.533333   75000      1.000000
```

[48]:
```
#Câu 14: tạo cột mới age_group phân loại thành young, middle-age và old
def categorize_age(age):
    if age < 30:
        return 'young'
    elif 30 <= age <= 60:
        return 'middle-age'
    else:
        return 'old'

data['age_group'] = data['age'].apply(categorize_age)
print(data)
```

```
      name        age  salary  age_chuanhoa   age_group
0    Alice  25.000000   50000      0.314685       young
1      Bob  30.000000   60000      0.839161  middle-age
```

```
2   Charlie  31.533333   70000      1.000000  middle-age
3     David  28.000000   55000      0.629371       young
4       Eva  22.000000   52000      0.000000       young
5     Frank  31.533333   80000      1.000000  middle-age
6     Grace  31.533333   72000      1.000000  middle-age
7    Hannah  31.533333   68000      1.000000  middle-age
8      Ivan  27.000000   61000      0.524476       young
9      Jack  29.000000   59000      0.734266       young
10    Kelly  31.533333   63000      1.000000  middle-age
11     Liam  31.533333   77000      1.000000  middle-age
12     Mona  26.000000   53000      0.419580       young
13     Nina  31.533333   66000      1.000000  middle-age
14    Oscar  31.533333   75000      1.000000  middle-age
```

[49]:
```
#Câu 15: tính tỷ lệ % thay đổi của cột salary
data['salary_tltd'] = data['salary'].pct_change() * 100

print(data)
```

```
      name        age  salary  age_chuanhoa   age_group  salary_tltd
0    Alice  25.000000   50000      0.314685       young          NaN
1      Bob  30.000000   60000      0.839161  middle-age    20.000000
2  Charlie  31.533333   70000      1.000000  middle-age    16.666667
3    David  28.000000   55000      0.629371       young   -21.428571
4      Eva  22.000000   52000      0.000000       young    -5.454545
5    Frank  31.533333   80000      1.000000  middle-age    53.846154
6    Grace  31.533333   72000      1.000000  middle-age   -10.000000
7   Hannah  31.533333   68000      1.000000  middle-age    -5.555556
8     Ivan  27.000000   61000      0.524476       young   -10.294118
9     Jack  29.000000   59000      0.734266       young    -3.278689
10   Kelly  31.533333   63000      1.000000  middle-age     6.779661
11    Liam  31.533333   77000      1.000000  middle-age    22.222222
12    Mona  26.000000   53000      0.419580       young   -31.168831
13    Nina  31.533333   66000      1.000000  middle-age    24.528302
14   Oscar  31.533333   75000      1.000000  middle-age    13.636364
```

[50]:
```
#câu 16: tìm các giá trị trùng lập trong cột name và loại bỏ các hàng trùng␣
 ↪lập, giữ lại hàng đầu tiên
data.drop_duplicates(subset='name', keep='first')

print(data)
```

```
      name        age  salary  age_chuanhoa   age_group  salary_tltd
0    Alice  25.000000   50000      0.314685       young          NaN
1      Bob  30.000000   60000      0.839161  middle-age    20.000000
2  Charlie  31.533333   70000      1.000000  middle-age    16.666667
3    David  28.000000   55000      0.629371       young   -21.428571
4      Eva  22.000000   52000      0.000000       young    -5.454545
```

5	Frank	31.533333	80000	1.000000	middle-age	53.846154
6	Grace	31.533333	72000	1.000000	middle-age	-10.000000
7	Hannah	31.533333	68000	1.000000	middle-age	-5.555556
8	Ivan	27.000000	61000	0.524476	young	-10.294118
9	Jack	29.000000	59000	0.734266	young	-3.278689
10	Kelly	31.533333	63000	1.000000	middle-age	6.779661
11	Liam	31.533333	77000	1.000000	middle-age	22.222222
12	Mona	26.000000	53000	0.419580	young	-31.168831
13	Nina	31.533333	66000	1.000000	middle-age	24.528302
14	Oscar	31.533333	75000	1.000000	middle-age	13.636364

[51]:
```
#Câu 17: lưu data thành file csv
data.to_csv('bài kiểm tra số 1.csv', index=False)
```