kiemtragiuaky

June 27, 2024

[41]:
```
import pandas as pd
import numpy as np
```

[42]:
```
# Cau 1:
data = {
    'Name': ['Alice', 'Bob',␣
 ↪'Charlie','David','Eva','frank','Grace','Hannah','Ivan','Jack','Kelly','Liam','Mona','Nina'
    '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)
print(df.columns)
print(df.index)
```

```
       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
Index(['Name', 'Age', 'Salary'], dtype='object')
RangeIndex(start=0, stop=15, step=1)
```

[43]:
```
# Cau2:
df.head(15)
```

[43]:

	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

[44]:
```
# Cau 3:
age_df = df[df['Age'] > 28]
print(age_df)
```

	Name	Age	Salary
1	Bob	30	60000
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

[45]:
```
# Cau 4:
average_salary = df['Salary'].mean()
print(average_salary)
```

```
64066.666666666664
```

[46]:
```
# Cau 5:
group_df = df.groupby('Age')['Salary'].sum().reset_index()
print(group_df)
```

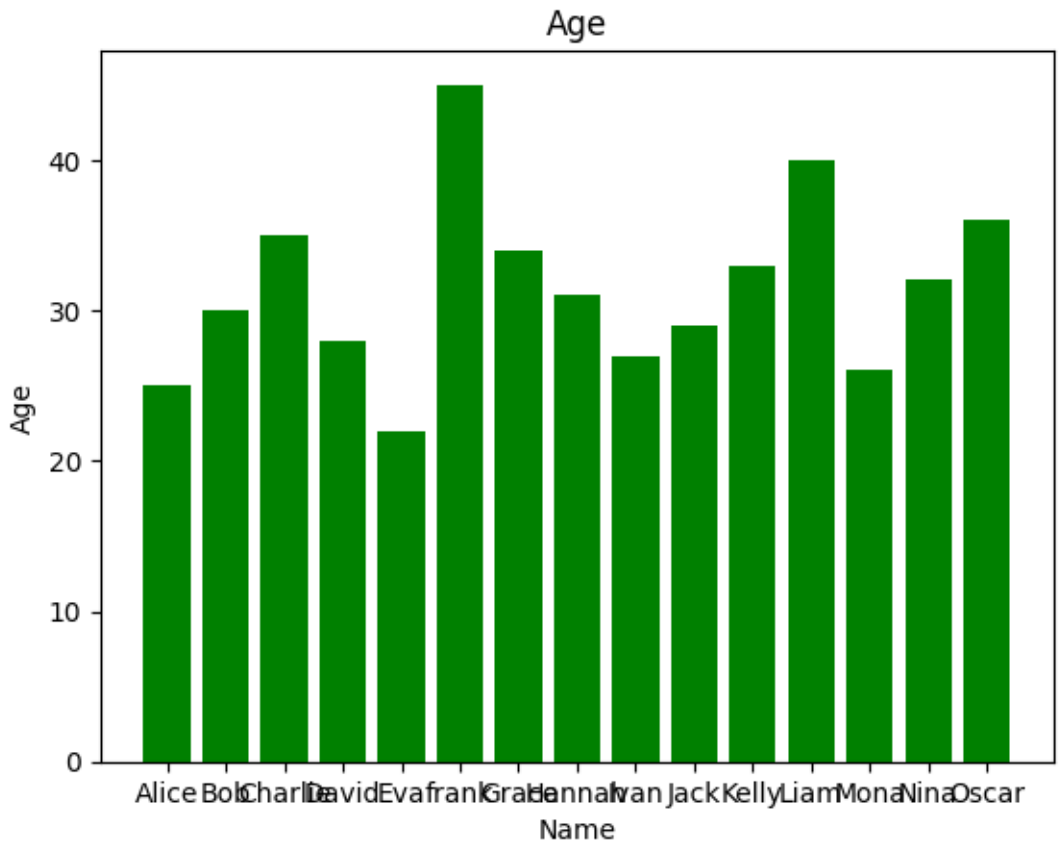
	Age	Salary
0	22	52000

```
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
```

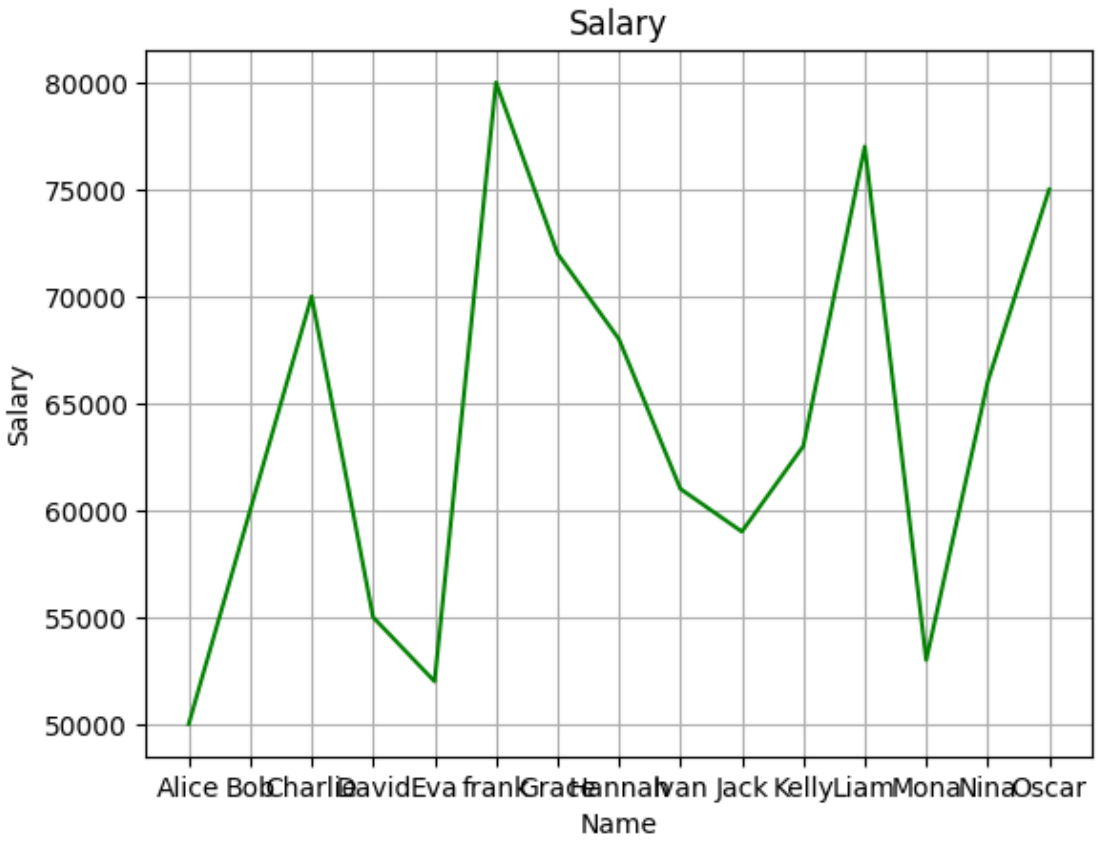
[47]:
```python
# Cau 6:
decrease_df = df.sort_values (by = 'Salary', ascending = False)
print(decrease_df)
```

```
       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
```

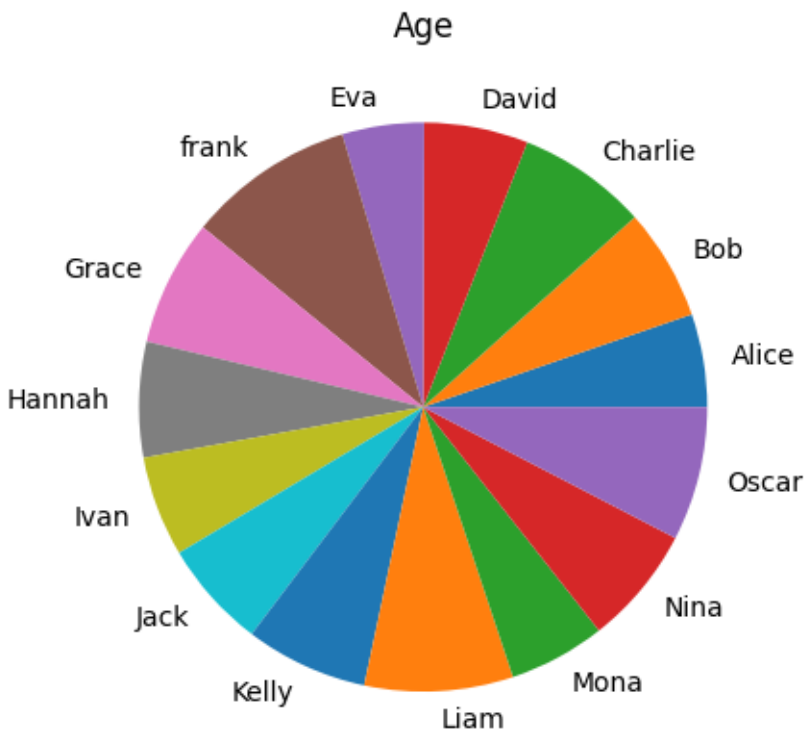
[48]:
```python
# Cau 7:
import matplotlib.pyplot as plt

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



[49]:
```
# Cau 8:
plt.plot(df['Name'], df['Salary'], color = 'green')
plt.title('Salary')
plt.xlabel('Name')
plt.ylabel('Salary')
plt.grid(True)
plt.show()
```

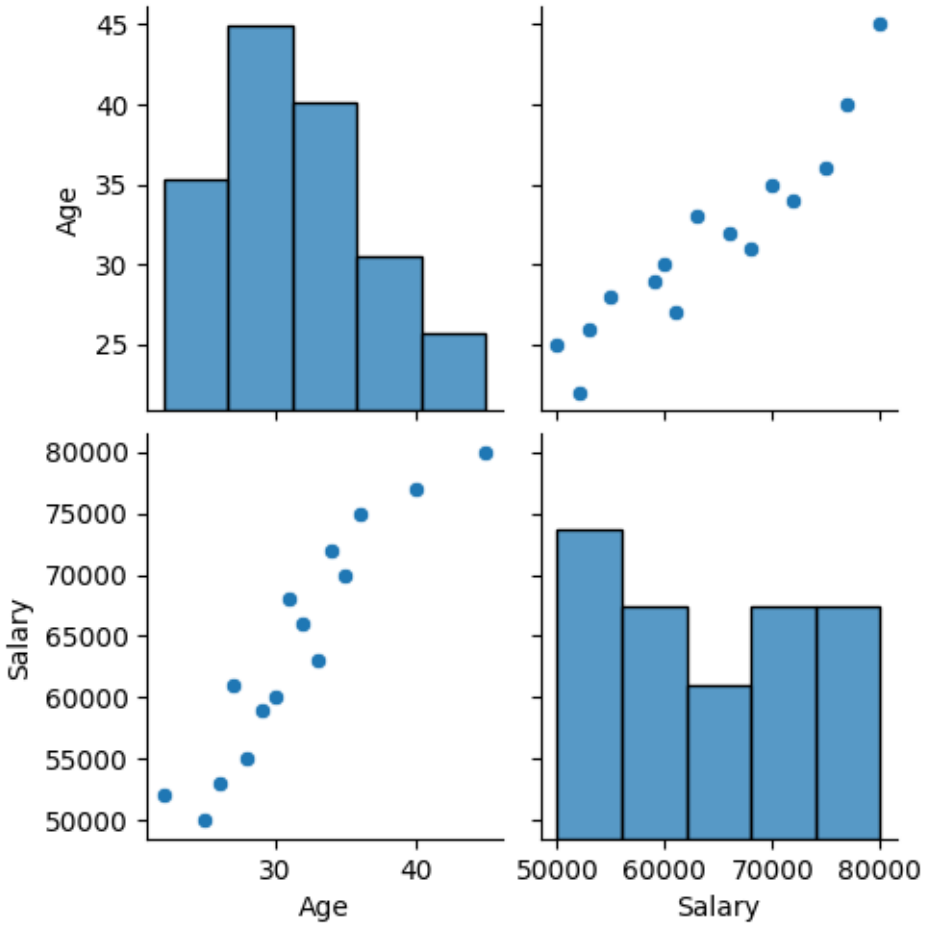


[50]:
```
# Cau 9:
plt.pie(df['Age'], labels=df['Name'])
plt.title('Age')
plt.show()
```



[51]:
```
# Cau 10:
import seaborn as sns

bieudophantan =df[['Age', 'Salary']]
sns.pairplot(bieudophantan)
```

[51]: <seaborn.axisgrid.PairGrid at 0x247b2a38e50>



[52]:
```
# Cau 11:
test = df.isna().sum()
print(test)
```

```
Name      0
Age       0
Salary    0
dtype: int64
```

[53]:
```
# Cau 12:
average_age = df['Age'].mean()
df.loc[df['Age'] > 30, 'Age'] = average_age
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
```

[54]:
```
# Cau 13:
df['Age_normalized'] = (df['Age'] - df['Age'].min()) / (df['Age'].max() -␣
 ↪df['Age'].min())
print(df)
```

```
      Name        Age  Salary  Age_normalized
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
```

[55]:
```
# Cau 14:
def sapxep_age(age):
    if age <= 30:
        return 'young'
    elif 30 < age < 60:
        return 'middle_aged'
    else:
        return 'old'
df['age_group'] = df['Age'].apply(sapxep_age)
print(df)
```

```
      Name        Age  Salary  Age_normalized    age_group
0    Alice  25.000000   50000        0.314685        young
```

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

[56]:
```
# Cau 15:
df['percentage'] = df['Salary'].pct_change() * 100
print(df)
```

```
      Name        Age  Salary  Age_normalized    age_group  percentage
0    Alice  25.000000   50000         0.314685        young         NaN
1      Bob  30.000000   60000         0.839161        young   20.000000
2  Charlie  31.533333   70000         1.000000  middle_aged   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_aged   53.846154
6    Grace  31.533333   72000         1.000000  middle_aged  -10.000000
7   Hannah  31.533333   68000         1.000000  middle_aged   -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_aged    6.779661
11    Liam  31.533333   77000         1.000000  middle_aged   22.222222
12    Mona  26.000000   53000         0.419580        young  -31.168831
13    Nina  31.533333   66000         1.000000  middle_aged   24.528302
14   Oscar  31.533333   75000         1.000000  middle_aged   13.636364
```

[57]:
```
# Cau 16:
df.drop_duplicates(subset=['Name', 'Age', 'Salary'])
print(df)
```

```
      Name        Age  Salary  Age_normalized    age_group  percentage
0    Alice  25.000000   50000         0.314685        young         NaN
1      Bob  30.000000   60000         0.839161        young   20.000000
2  Charlie  31.533333   70000         1.000000  middle_aged   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_aged   53.846154
6    Grace  31.533333   72000         1.000000  middle_aged  -10.000000
```

7	Hannah	31.533333	68000	1.000000	middle_aged	-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_aged	6.779661
11	Liam	31.533333	77000	1.000000	middle_aged	22.222222
12	Mona	26.000000	53000	0.419580	young	-31.168831
13	Nina	31.533333	66000	1.000000	middle_aged	24.528302
14	Oscar	31.533333	75000	1.000000	middle_aged	13.636364

[58]:
```
# Cau 17:
df.to_csv('baikiemtraso1.csv', index=True)
```