

a. create a profile for age x income

Based on scatterplot of age and income, there are clear groupings of age and income:

- age: 18-39 (younger)
 - low income: 0-99,999
 - mid income: 100,000-199,999
 - high income: 200,000-400,000
- age: 40-85 (older)
 - low income: 0-179,999
 - mid income: 180,000-299,999
 - high income: 300,000-600,000

```
# for-loop if-statement: assign criterias
```

```
result = []
```

```
for index, row in opk_mkss.iterrows():
```

```
    income = row['income']
```

```
    age = row['age']
```

```
    if (18 <= age <= 39) and (0 <= income <= 99999):
```

```
        result.append('younger low income')
```

```
    elif (18 <= age <= 39) and (100000 <= income <= 199999):
```

```
        result.append('younger mid income')
```

```
    elif (18 <= age <= 39) and (200000 <= income <= 400000):
```

```
        result.append('younger high income')
```

```
    elif (40 <= age <= 85) and (0 <= income <= 179999):
```

```
        result.append('older low income')
```

```
    elif (40 <= age <= 85) and (180000 <= income <= 299999):
```

```
        result.append('older mid income')
```

```
    elif (40 <= age <= 85) and (300000 <= income <= 600000):
```

```
        result.append('older high income')
```

```
    else:
```

```
        result.append('NA')
```

```
# input result in new column named 'age_income'
```

```
opk_mkss['age_income_status'] = result
```

```
# count of each group
```

```
opk_mkss['age_income_status'].value_counts().sort_index()
```

```
older high income      23125
```

```
older low income      5866911
```

```
older mid income      46377
```

```
younger high income   11589
```

```
younger low income   3020502
```

```
younger mid income    89845
```

```
Name: age_income_status, dtype: int64
```