11/15

trial

June 10, 2024

[ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

[ ]:
```
data = pd.read_excel(io='demo.xlsx')
data
```

[ ]:
```
     Age Gender Marital Status  Address  Income  Income Category  Job Category
0     55      f              1       12      72                3             3
1     56      m              0       29     153                4             3
2     28      f      no answer        9      28                2             1
3     24      m              1        4      26                2             1
4     25      m      no answer        2      23                1             2
..   …    …            …      …     …           …          …
195   45      f              0        3      86                4             3
196   23      f              1        2      27                2             1
197   66      f              1       32      11                1             2
198   49      m              0        4      30                2             1
199   45      m              0        1     147                4             3

[200 rows x 7 columns]
```

[ ]:
```
data.rename(columns={'Age': 'age','Gender':'gender','Marital Status':
 ↪'marital_status','Address':'address',
                     'Income':'income','Income Category':'income_category','Job␣
 ↪Category':'job_category'})
```

[ ]:
```
     age gender marital_status  address  income  income_category  job_category
0     55      f              1       12      72                3             3
1     56      m              0       29     153                4             3
2     28      f      no answer        9      28                2             1
3     24      m              1        4      26                2             1
4     25      m      no answer        2      23                1             2
..   …    …            …      …     …           …          …
195   45      f              0        3      86                4             3
196   23      f              1        2      27                2             1
197   66      f              1       32      11                1             2
```

```
198   49      m              0        4     30                2             1
199   45      m              0        1    147                4             3

[200 rows x 7 columns]
```

```
[ ]: data.columns
```

```
[ ]: Index(['Age', 'Gender', 'Marital Status', 'Address', 'Income',
            'Income Category', 'Job Category'],
           dtype='object')
```

```
[ ]: data.describe()
```

```
[ ]:              Age     Address      Income  Income Category  Job Category
     count  200.000000  200.000000  200.000000       200.000000    200.000000
     mean    42.475000   11.485000   76.305000         2.520000      1.950000
     std     12.801122   10.365665  107.554647         1.065493      0.781379
     min     19.000000    0.000000   11.000000         1.000000      1.000000
     25%     32.000000    3.000000   27.000000         2.000000      1.000000
     50%     43.000000    9.000000   44.500000         2.000000      2.000000
     75%     51.000000   17.000000   76.000000         4.000000      3.000000
     max     76.000000   51.000000  873.000000         4.000000      3.000000
```

```
[ ]: #5. Display some basic statistics about the categorical variables in the dataset
     data.describe(include='object')
```

```
[ ]:        Gender  Marital Status
     count     200             200
     unique      4               3
     top         f               0
     freq       99             102
```

```
[ ]: #6.What are the unique observations under gender?
     data['Gender'].unique()
```

```
[ ]: array(['f', 'm', '  f', '   m'], dtype=object)
```

```
[ ]: #7.Can you fix any problems observed under the gender, give brief explanations␣
       ↪why and how
```

```
[ ]: #8.How many observations have ‘no answer’ for marital status?
     data['Marital Status'].value_counts()
     # they are 5
```

```
[ ]: Marital Status
     0             102
     1              93
```

```
no answer      5
Name: count, dtype: int64
```

[ ]:

```
#9.Write some piece of code to return only numeric variables from the dataset
data.select_dtypes(include=[int,float])
```

✓

[ ]:

```
     Age  Address  Income  Income Category  Job Category
0     55       12      72                3             3
1     56       29     153                4             3
2     28        9      28                2             1
3     24        4      26                2             1
4     25        2      23                1             2
..   ...      ...     ...              ...           ...
195   45        3      86                4             3
196   23        2      27                2             1
197   66       32      11                1             2
198   49        4      30                2             1
199   45        1     147                4             3

[200 rows x 5 columns]
```

[ ]:

```
#10.Are there any missing values in the dataset?
data.isna().sum()
#No there are no missing values
```

✓

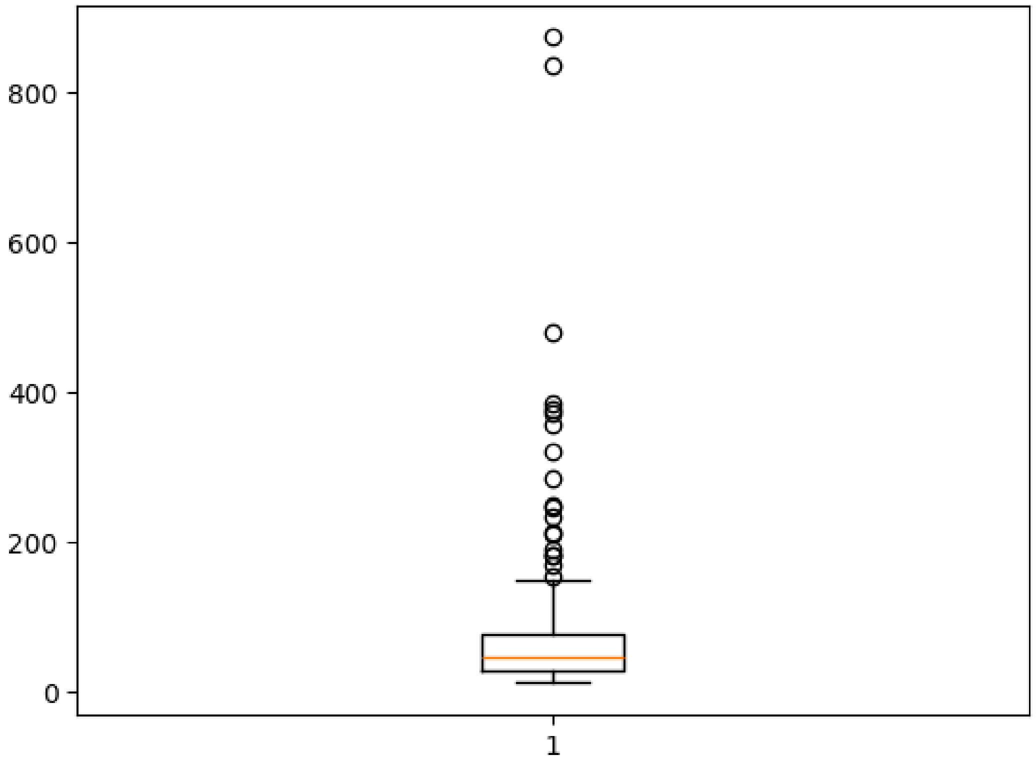
[ ]:

```
Age                0
Gender             0
Marital Status     0
Address            0
Income             0
Income Category    0
Job Category       0
dtype: int64
```

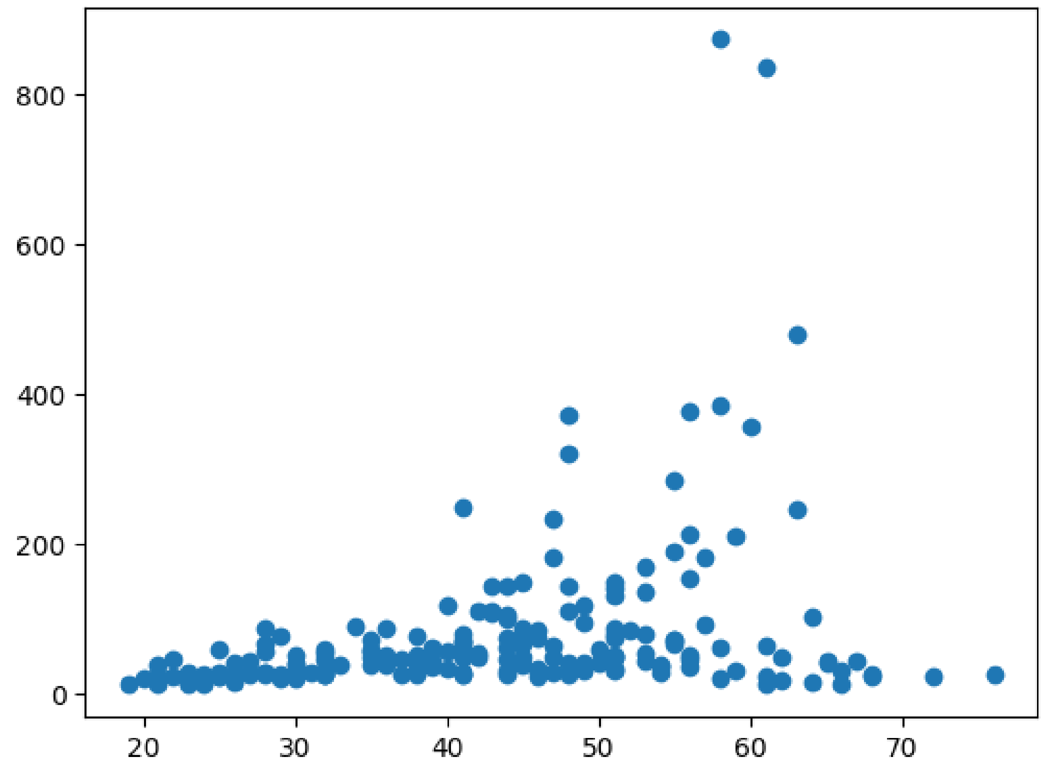
[ ]:

```
#11.Are there any outliers in the income variable?
plt.boxplot(data.Income)
plt.show()
# Yes they are there
```

✓



```
[ ]: #12.Investigate the relationship between age and income
     plt.scatter(data.Age,data.Income)
     plt.show()
     #There is no relationship
```



```
[ ]: #13.How many people earn more than 300 units?
```

```
[ ]: #14.What data type is the marital status?
     type(data['Marital Status'])
```

```
[ ]: pandas.core.series.Series
```

```
[ ]: #15.Create dummy variables for gender
```