Experiment 2

Aim -

Data Visualization/ Exploratory Data Analysis using Matplotlib and Seaborn.

Todo -

1. Create a bar graph, and contingency table using any 2 features.
2. Plot Scatter plot, box plot, and Heatmap using seaborn.
3. Create a histogram and normalized Histogram.
4. Describe what this graph and table indicate.
5. Handle outlier using box plot and Interquartile range.

Dataset -

The dataset contains data about customers' purchases during the Black Friday sale. This dataset was taken from Kaggle. The dataset has 550k rows and 12 columns. The various columns of the dataset are age, marital status, gender, total purchase amount, and many other features.

Theory -

Matplotlib -

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram, etc.

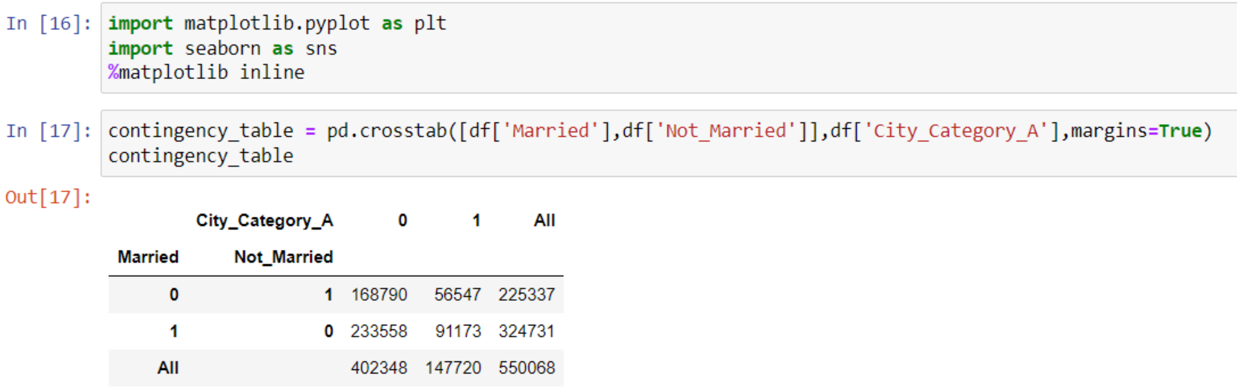
Seaborn -

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on top of the matplotlib library and is also closely integrated with the data structures from pandas.

Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs so that we can switch between different visual representations for the same variables for a better understanding of the dataset.

Results -

1. Create a bar graph, and contingency table using any 2 features.

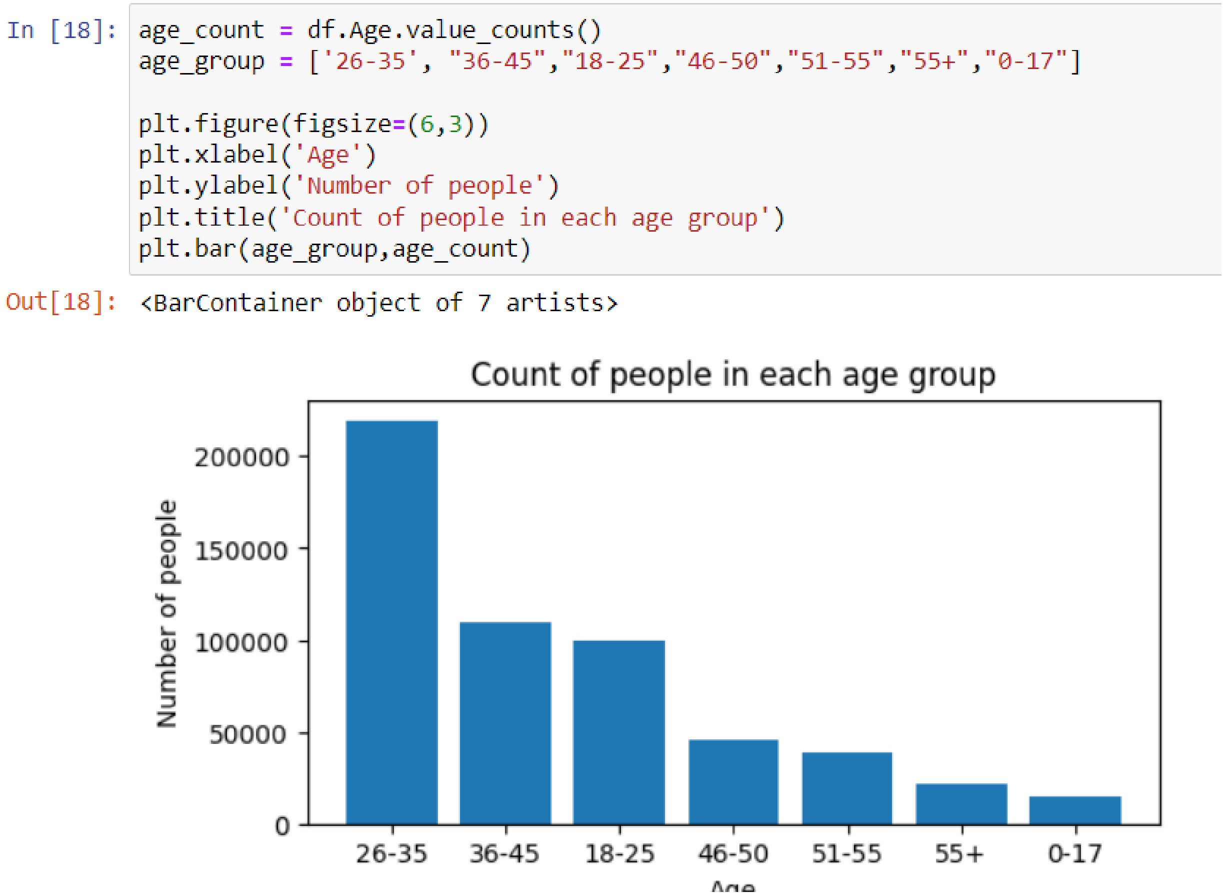


```
In [16]: import matplotlib.pyplot as plt
         import seaborn as sns
         %matplotlib inline
```

```
In [17]: contingency_table = pd.crosstab([df['Married'],df['Not_Married']],df['City_Category_A'],margins=True)
         contingency_table
```

Out[17]:

	City_Category_A	0	1	All
Married	Not_Married			
0	1	168790	56547	225337
1	0	233558	91173	324731
All		402348	147720	550068

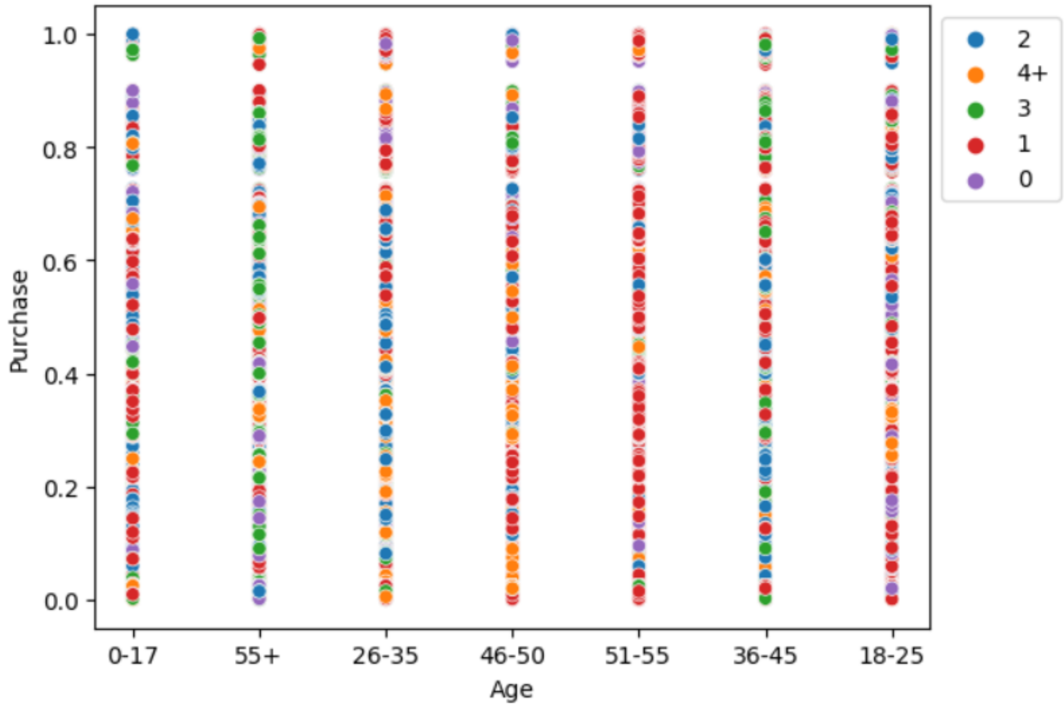


```
In [18]: age_count = df.Age.value_counts()
         age_group = ['26-35', "36-45","18-25","46-50","51-55","55+","0-17"]

         plt.figure(figsize=(6,3))
         plt.xlabel('Age')
         plt.ylabel('Number of people')
         plt.title('Count of people in each age group')
         plt.bar(age_group,age_count)

Out[18]: <BarContainer object of 7 artists>
```

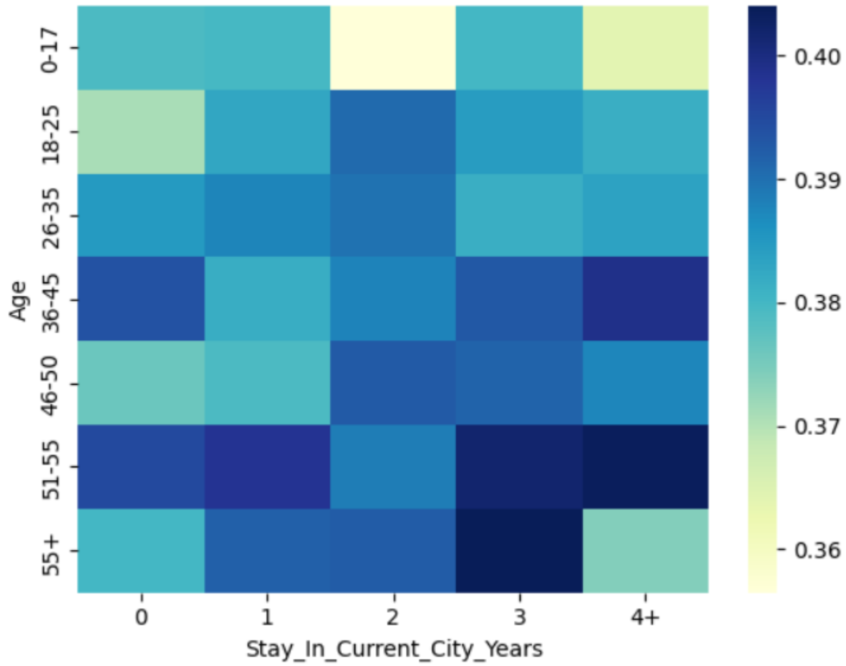
2. Plot Scatter plot, box plot, and Heatmap using seaborn.

```
In [22]: sns.scatterplot(data=df,x='Age',y='Purchase',hue='Stay_In_Current_City_Years')
         plt.legend(bbox_to_anchor=(1,1))
```

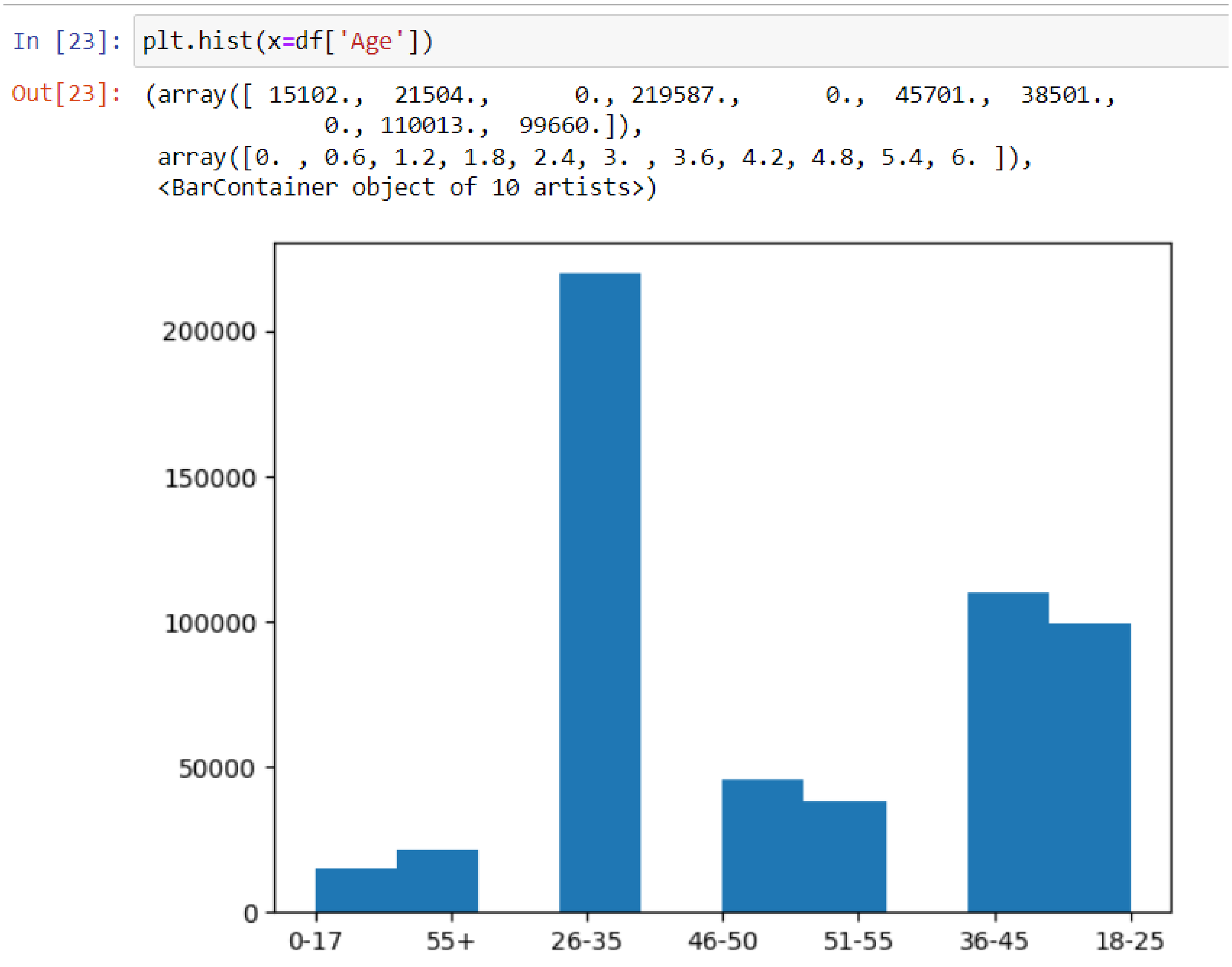
```
Out[22]: <matplotlib.legend.Legend at 0x26f0c3afbe0>
```



```
In [19]: heat = df.pivot_table(index='Age',columns = 'Stay_In_Current_City_Years',values='Purchase')
         sns.heatmap(heat,cmap="YlGnBu")
         plt.show()
```

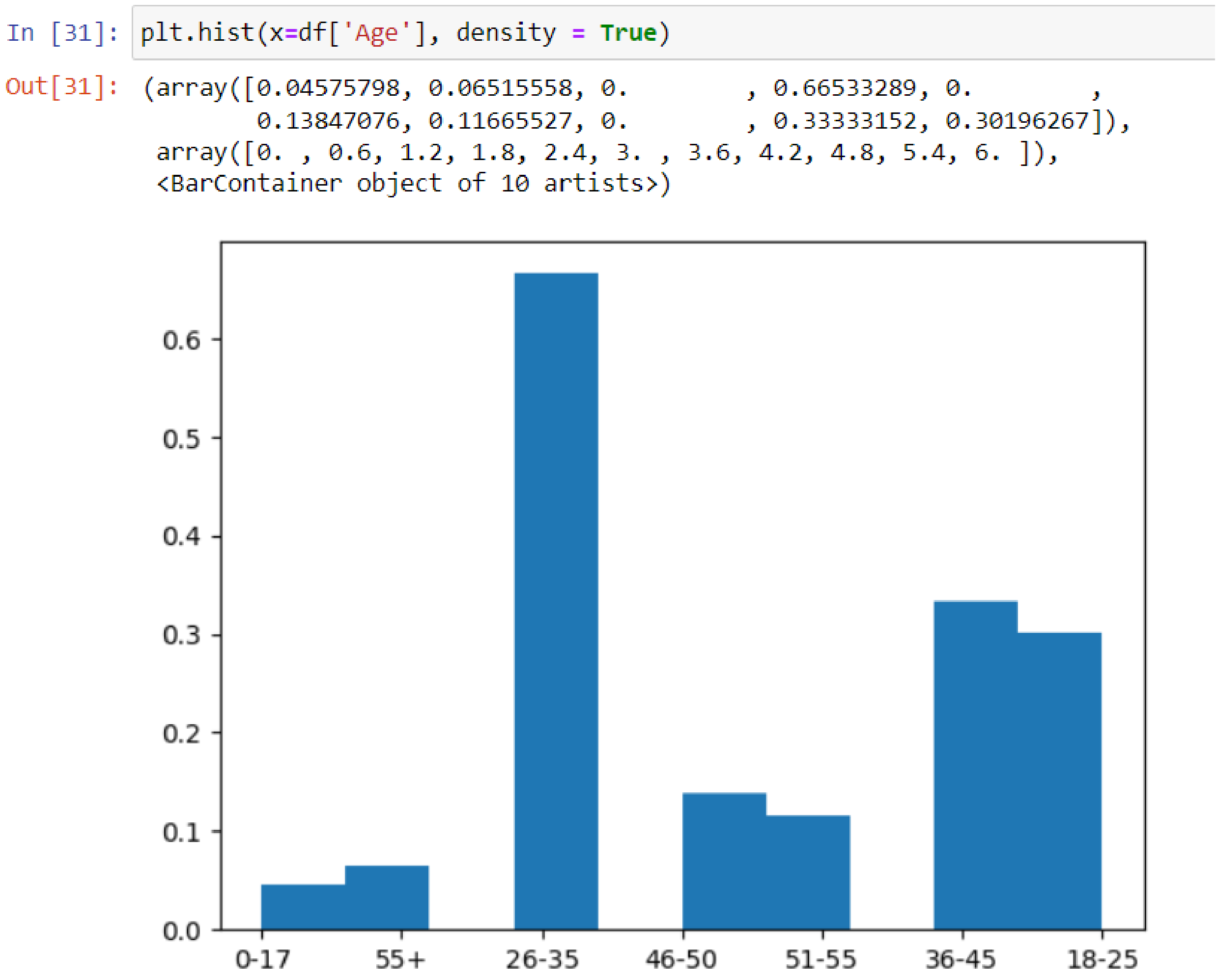


3. Create a histogram and normalized Histogram.



```
In [23]: plt.hist(x=df['Age'])
```

```
Out[23]: (array([ 15102.,  21504.,      0., 219587.,      0.,  45701.,  38501.,
              0., 110013.,  99660.]),
          array([0. , 0.6, 1.2, 1.8, 2.4, 3. , 3.6, 4.2, 4.8, 5.4, 6. ]),
          <BarContainer object of 10 artists>)
```

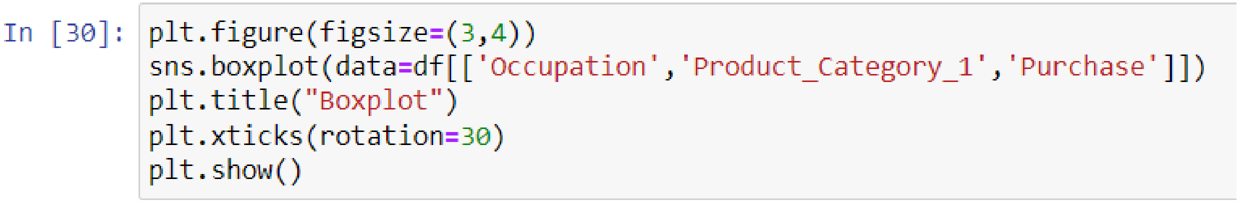


```
In [31]: plt.hist(x=df['Age'], density = True)
```

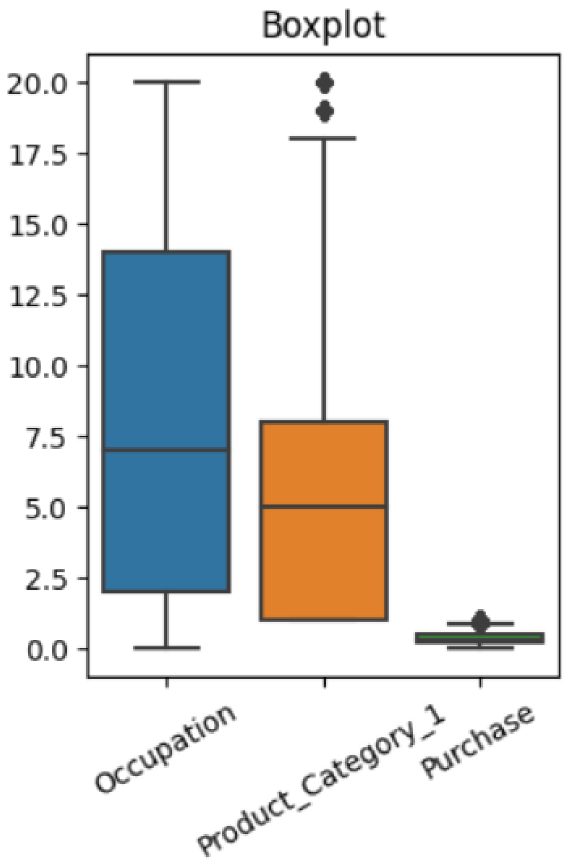
```
Out[31]: (array([0.04575798, 0.06515558, 0.        , 0.66533289, 0.        ,
                 0.13847076, 0.11665527, 0.        , 0.33333152, 0.30196267]),
          array([0. , 0.6, 1.2, 1.8, 2.4, 3. , 3.6, 4.2, 4.8, 5.4, 6. ]),
          <BarContainer object of 10 artists>)
```

4. Describe what this graph and table indicate.
   This graph represents the normalized age group. The age group is divided into bins(represented as bars) and each bin has the count of data points that fall in the bin's range.

5. Handle outlier using box plot and Interquartile range.



```
In [30]: plt.figure(figsize=(3,4))
         sns.boxplot(data=df[['Occupation','Product_Category_1','Purchase']])
         plt.title("Boxplot")
         plt.xticks(rotation=30)
         plt.show()
```



Conclusion-

We have successfully visualized the data causing matplotlib and seaborn. We made different types of graphs to get insights about the data and find pattern and link between attributes.