

vishaulization

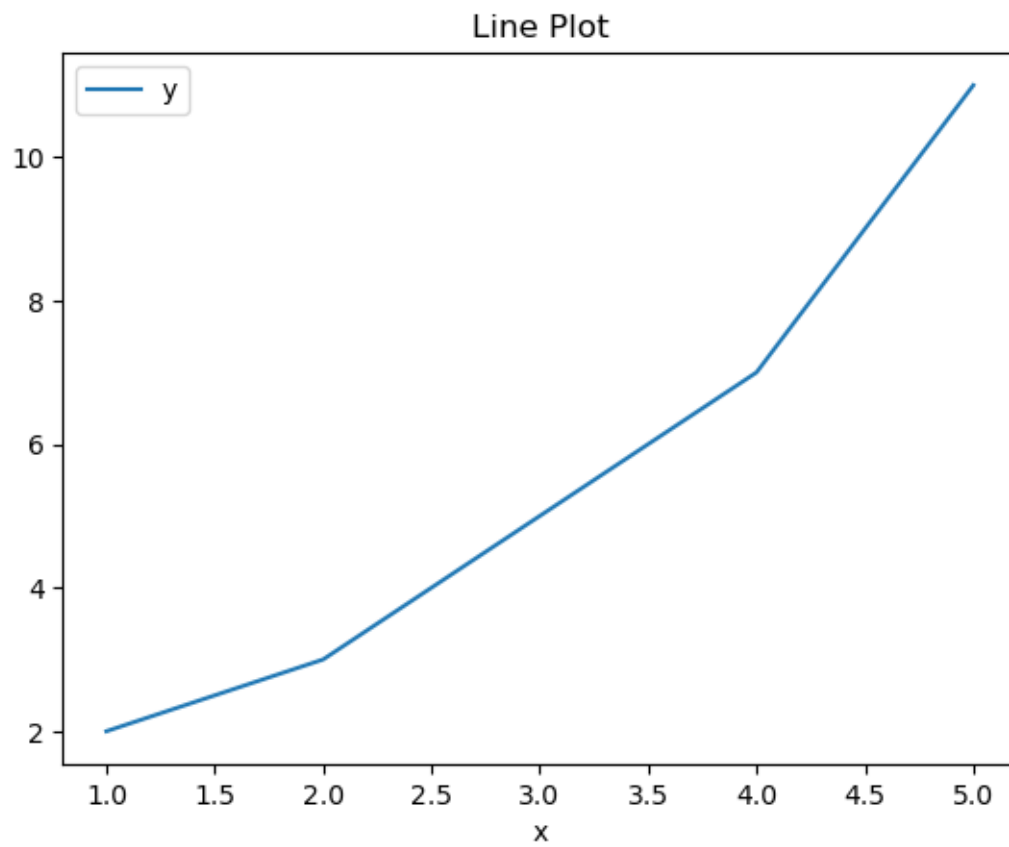
January 12, 2024

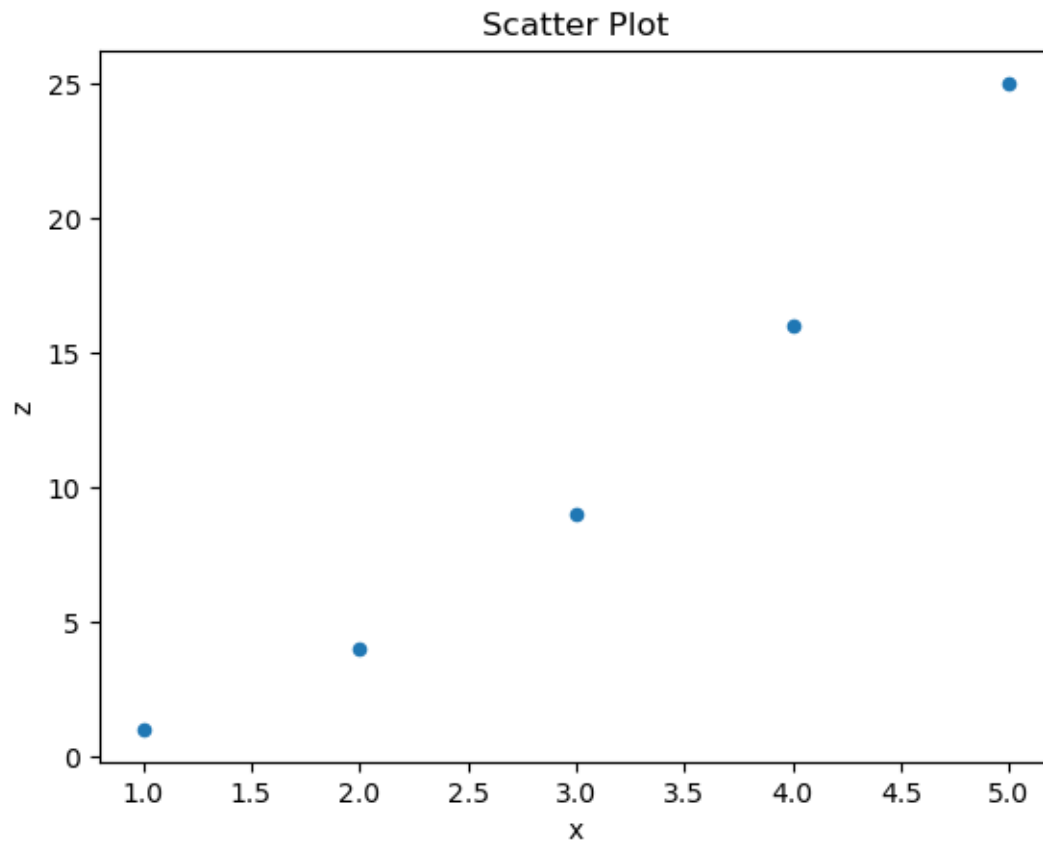
```
[ ]: 08.1.2024 EX-3 Data Visualization through Python
```

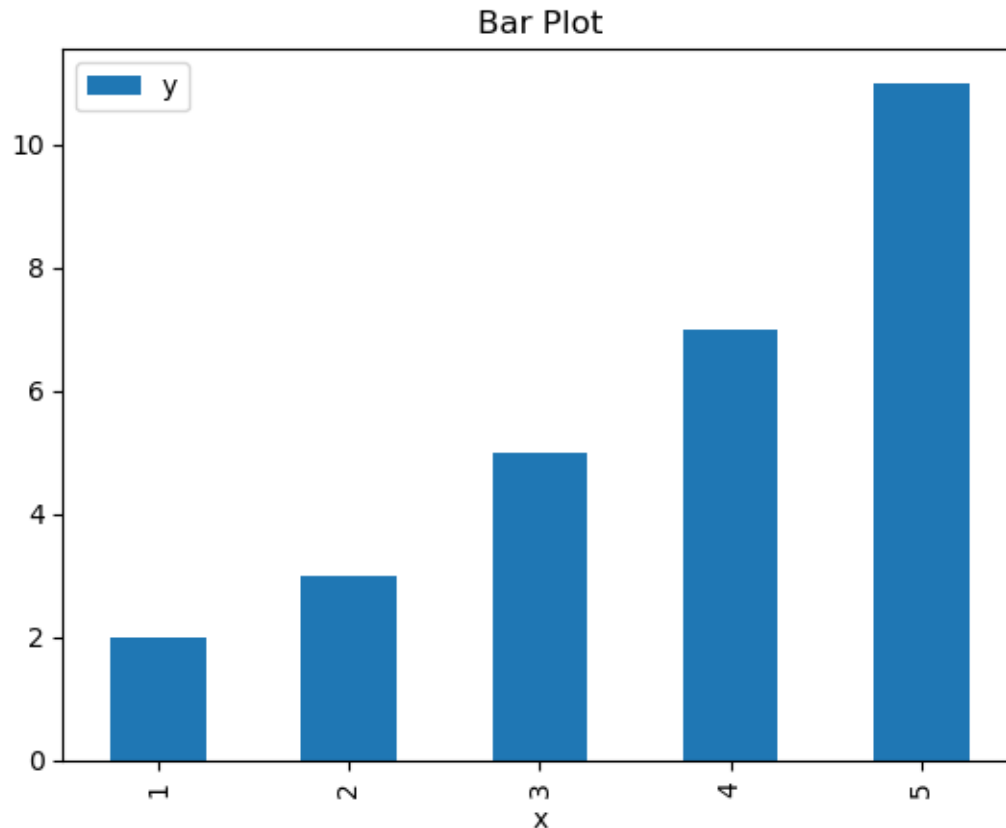
```
[ ]: '''  
URK22AI1048  
HARIHARAN K
```

```
[1]: #COMMON QUESTION URK22AI1048  
import pandas as pd  
data = {  
    'x': [1, 2, 3, 4, 5],  
    'y': [2, 3, 5, 7, 11],  
    'z': [1, 4, 9, 16, 25]  
}  
  
df = pd.DataFrame(data)  
df.plot(x='x', y='y', kind='line', title='Line Plot')  
df.plot(x='x', y='z', kind='scatter', title='Scatter Plot')  
df.plot(x='x', y='y', kind='bar', title='Bar Plot')
```

```
[1]: <Axes: title={'center': 'Bar Plot'}, xlabel='x'>
```







```
[ ]: '''  
Aim  
Write a python program to demonstrate the data visualization for data science_  
    ↪application.  
  
Description  
Data visualization is the process of translating large data sets and metrics_  
    ↪into charts,  
graphs and other visuals. The resulting visual representation of data makes it_  
    ↪easier to identify  
and share real-time trends, outliers, and new insights about the information_  
    ↪represented in the  
data.  
  
Matplotlib  
Matplotlib is an easy-to-use, low-level data visualization library that is_  
    ↪built on  
NumPy arrays.
```

```
import pandas as pd
import matplotlib.pyplot as plt
```

Dot Plots

These plots provide a visual representation of a function $y = f(x)$ defined by a set of (x,y) points. Dot plots just show the data points

Line Plot

Line Chart is used to represent a relationship between two data X and Y on a different axis. It is plotted using the plot () function.

```
plt.plot(x, y, linewidth=0, marker='o', color='lightblue');
```

Bar Plot and Pie Chart

A bar plot and Pie Chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent.

```
plt.bar(y, x, color=['red','blue','orange'])
plt.pie(x, labels = y, autopct='%1.2f%%');
```

Scatter Plot

Scatter plots are used to observe relationships between variables and uses dots to represent the relationship between them. The scatter () method in the matplotlib library is used to draw a scatter plot.

```
plt.scatter(x, y, c='red');
```

Histograms

Histograms are an accurate representation of frequency distribution of numerical data; it relates only one variable. Divide the entire range of values into a series of intervals(horizontal) and then count or frequency (vertical) how many values fall in to each interval.

```
plt.hist(x, color='red', edgecolor='white', bins=5)
```

Box and whisker plots

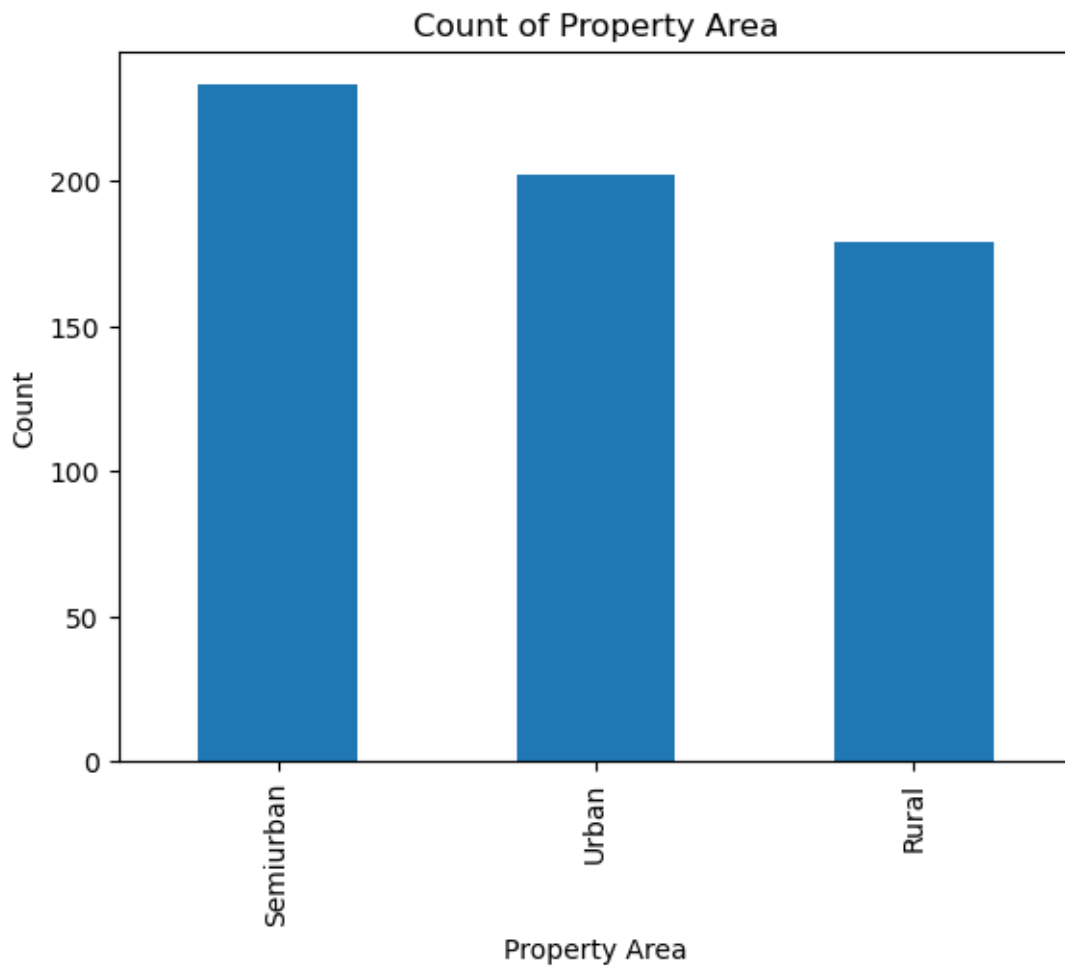
A box and whisker plot-also called a box plot-displays the five-number summary of data. The five-number summary is the minimum, first quartile (25%), median (50%), third quartile (75%), and maximum.

```
Df [['&#39;Column name&#39;]]. boxplot ()
```

```
[10]: #URK22AI1048 1q
#1. Draw a bar chart with Property_area and its count
import pandas as pd
import matplotlib.pyplot as plt

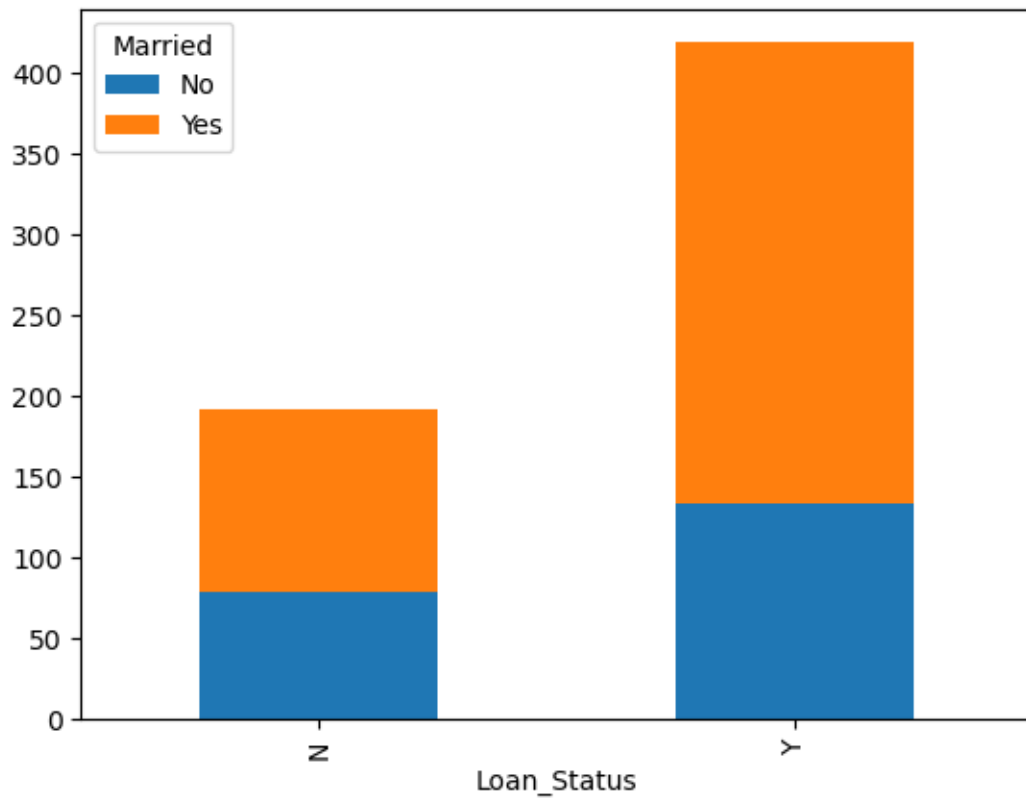
file = "loan.csv"
df = pd.read_csv(file)

df['Property_Area'].value_counts().plot(kind='bar')
plt.xlabel('Property Area')
plt.ylabel('Count')
plt.title('Count of Property Area')
plt.show()
```

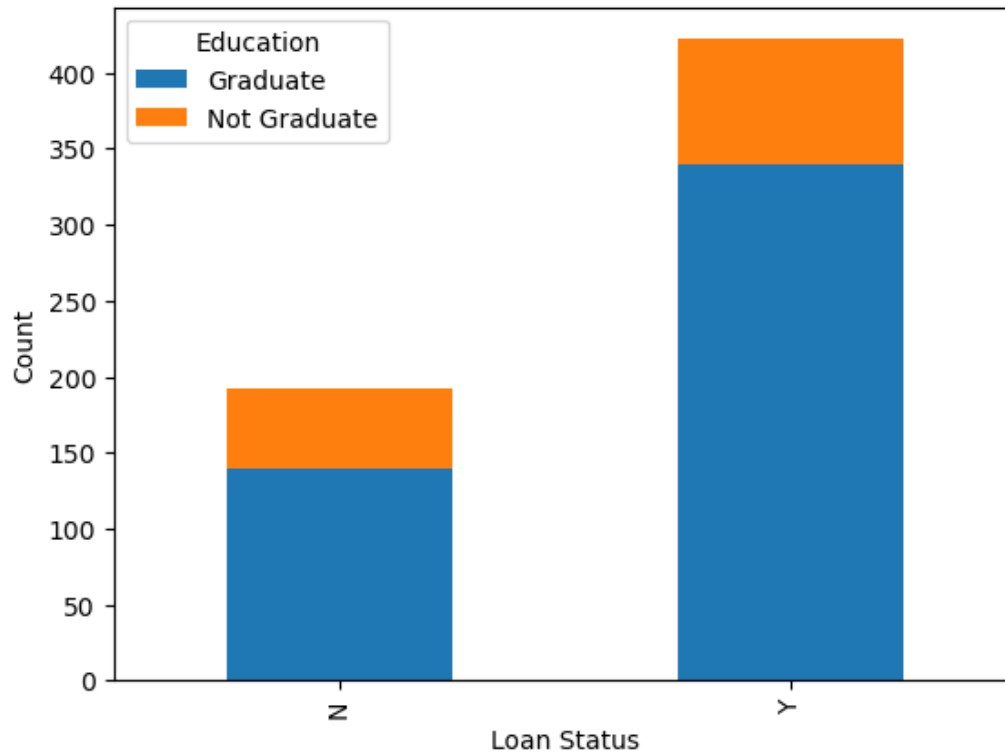


```
[13]: #URK22AI1048 2q
#2. Draw a comparative bar chart for "Graduate" and "Married" columns against
↳Loan_status.

pd.crosstab(df['Loan_Status'], df['Married']).plot(kind='bar', stacked=True)
pd.crosstab(df['Loan_Status'], df['Education']).plot(kind='bar', stacked=True)
plt.xlabel('Loan Status')
plt.ylabel('Count')
plt.title('Comparative Bar Chart of Loan Status by Marital Status and
↳Education')
plt.show()
```

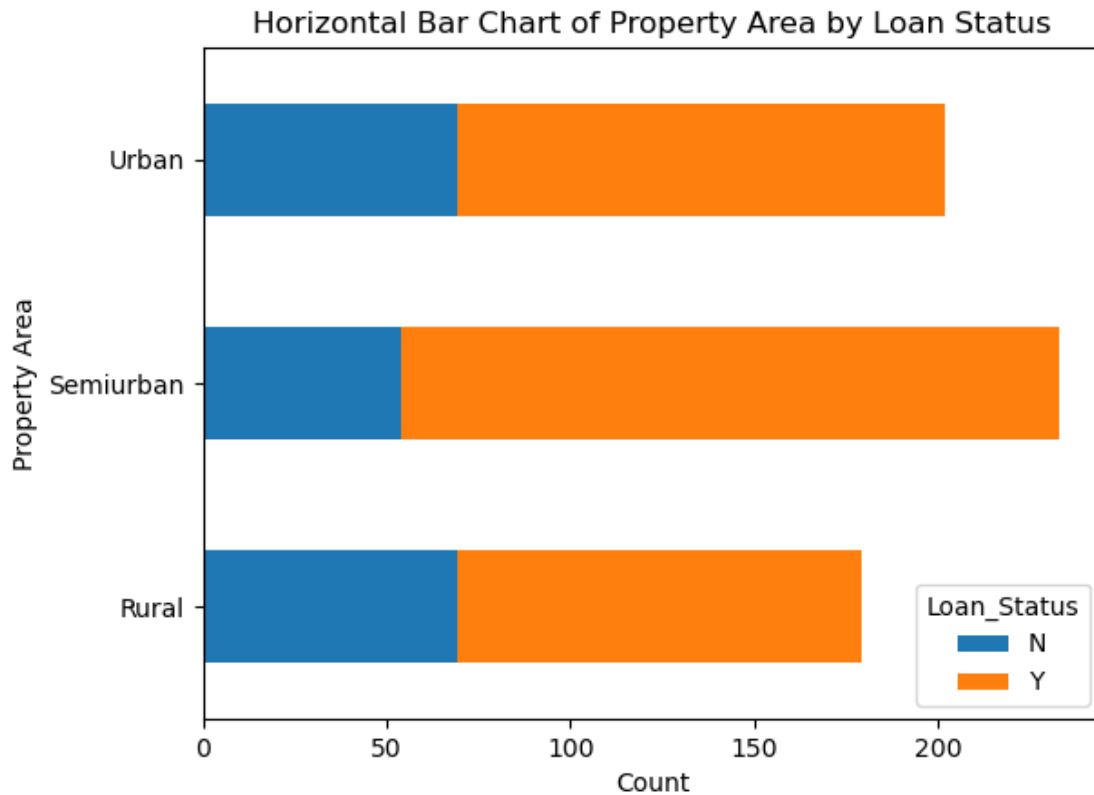


Comparative Bar Chart of Loan Status by Marital Status and Education



```
[17]: #URK22AI1048 3q
#3. Draw a horizontal bar chart for "Property_area" and "Loan_status"

pd.crosstab(df['Property_Area'], df['Loan_Status']).plot(kind='barh',
    ↪stacked=True)
plt.xlabel('Count')
plt.ylabel('Property Area')
plt.title('Horizontal Bar Chart of Property Area by Loan Status')
plt.show()
```

```
[22]: #urk22ai1048 4q
# Draw a stacked bar chart for Applicant_Income and Coapplicant Income against
↳ the Loan_status

import pandas as pd
import matplotlib.pyplot as plt

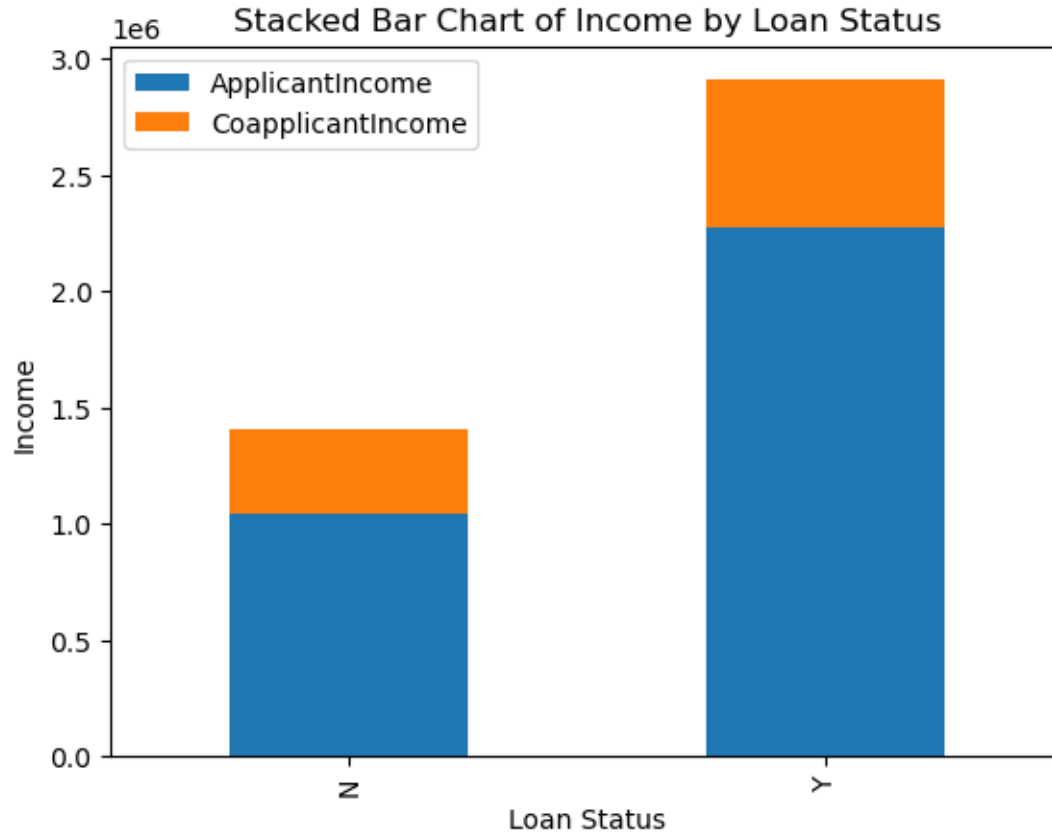
df = pd.read_csv("loan.csv")

# Create the stacked bar chart
df.groupby('Loan_Status')['ApplicantIncome', 'CoapplicantIncome'].sum().
↳ plot(kind='bar', stacked=True)
plt.xlabel('Loan Status')
plt.ylabel('Income')
plt.title('Stacked Bar Chart of Income by Loan Status')
plt.show()
```

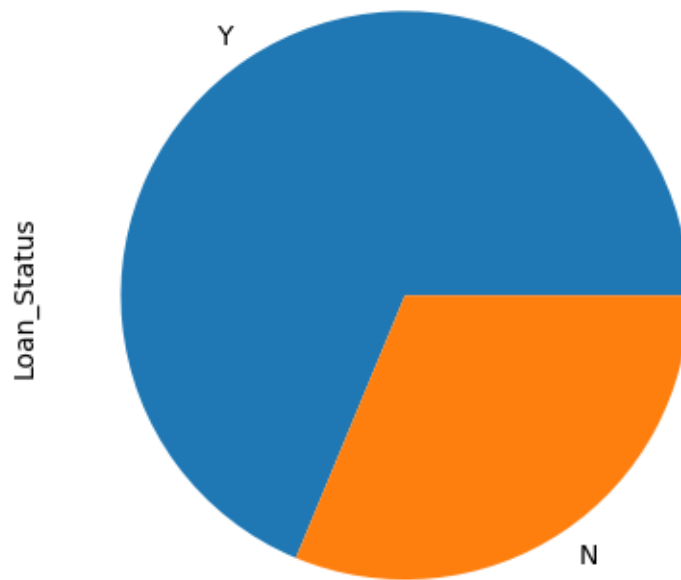
/tmp/ipykernel_2012047/3131581000.py:11: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

```
df.groupby('Loan_Status')['ApplicantIncome',
```

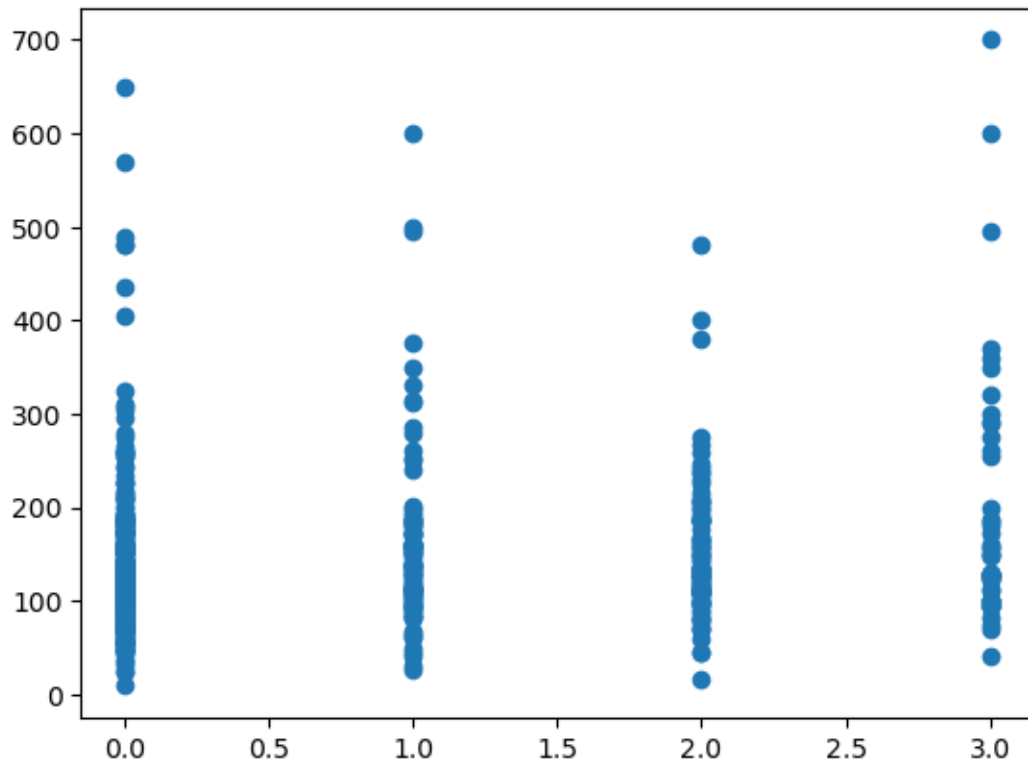
```
'CoapplicantIncome'].sum().plot(kind='bar', stacked=True)
```



```
[30]: #URK22AI1048 5Q
plt.plot(kind='pie')
loan= df['Loan_Status'].value_counts().plot(kind='pie')
plt.show(loan)
```



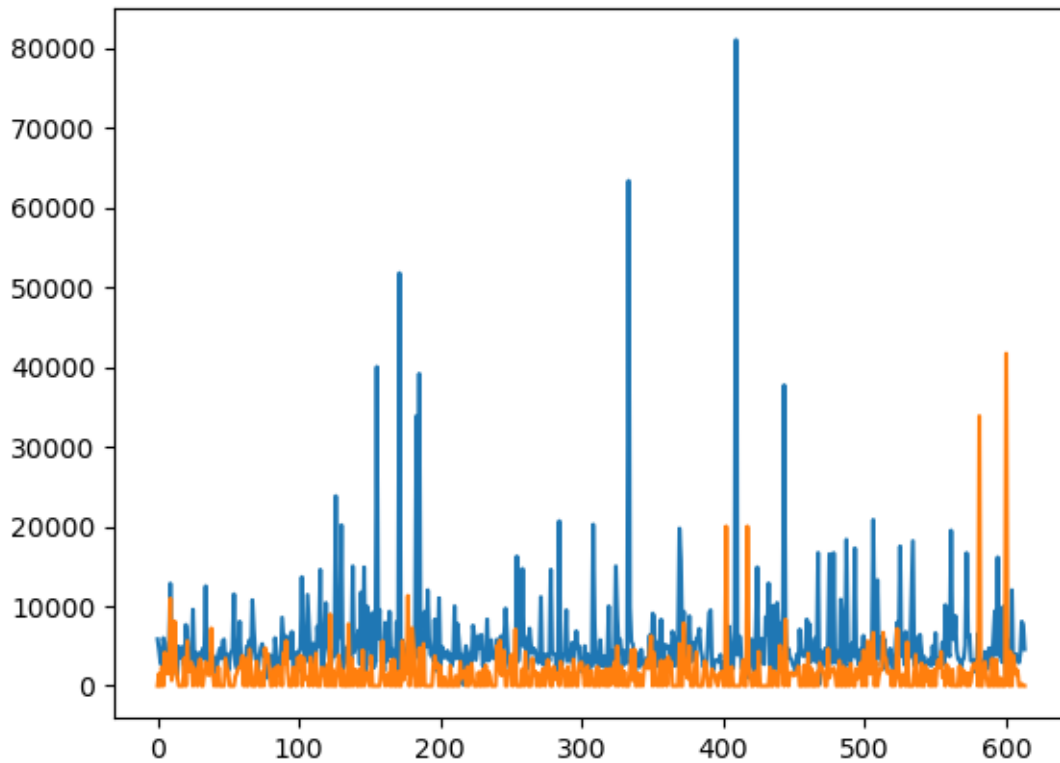
```
[41]: #urk22ai0148 6Q  
      #Draw the dot plot between Gender and Loan_amount  
      plt.scatter(df['Dependents'], df['LoanAmount'])  
      plt.show()
```



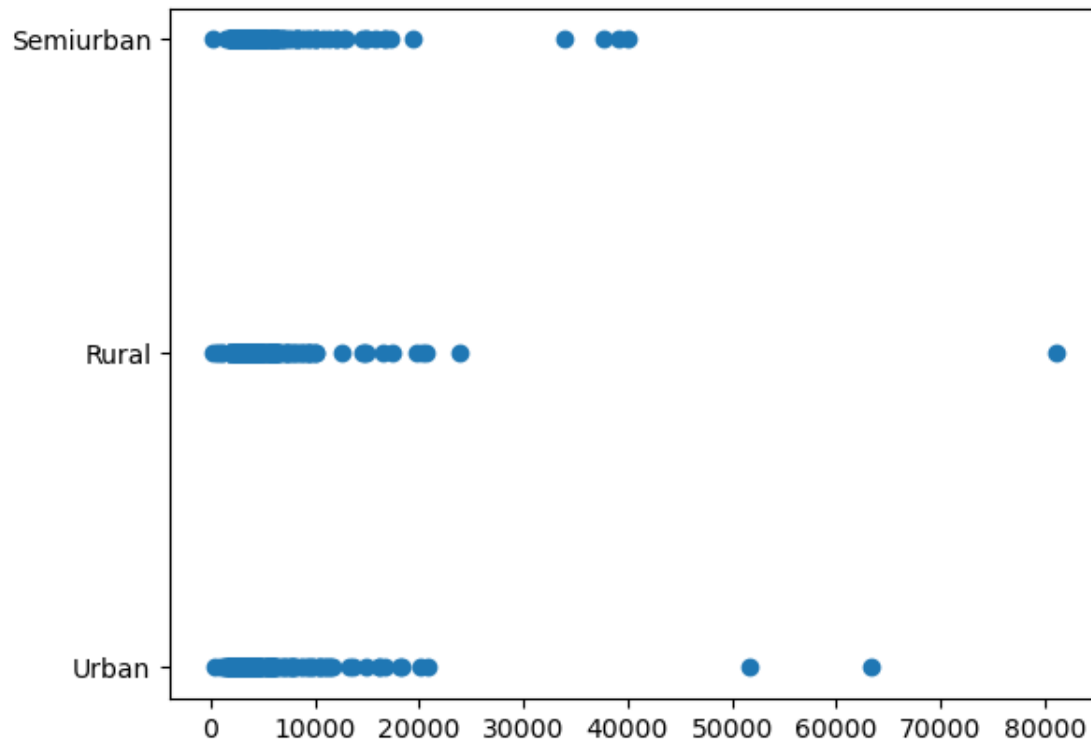
```
[44]: #urk22ai1048 7Q
#Draw the line plot between ApplicantIncome and coapplicantIncome. Observe the
↪ trend line.

plt.plot(df['ApplicantIncome'])
plt.plot(df['CoapplicantIncome'])

plt.show()
```

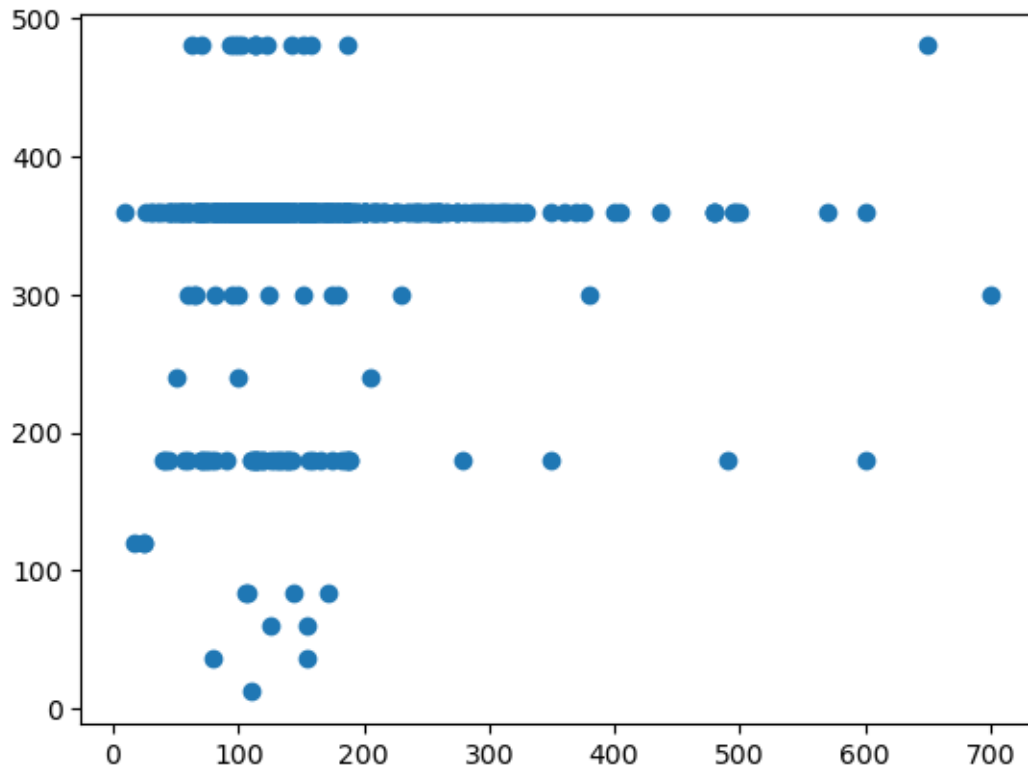


```
[45]: #urk22ai1048 8Q
#Draw the scatter plot between ApplicantIncome and Property_Area. Observe the
↪ correlation
plt.scatter(df['ApplicantIncome'], df['Property_Area'])
plt.show()
```



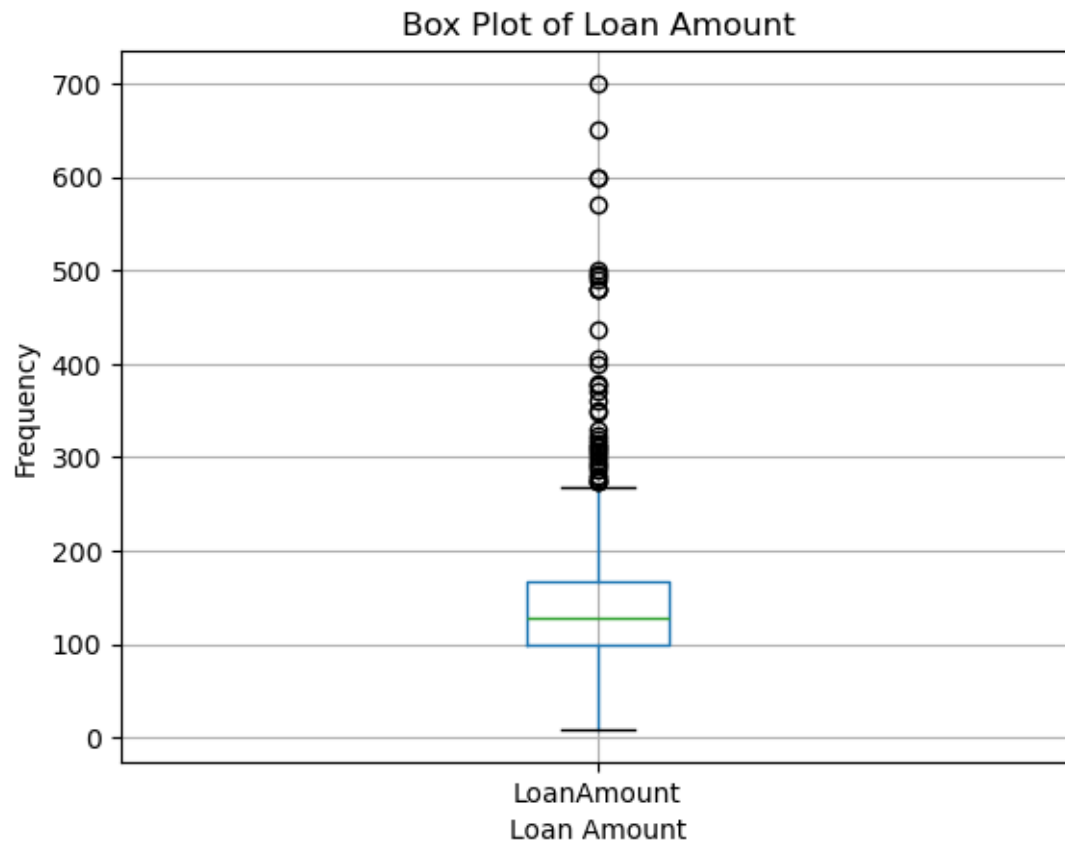
```
[48]: #urk22ai1048 9Q
#Draw the scatter plot between Loan_Amount and Loan_Term. Observe the
↳correlation

plt.scatter(df['LoanAmount'], df['Loan_Amount_Term'])
plt.show()
```



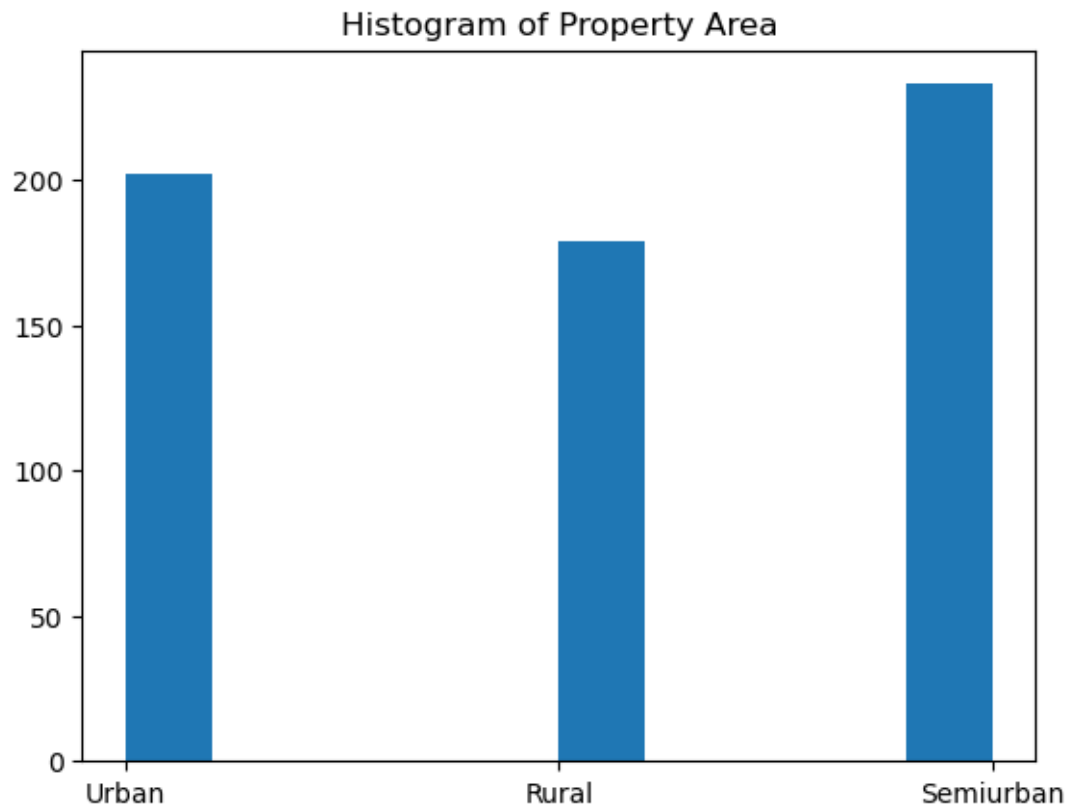
```
[19]: #URK22AI1048 10Q
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_csv('loan.csv')
df[['LoanAmount']].boxplot ()
plt.title('Box Plot of Loan Amount')
plt.xlabel('Loan Amount')
plt.ylabel('Frequency')
plt.show()
```

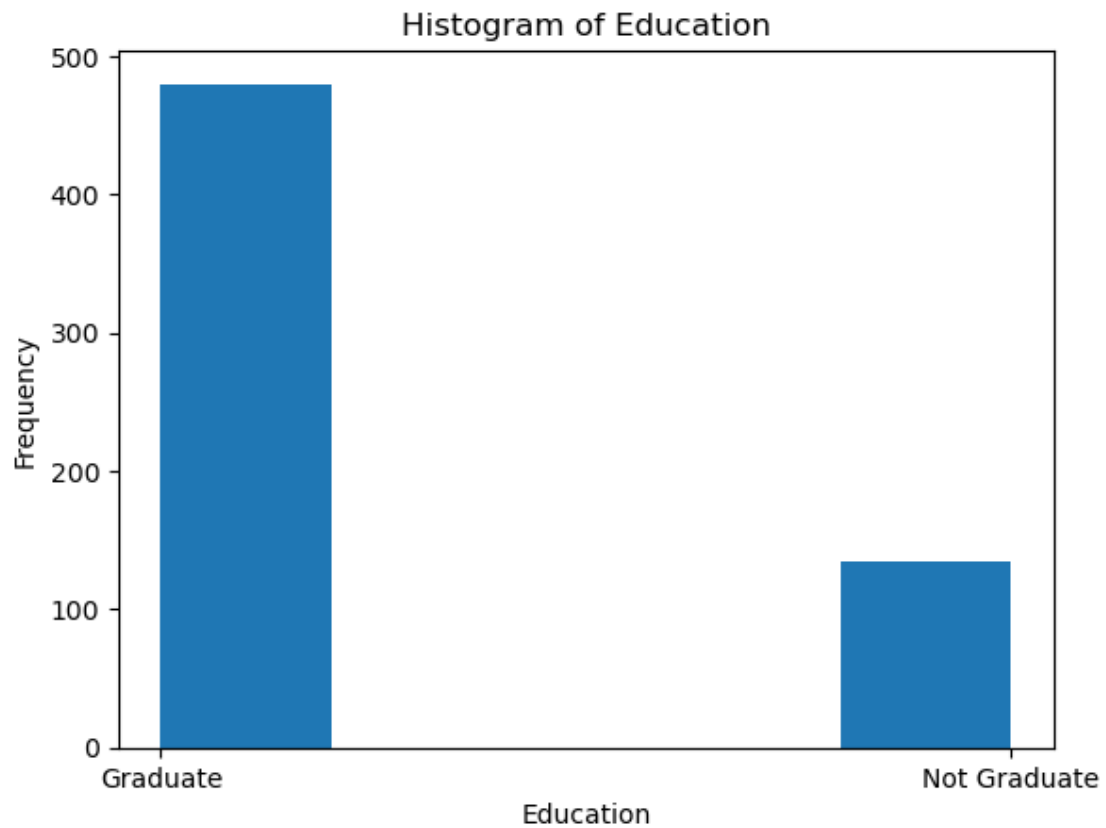


```
[52]: #urk22ai1048 11Q
#11.Draw the histogram plot for Property_Area column

plt.hist(df['Property_Area'], bins=10)
plt.title('Histogram of Property Area')
plt.show()
```

```
[58]: #urk22ai1048 12Q
plt.hist(df['Education'], bins=5)
plt.xlabel('Education')
plt.ylabel('Frequency')
plt.title('Histogram of Education')
plt.show()
```



```
[ ]: '''  
    RESULT  
    The above programs were created and executed successfully.  
    '''
```