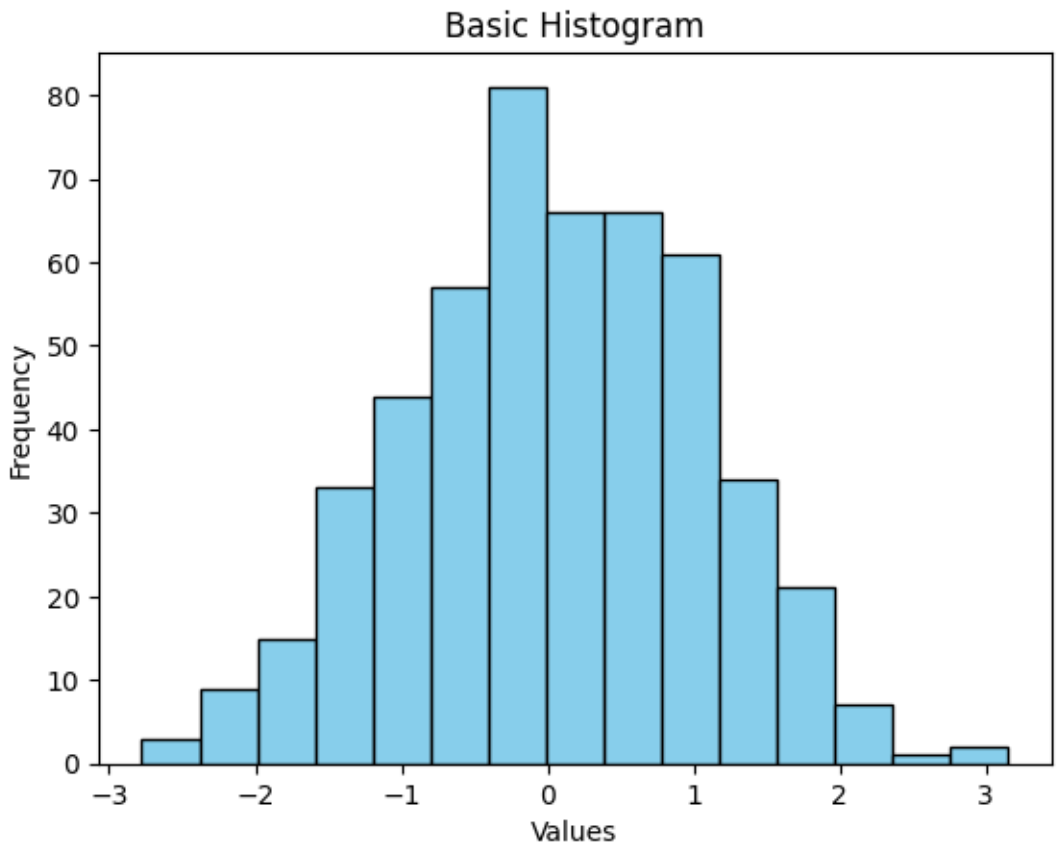
# lab1-datamining-211220051

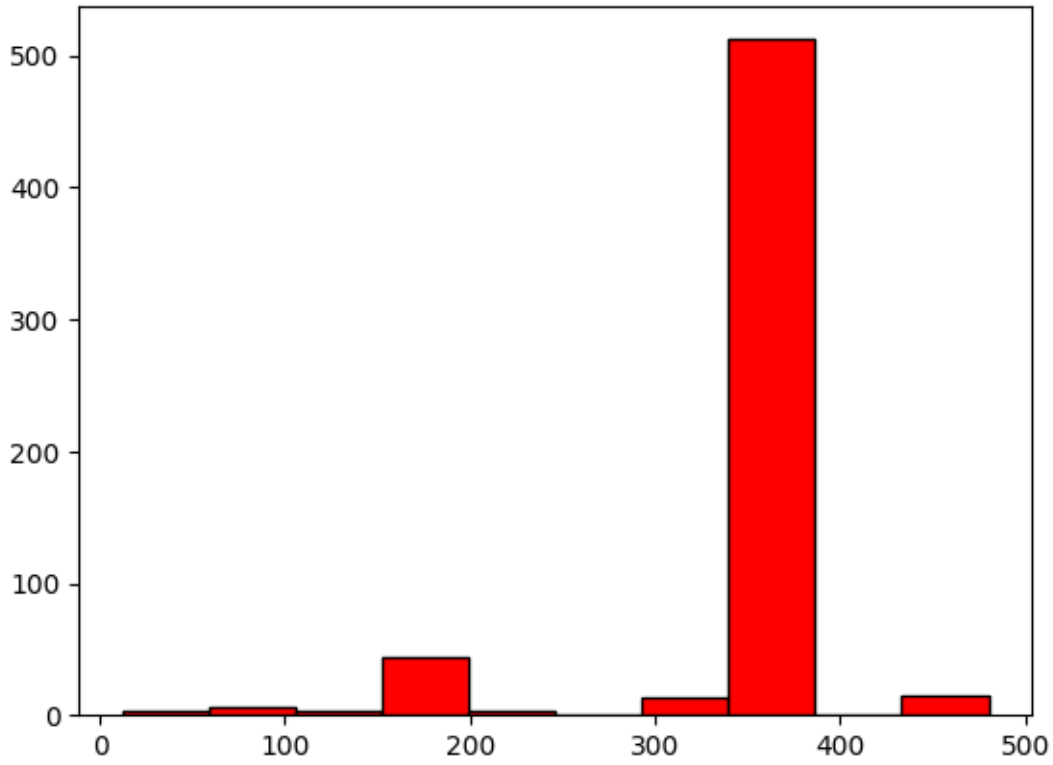
April 29, 2024

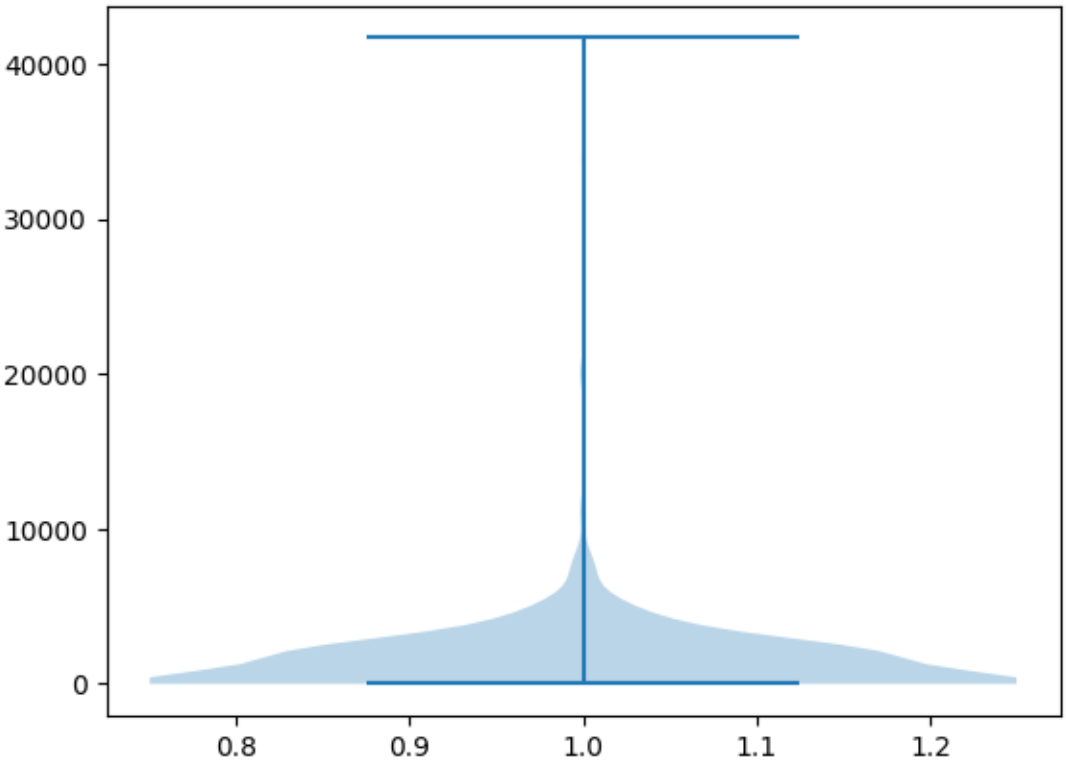
[ ]:
```
import matplotlib.pyplot as plt
import numpy as np

# Generate random data for the histogram
data = np.random.randn(500)
# print(data)
# Plotting a basic histogram
plt.hist(data, bins=15, color='skyblue', edgecolor='black')

# Adding labels and title
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.title('Basic Histogram')

# Display the plot
plt.show()
```

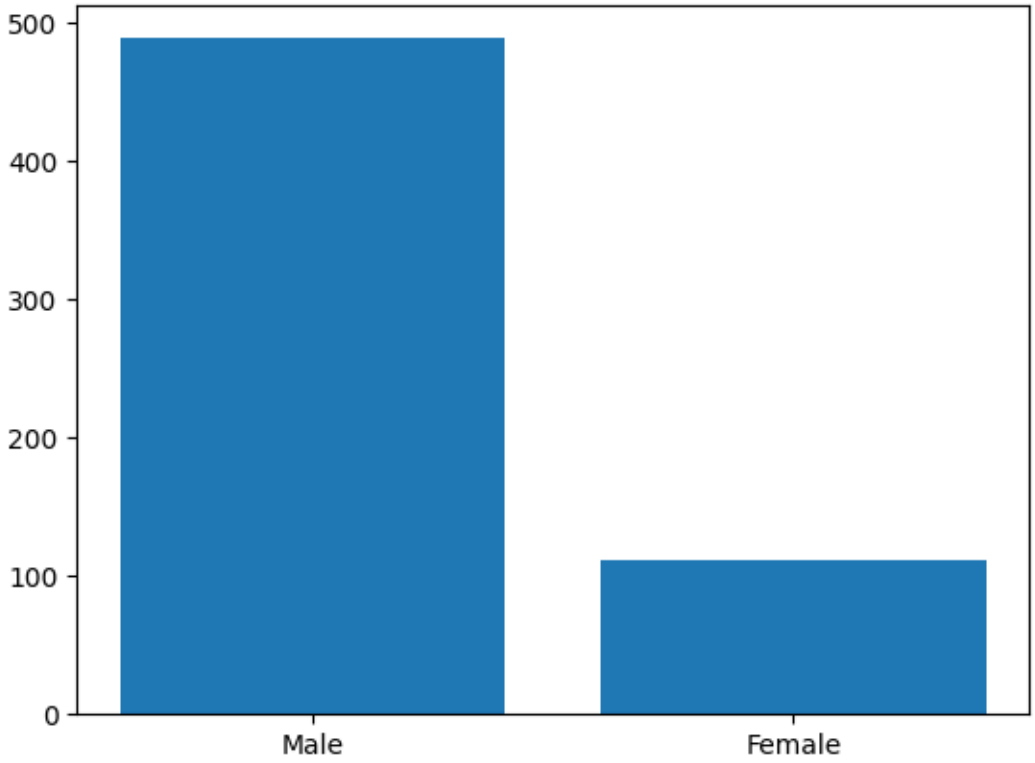



[ ]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

data = pd.read_csv('/content/train_u6lujuX_CVtuZ9i.csv')

#UNIVARIATE PLOTS - 1. Histogram 2. Violin Plot 3. Bar Plot

#histogram
plt.hist(data['Loan_Amount_Term'], bins = 10, edgecolor = 'black', color =␣
 ↪'red')
plt.show()

#violin plot
plt.violinplot(data['CoapplicantIncome'])
plt.show()

#bar plot
df = data['Gender'].value_counts()
plt.bar(df.index, df)
```

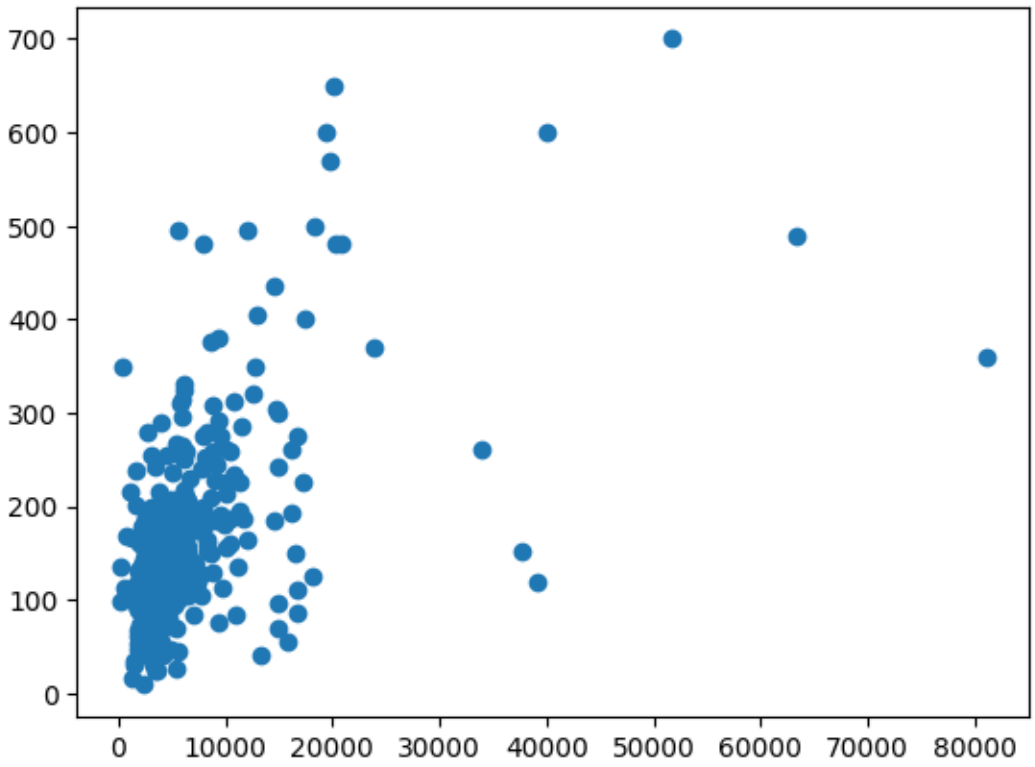
```
plt.show()
```

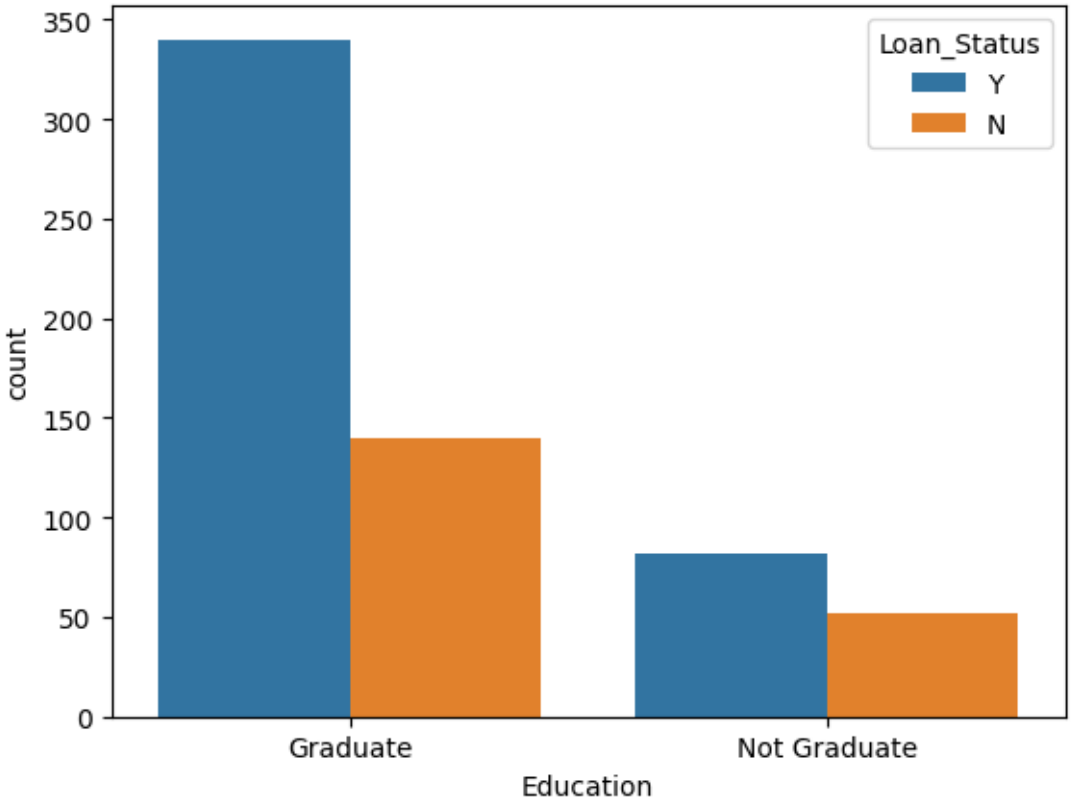


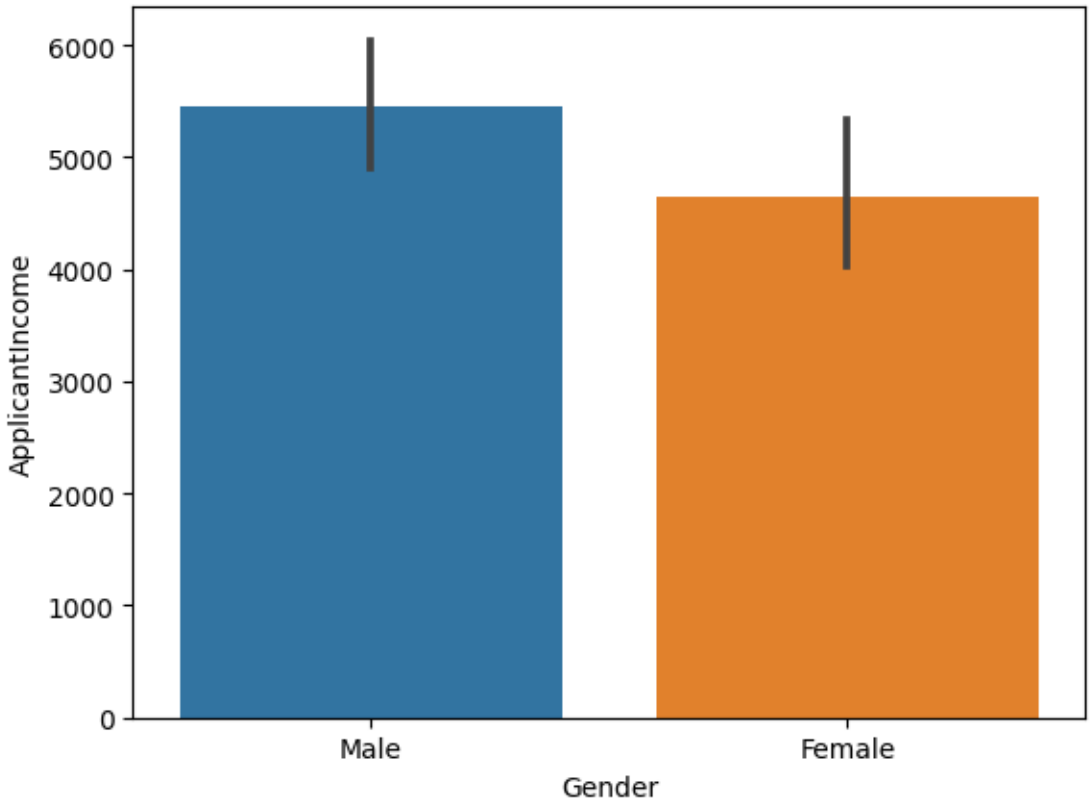

40000
30000
20000
10000
0
0.8
0.9
1.0
1.1
1.2


500
400
300
200
100
0
Male
Female

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

data = pd.read_csv('/content/train_u6lujuX_CVtuZ9i.csv')

#BIVARIATE PLOTS - 1. Scatterplot 2. Count Plot 3. Bar Plot

#scatterplot
plt.scatter(data['ApplicantIncome'], data['LoanAmount'])
plt.show()

#count plot
sns.countplot(x = 'Education', hue = 'Loan_Status', data = data)
plt.show()

#bar plot
sns.barplot(x = 'Gender', y = 'ApplicantIncome', data = data)
plt.show()
```



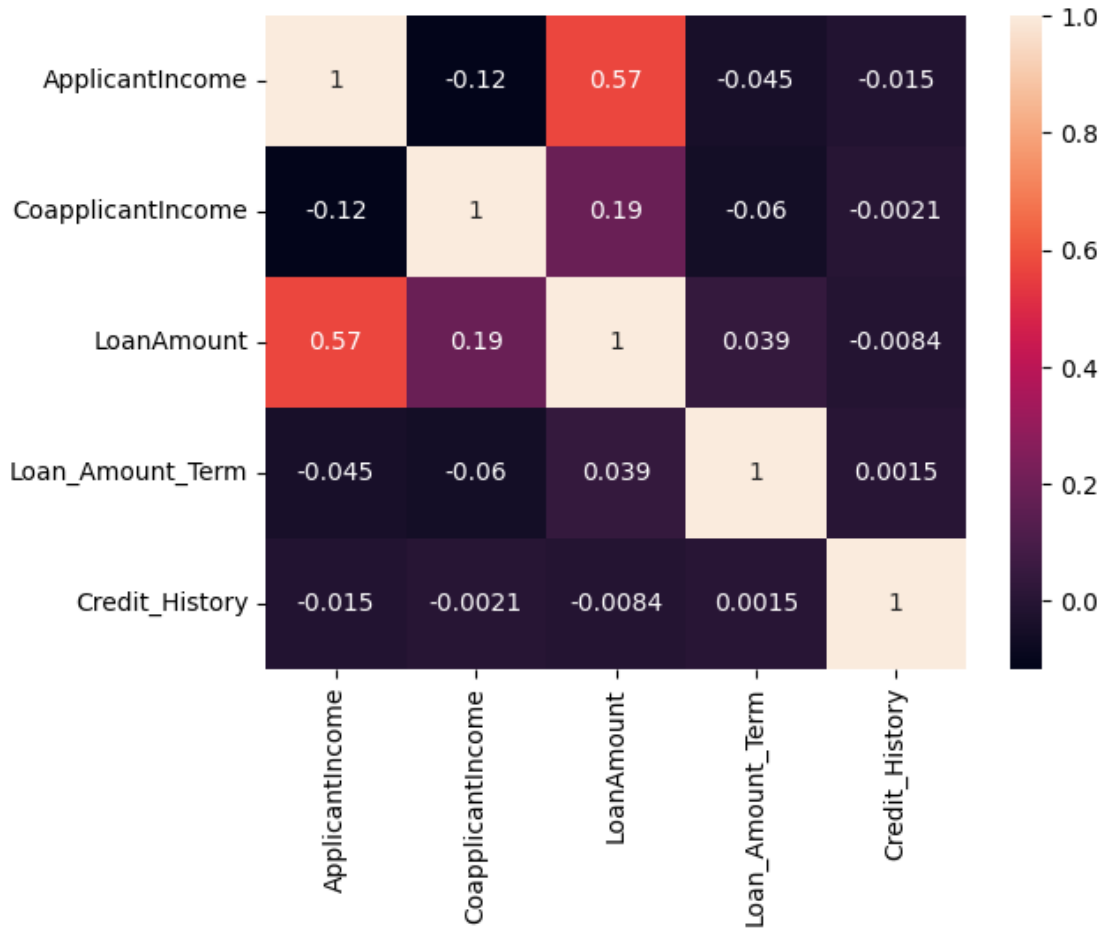

Loan_Status
Y
N
350
300
250
200
150
100
50
0
count
Graduate
Not Graduate
Education

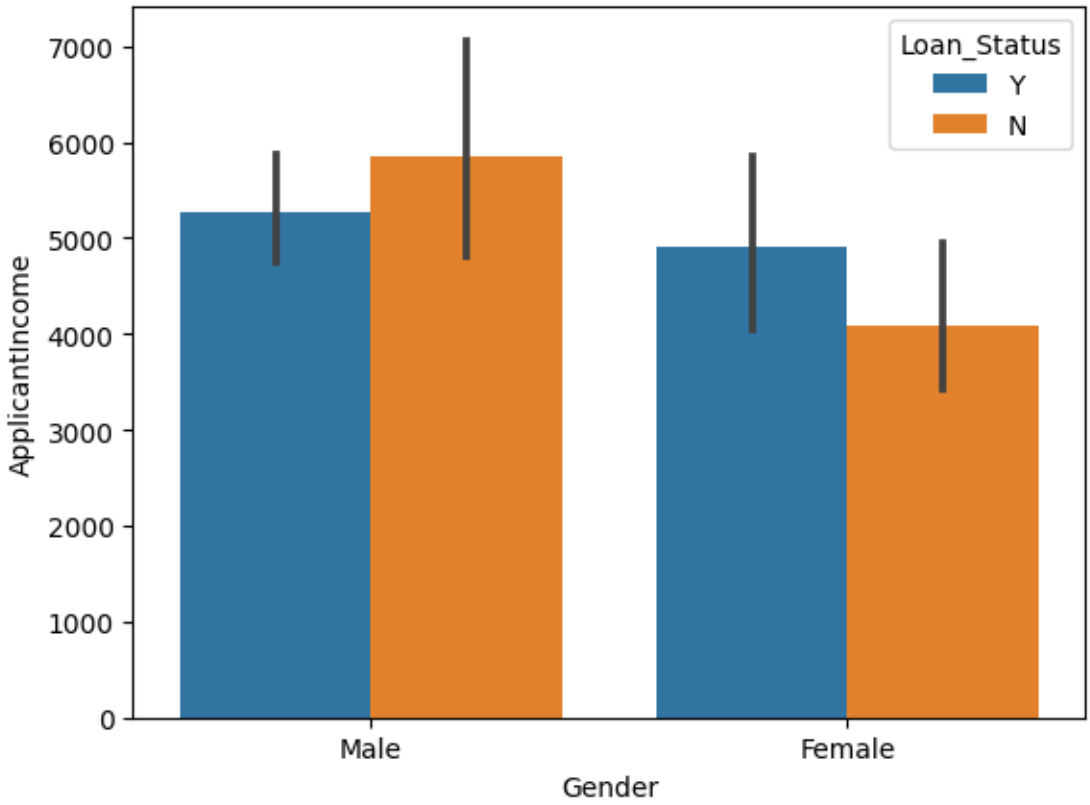



[ ]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns

data = pd.read_csv('/content/train_u6lujuX_CVtuZ9i.csv')

#MULTIVARIATE PLOTS - 1. Heatmap 2. Bar Plot

#heatmap
sns.heatmap(data = data.corr(), annot = True)
plt.show()

#bar plot
sns.barplot(x = 'Gender', y = 'ApplicantIncome', hue = 'Loan_Status', data =␣
 ↪data)
plt.show()
```

```
<ipython-input-67-bf64d3ec8922>:11: FutureWarning: The default value of
numeric_only in DataFrame.corr is deprecated. In a future version, it will
default to False. Select only valid columns or specify the value of numeric_only
```

```
to silence this warning.
  sns.heatmap(data = data.corr(), annot = True)
```




Loan_Status
Y
N
7000
6000
5000
4000
3000
2000
1000
0
ApplicantIncome
Male
Female
Gender