on9vw2bry

April 29, 2024

[ ]:

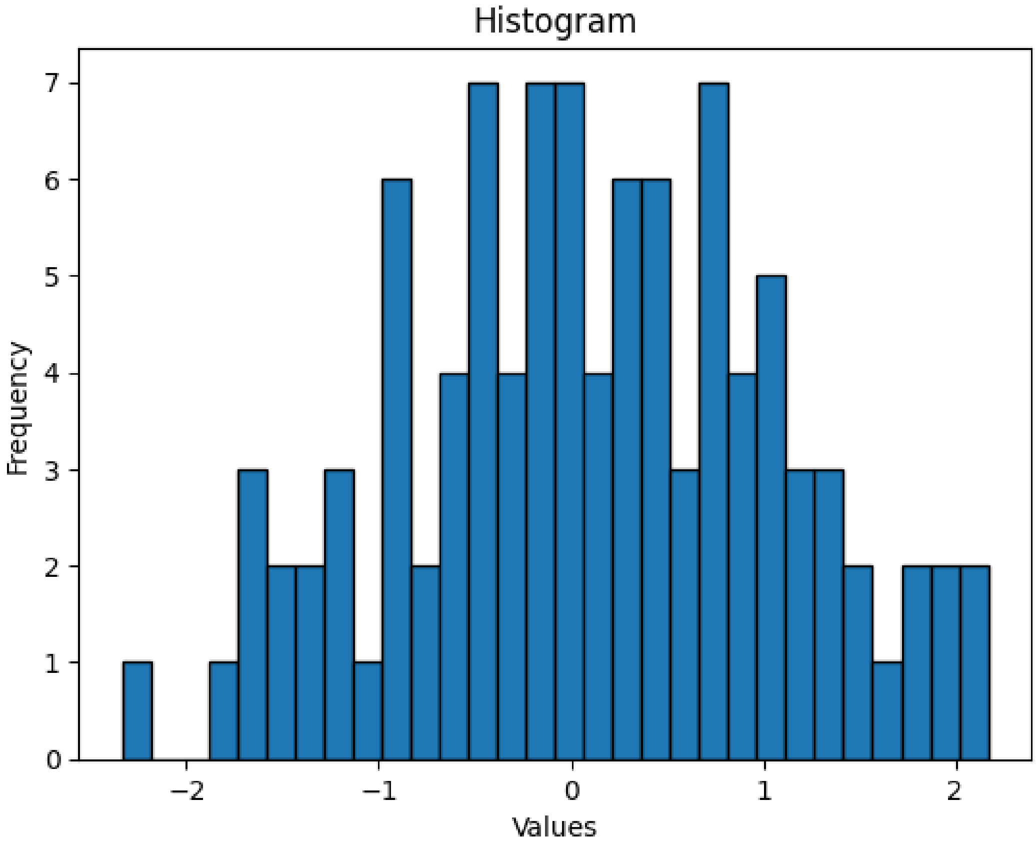
```
# Plotting a basic histogram
import matplotlib.pyplot as plt
import numpy as np

data = np.random.randn(100)


plt.hist(data, bins=30, edgecolor='black')

plt.xlabel('Values')
plt.ylabel('Frequency')
plt.title('Histogram')

plt.show()
```

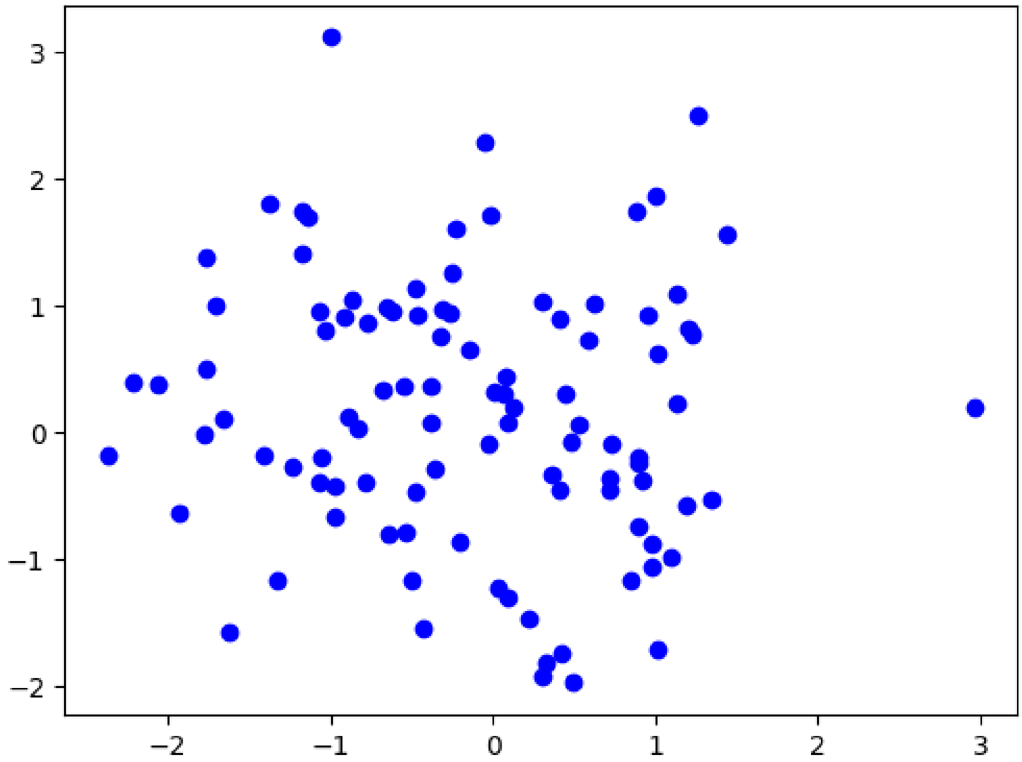


[ ]:

```
#scatterplot
import matplotlib.pyplot as plt
import numpy as np
x =np.random.randn(100)
y =np.random.randn(100)

plt.scatter(x, y, c ="blue")

plt.show()
```



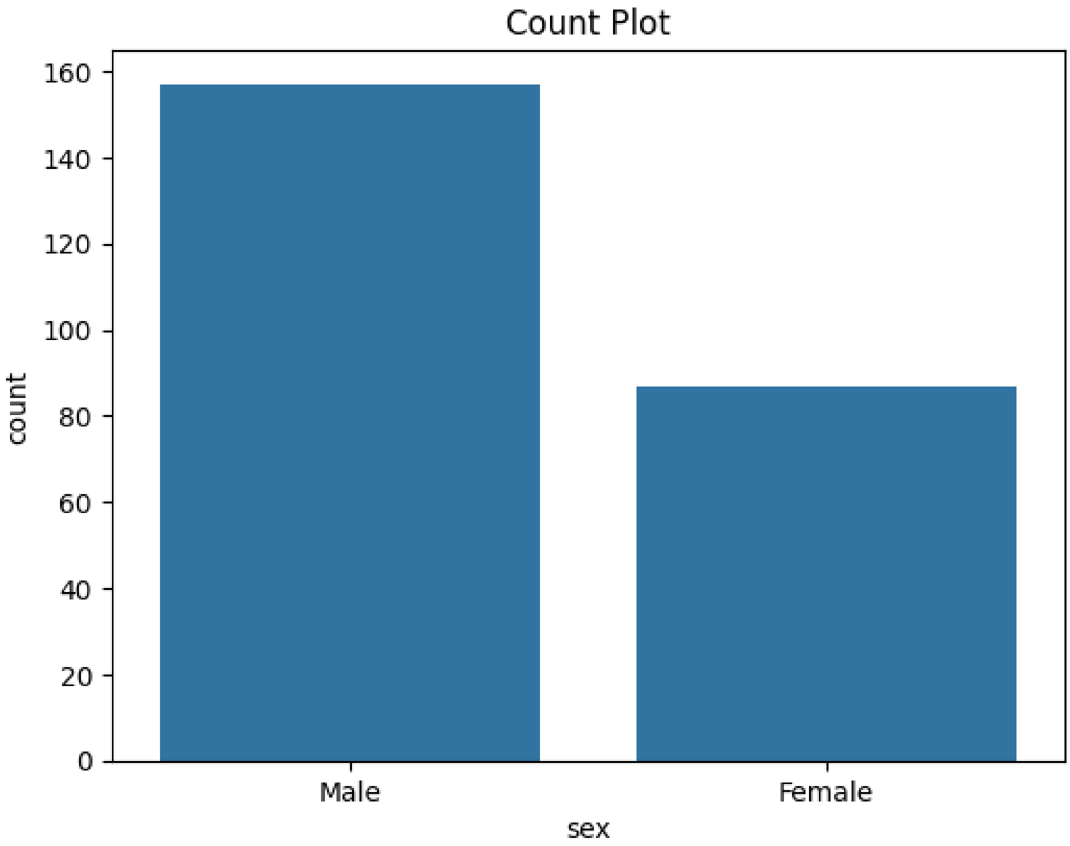
[ ]:

```
#count plot
import seaborn as sns
import matplotlib.pyplot as plt

df = sns.load_dataset('tips')

sns.countplot(x ='sex', data = df)

plt.title('Count Plot')
plt.show()
```



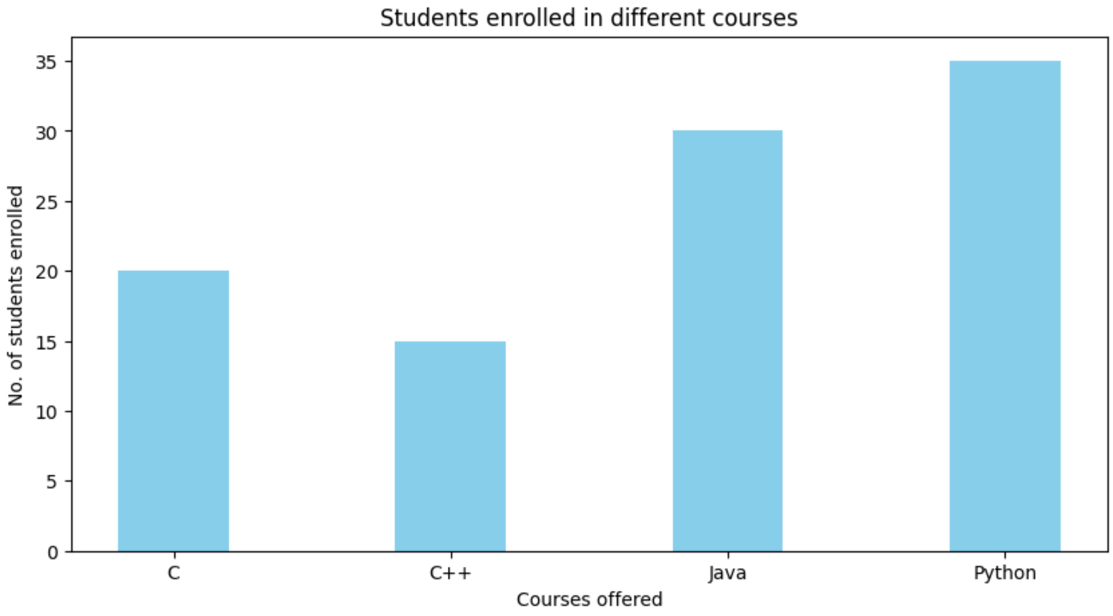
```
[ ]: import numpy as np
     import matplotlib.pyplot as plt


     # creating the dataset
     data = {'C':20, 'C++':15, 'Java':30,
             'Python':35}
     courses = list(data.keys())
     values = list(data.values())

     fig = plt.figure(figsize = (10, 5))

     # creating the bar plot
     plt.bar(courses, values, color ='skyblue',
             width = 0.4)

     plt.xlabel("Courses offered")
     plt.ylabel("No. of students enrolled")
     plt.title("Students enrolled in different courses")
     plt.show()
```



[ ]:
```
import pandas as pd
data=pd.read_csv("tes.csv")
data.describe()
```

[ ]:
```
       ApplicantIncome  CoapplicantIncome  LoanAmount  Loan_Amount_Term  \
count       367.000000         367.000000  362.000000        361.000000
mean       4805.599455        1569.577657  136.132597        342.537396
std        4910.685399        2334.232099   61.366652         65.156643
min           0.000000           0.000000   28.000000          6.000000
25%        2864.000000           0.000000  100.250000        360.000000
50%        3786.000000        1025.000000  125.000000        360.000000
75%        5060.000000        2430.500000  158.000000        360.000000
max       72529.000000       24000.000000  550.000000        480.000000

       Credit_History
count      338.000000
mean         0.825444
std          0.380150
min          0.000000
25%          1.000000
50%          1.000000
75%          1.000000
max          1.000000
```

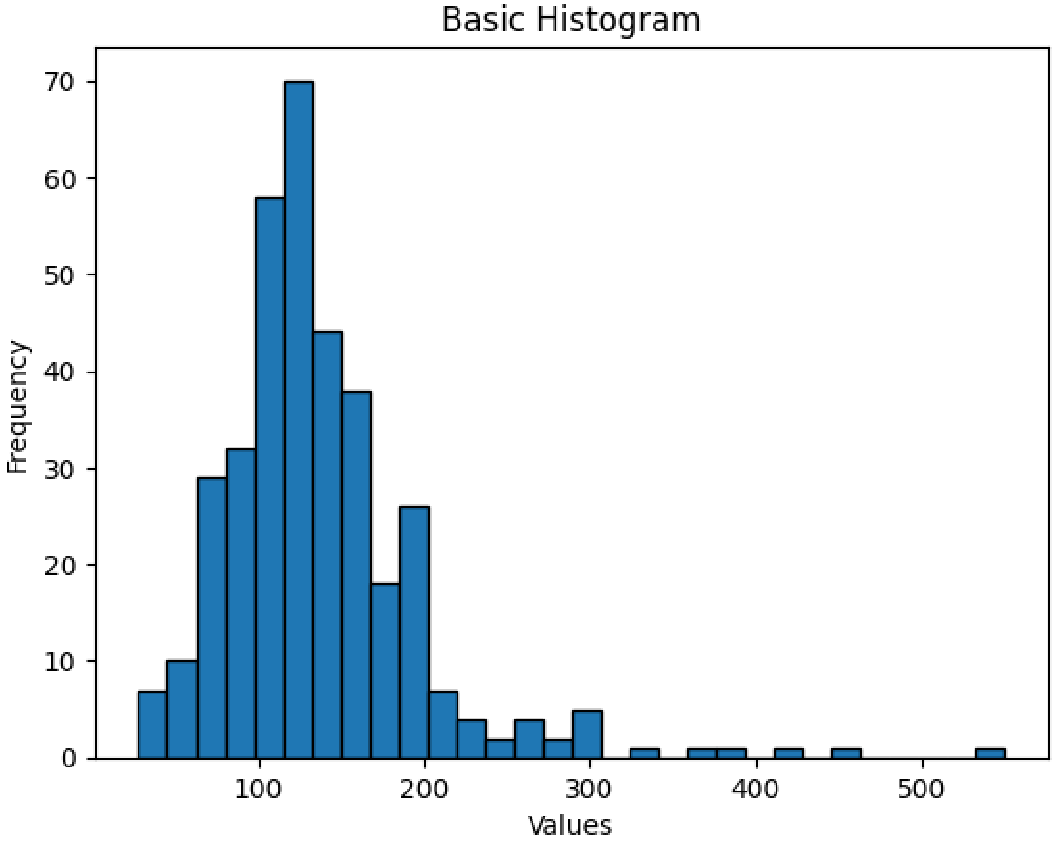
[ ]:
```
import matplotlib.pyplot as plt
import numpy as np
```

[ ]:

```
#Univariate Histogram
plt.hist(data['LoanAmount'], bins=30, edgecolor='black')

plt.xlabel('Values')
plt.ylabel('Frequency')
plt.title('Basic Histogram')

plt.show()
```



[ ]:

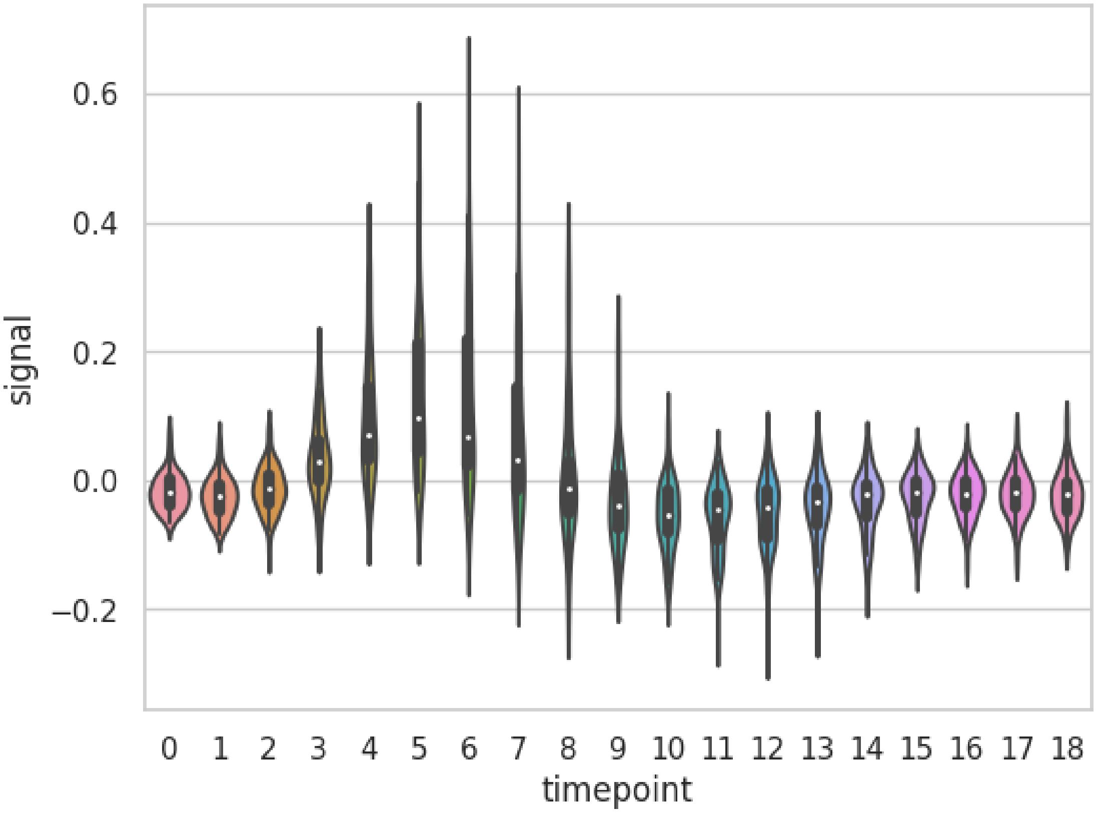
```
import seaborn


seaborn.set(style = 'whitegrid')
fmri = seaborn.load_dataset("fmri")

seaborn.violinplot(x ="timepoint",
                   y ="signal",
                   data = fmri)
```

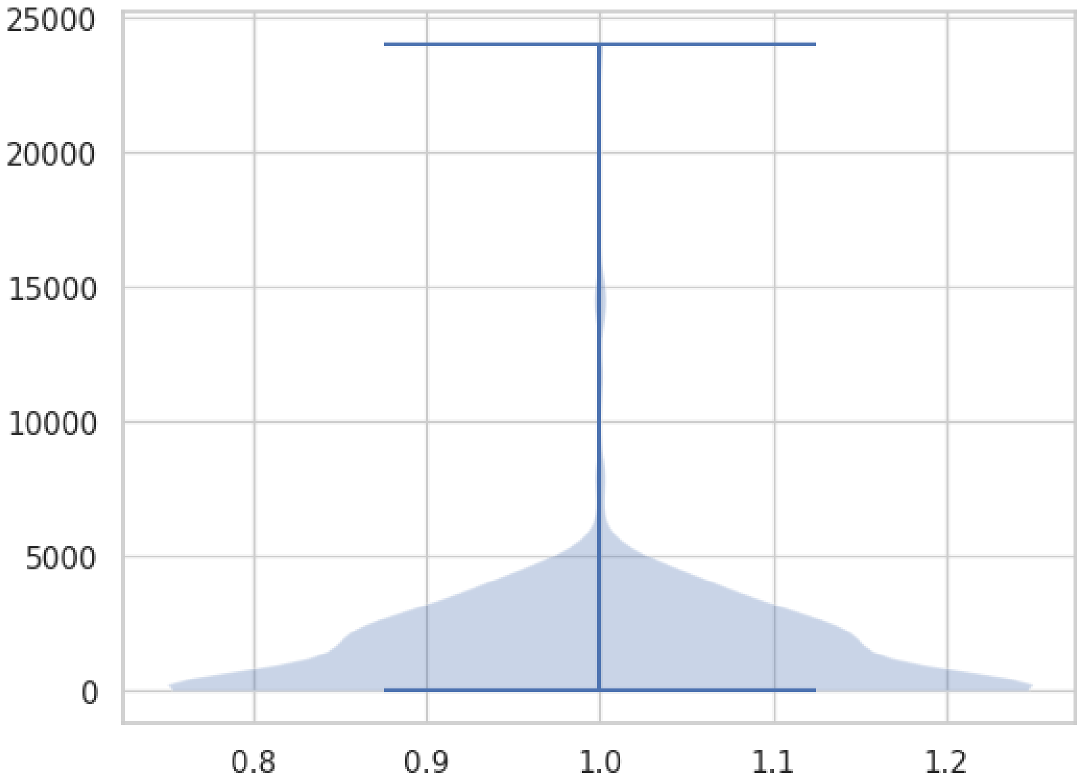
[ ]:

```
<Axes: xlabel='timepoint', ylabel='signal'>
```



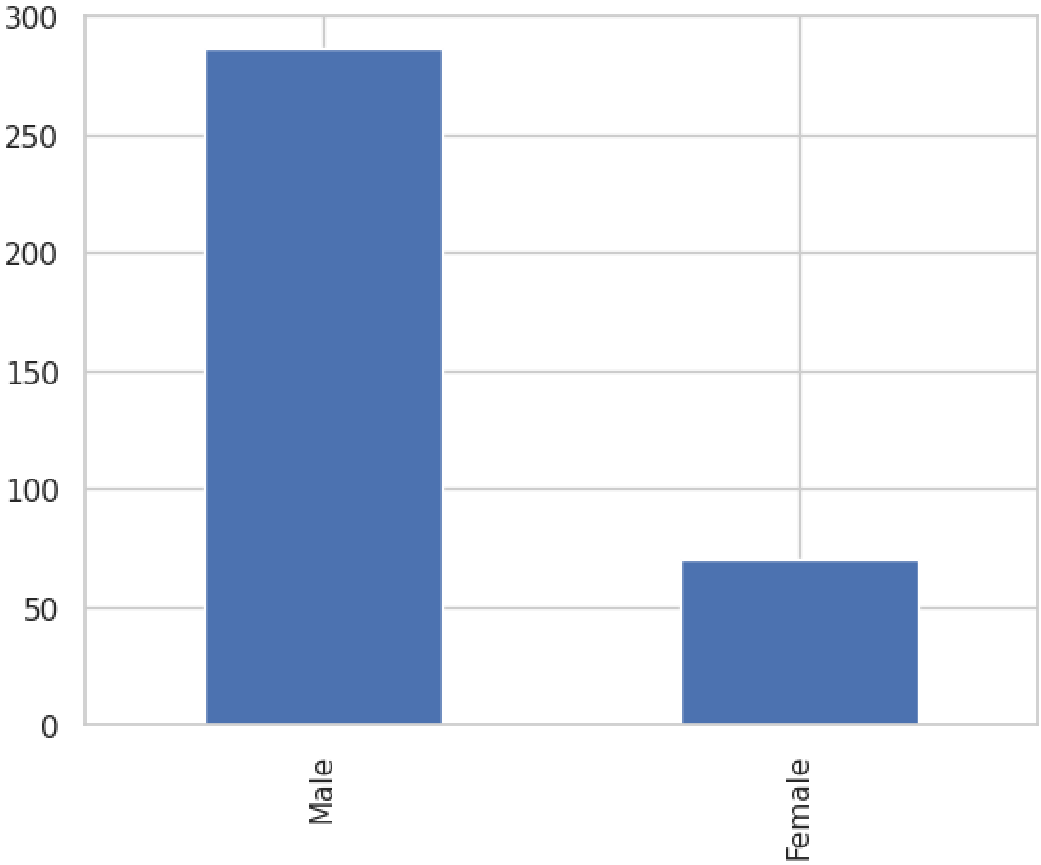
```
[ ]: #Violin plot
     plt.violinplot(data["CoapplicantIncome"] )
```

```
[ ]: {'bodies': [<matplotlib.collections.PolyCollection at 0x786ee9f7d5a0>],
      'cmaxes': <matplotlib.collections.LineCollection at 0x786ee9f7d540>,
      'cmins': <matplotlib.collections.LineCollection at 0x786ee9f7da50>,
      'cbars': <matplotlib.collections.LineCollection at 0x786ee9f7ddb0>}
```



```
[ ]: #bar graph
     gen=data["Gender"].value_counts()
     gen.plot(kind='bar')
```

```
[ ]: <Axes: >
```

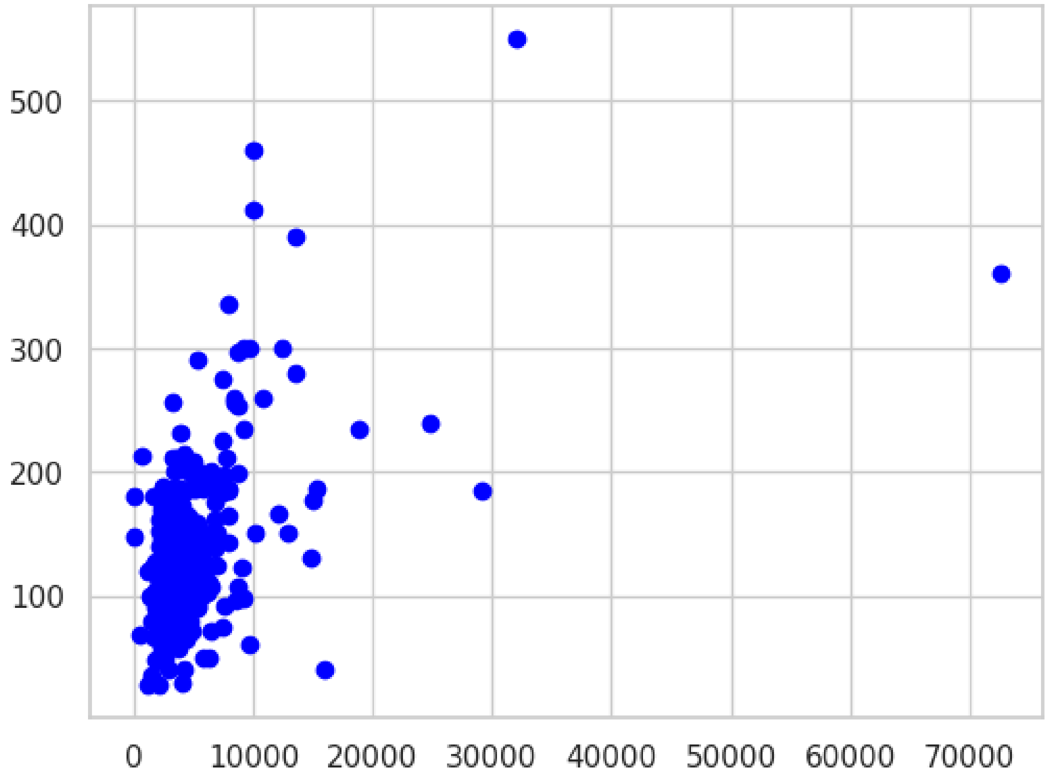


Bivariate

```
[ ]: #scatter plot
     x =data["ApplicantIncome"]
     y =data["LoanAmount"]

     plt.scatter(x, y, c ="blue")

     plt.show()
```

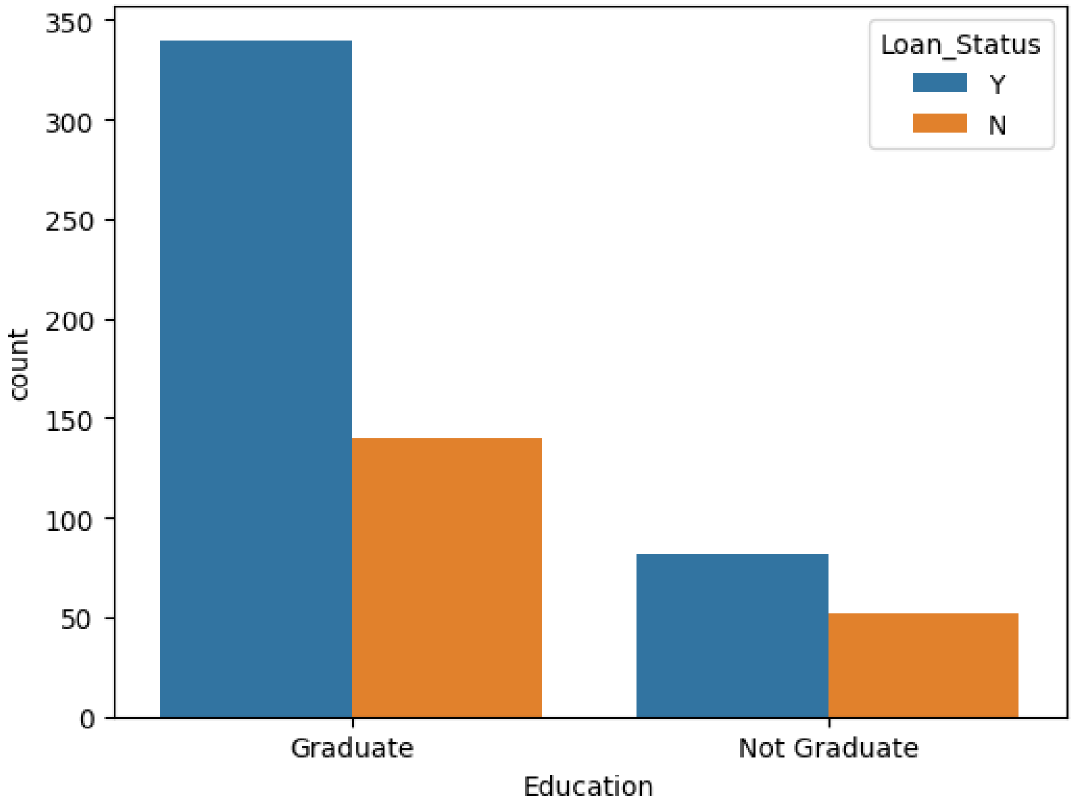


[ ]:
```
cou=data["Education"].value_counts()
cou.countplot(hue=data["LoanStatus"])

plt.show()
```

[ ]:
```
data2=pd.read_csv("train.csv")
```

[ ]:
```
import seaborn as sns
sns.countplot(x='Education',hue='Loan_Status',data=data2)
plt.show()
```



```
[ ]: fig = plt.figure(figsize = (10, 5))
     gen=data["Gender"].value_counts()
     # creating the bar plot
     plt.bar(data["Gender"],data["ApplicantIncome"], values, color ='maroon',
             width = 0.4)
```

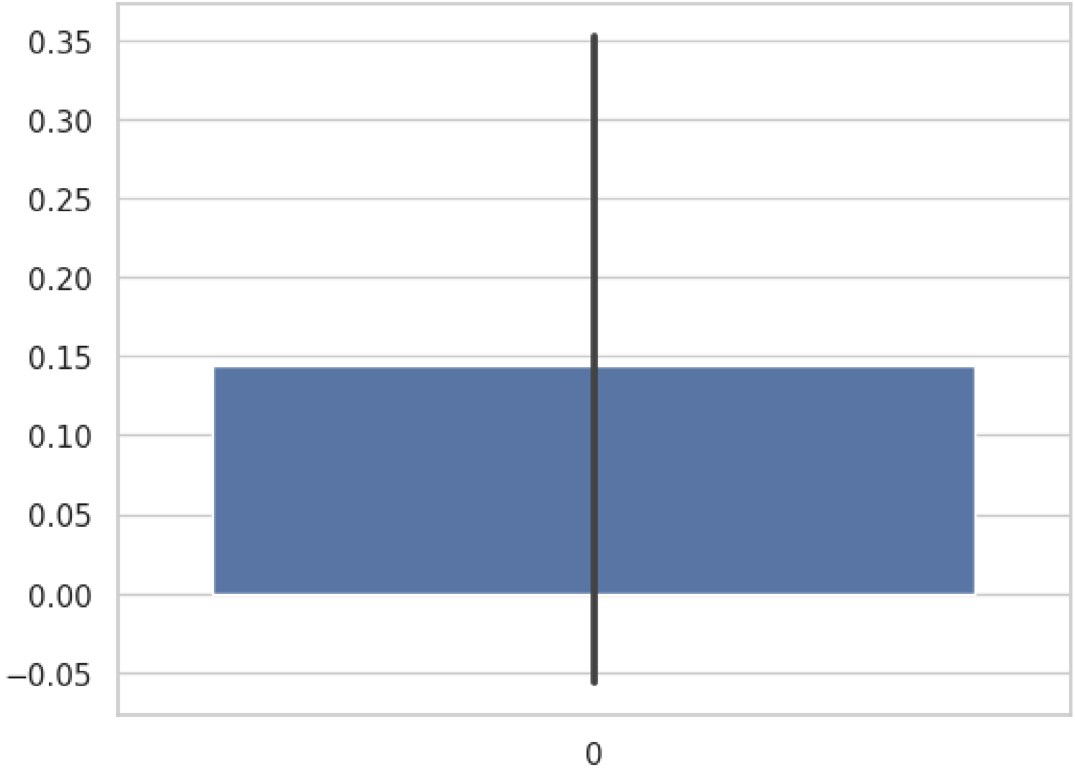
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-7-0b4162a5bd63> in <cell line: 4>()
      2 gen=data["Gender"].value_counts()
      3 # creating the bar plot
----> 4 plt.bar(data["Gender"],data["ApplicantIncome"], values, color ='maroon'

      5         width = 0.4)

NameError: name 'values' is not defined
```

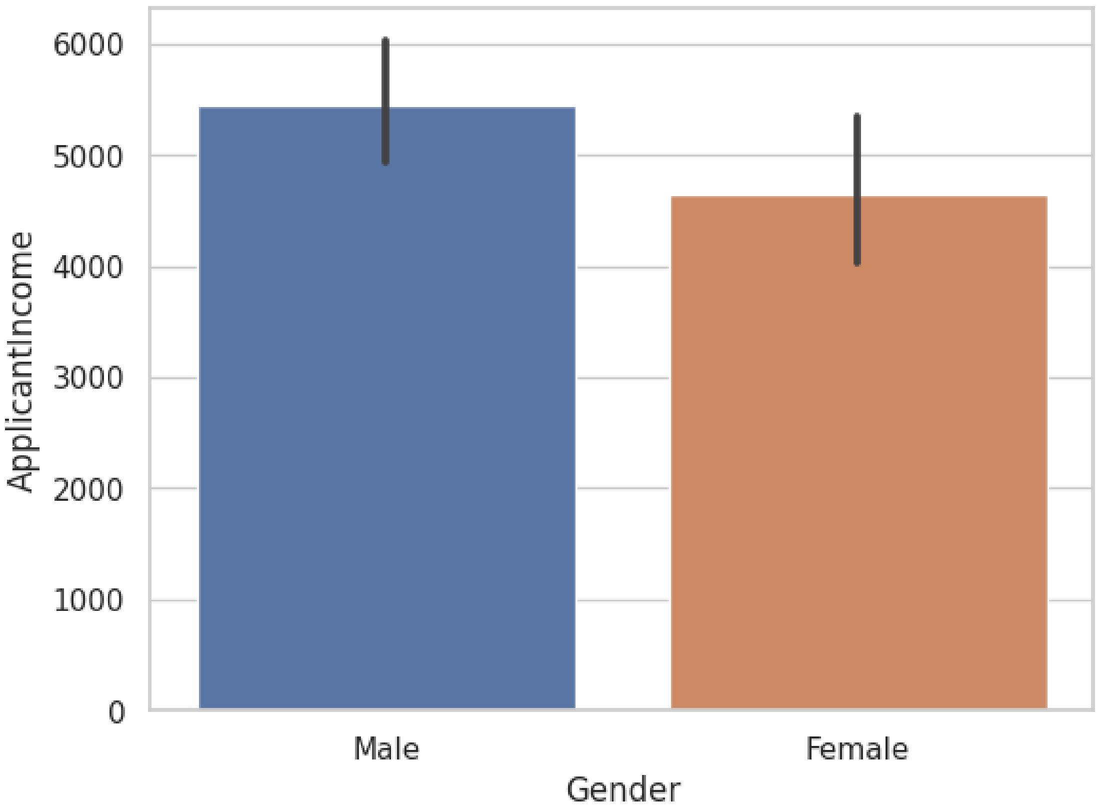
```
<Figure size 1000x500 with 0 Axes>
```

```
[ ]: sns.barplot(data)
[ ]: <Axes: >
```



```
[ ]: sns.barplot(data2,x='Gender',y='ApplicantIncome')
```

```
[ ]: <Axes: xlabel='Gender', ylabel='ApplicantIncome'>
```

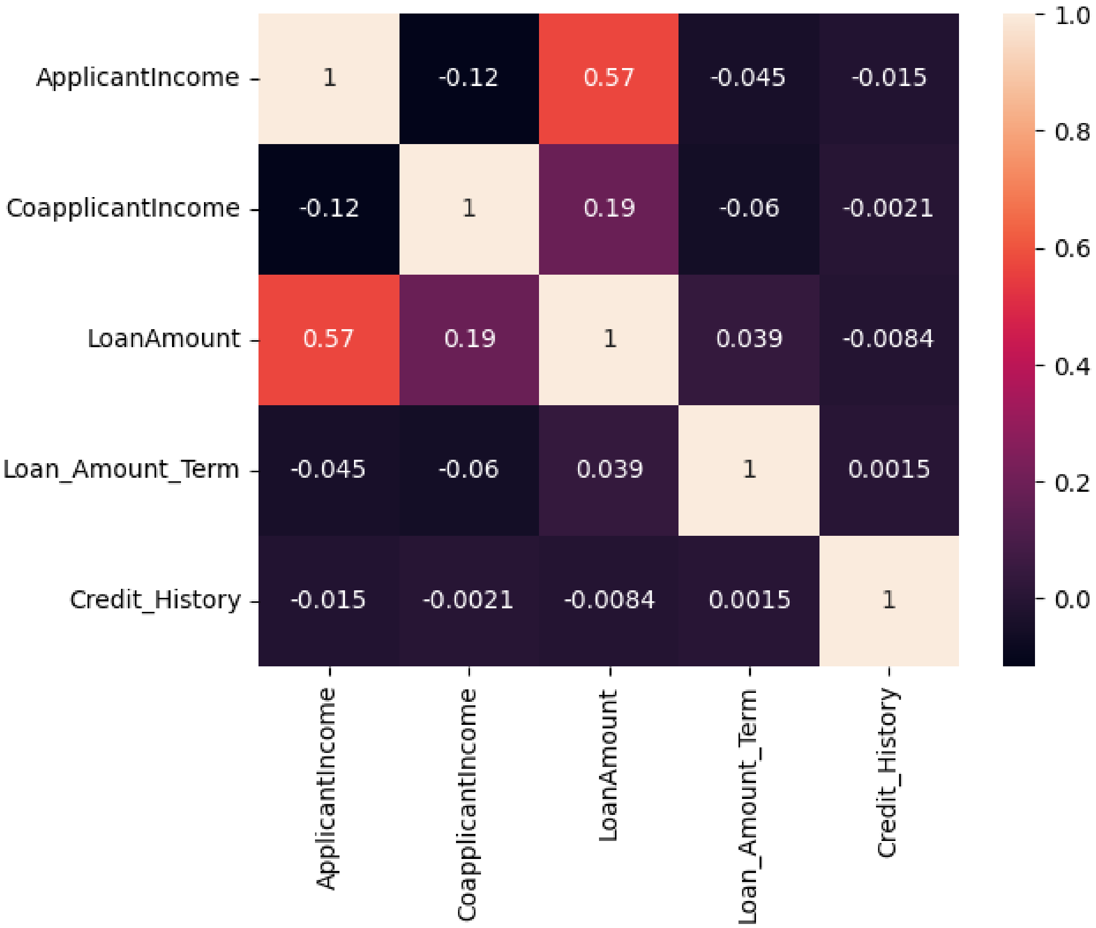


Multivariate

```
[ ]: #Heatmap
     hm = sns.heatmap(data2.corr(),annot=True)

     # displaying the plotted heatmap
     plt.show()
```

```
<ipython-input-11-ca9308b22801>:2: 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.
  hm = sns.heatmap(data2.corr(),annot=True)
```



[ ]:

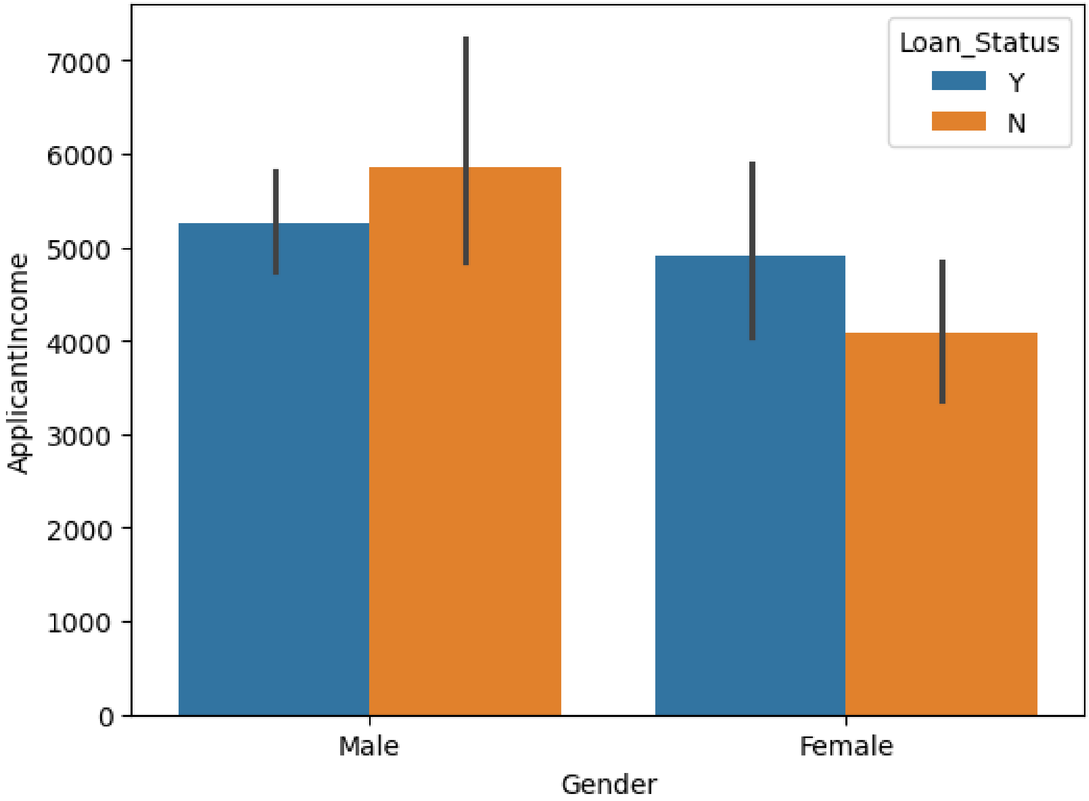
```
```

[ ]:

```
#barplot
sns.barplot(data2,x='Gender',y='ApplicantIncome',hue='Loan_Status')
```

[ ]:

```
<Axes: xlabel='Gender', ylabel='ApplicantIncome'>
```


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