Linear_Reg

August 4, 2023

1 LoanTap

- Problem Statement:

  LoanTap aims to leverage a dataset comprising diverse customer attributes to develop a robust predictive model capable of accurately determining the credit rating of customers.This predictive model holds immense value for LoanTap and other financial institutions as it allows them to mitigate the risk of defaults and make precise evaluations of customer creditworthiness.By utilizing this model, LoanTap can make informed decisions concerning credit approvals, loan terms, and interest rates, thus enhancing their ability to offer appropriate financial solutions to customers while maintaining a secure lending portfolio.

[200]:
```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="darkgrid")
import scipy.stats as stats

from statsmodels.graphics.gofplots import qqplot
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

[201]:
```
df = pd.read_csv("Credit.csv")
df.head()
```

[201]:
```
   Unnamed: 0   Income  Limit  Rating  Cards  Age  Education  Gender Student  \
0           1   14.891   3606     283      2   34         11    Male      No
1           2  106.025   6645     483      3   82         15  Female     Yes
2           3  104.593   7075     514      4   71         11    Male      No
3           4  148.924   9504     681      3   36         11  Female      No
4           5   55.882   4897     357      2   68         16    Male      No
```

```
  Married  Ethnicity  Balance
0     Yes  Caucasian      333
1     Yes      Asian      903
2      No      Asian      580
3      No      Asian      964
4     Yes  Caucasian      331
```

[202]:
```
df.drop("Unnamed: 0",inplace=True,axis=1)
```

[203]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 11 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Income     400 non-null    float64
 1   Limit      400 non-null    int64
 2   Rating     400 non-null    int64
 3   Cards      400 non-null    int64
 4   Age        400 non-null    int64
 5   Education  400 non-null    int64
 6   Gender     400 non-null    object
 7   Student    400 non-null    object
 8   Married    400 non-null    object
 9   Ethnicity  400 non-null    object
 10  Balance    400 non-null    int64
dtypes: float64(1), int64(6), object(4)
memory usage: 34.5+ KB
```

[204]:
```
df.describe()
```

[204]:
```
           Income         Limit      Rating       Cards         Age  \
count  400.000000    400.000000  400.000000  400.000000  400.000000
mean    45.218885   4735.600000  354.940000    2.957500   55.667500
std     35.244273   2308.198848  154.724143    1.371275   17.249807
min     10.354000    855.000000   93.000000    1.000000   23.000000
25%     21.007250   3088.000000  247.250000    2.000000   41.750000
50%     33.115500   4622.500000  344.000000    3.000000   56.000000
75%     57.470750   5872.750000  437.250000    4.000000   70.000000
max    186.634000  13913.000000  982.000000    9.000000   98.000000

        Education     Balance
count  400.000000  400.000000
mean    13.450000  520.015000
std      3.125207  459.758877
min      5.000000    0.000000
25%     11.000000   68.750000
```

```
50%      14.000000   459.500000
75%      16.000000   863.000000
max      20.000000  1999.000000
```

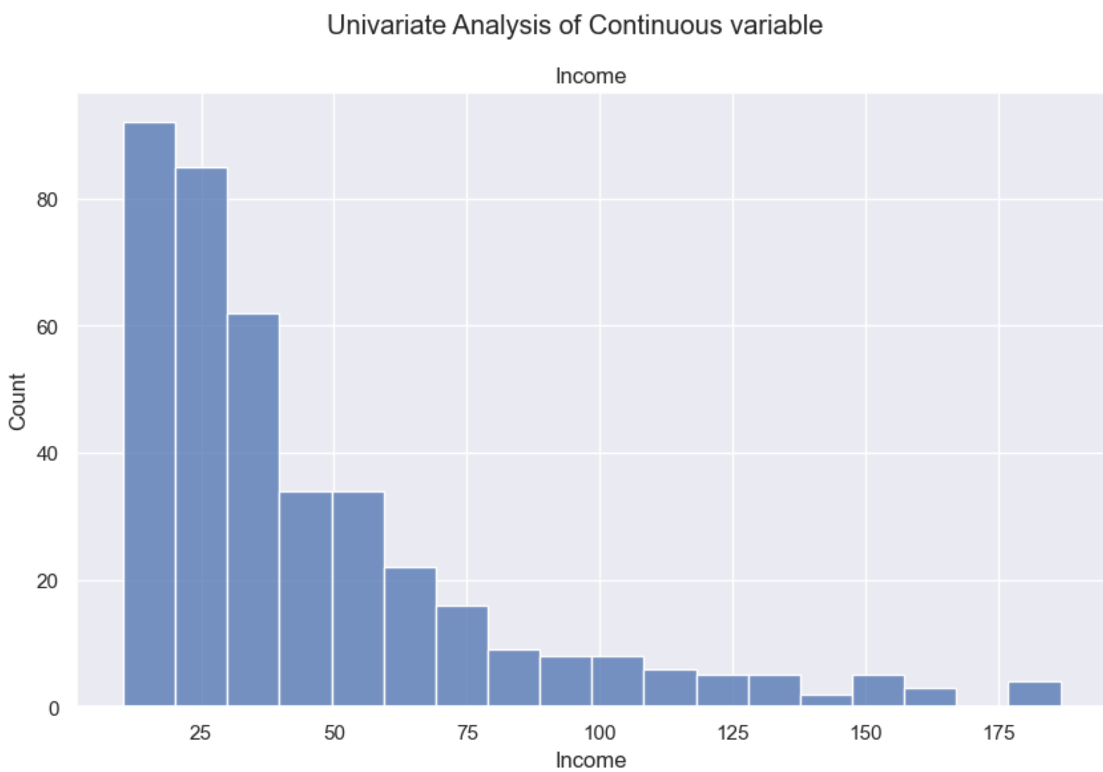
[205]:
```
df.describe(include=object)
```

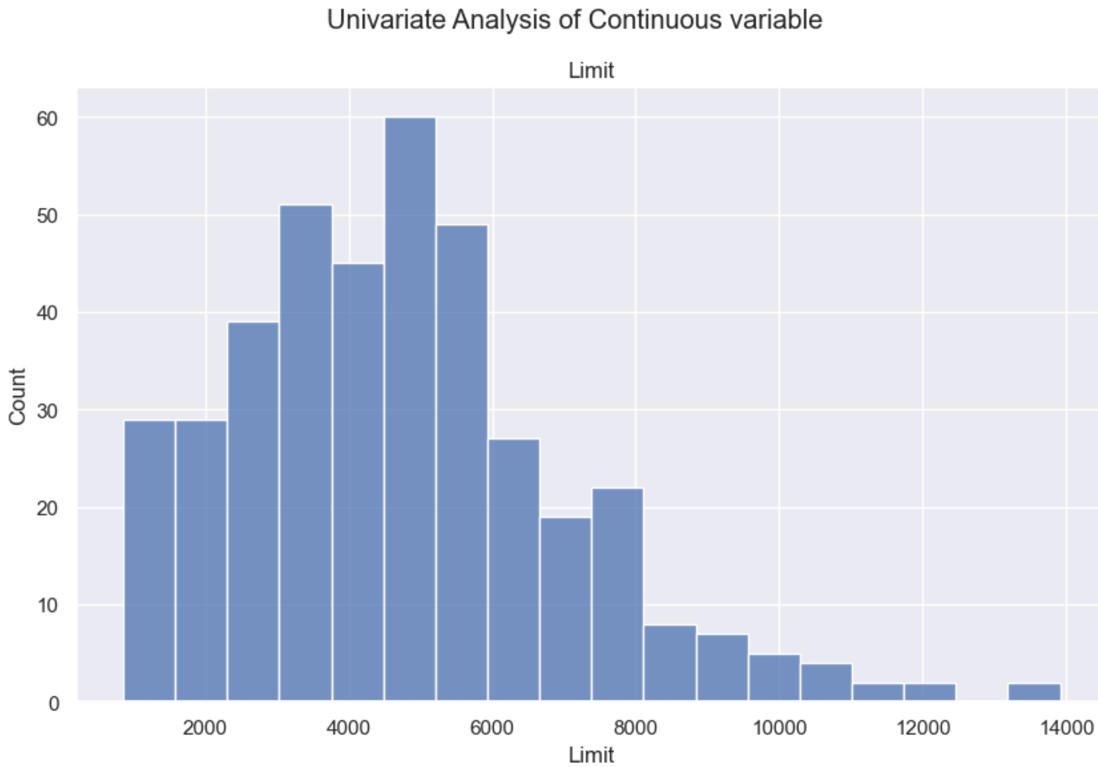
[205]:

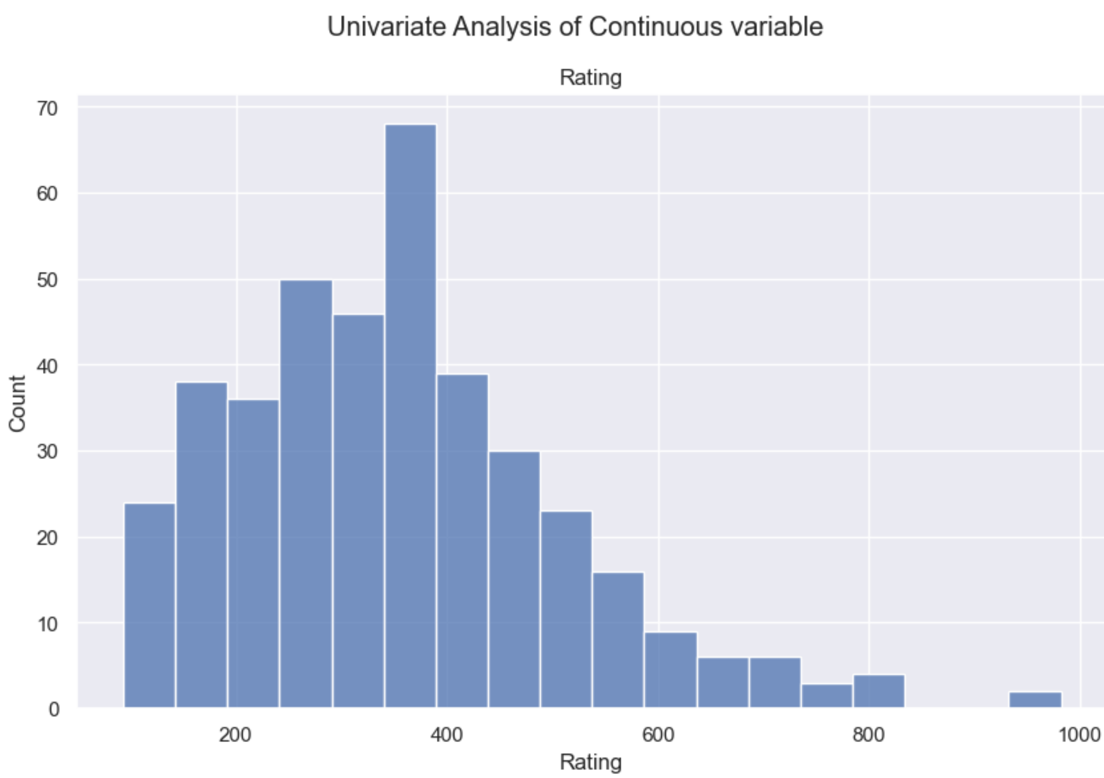
	Gender	Student	Married	Ethnicity
count	400	400	400	400
unique	2	2	2	3
top	Female	No	Yes	Caucasian
freq	207	360	245	199

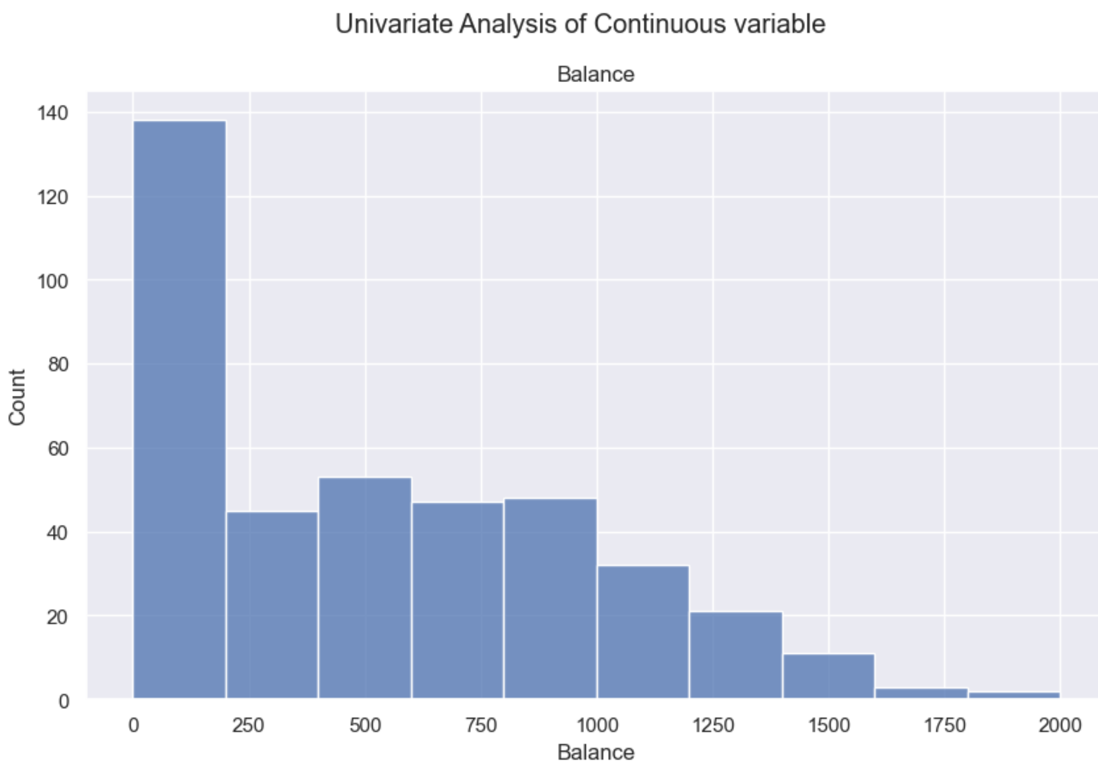
Univariate Analysis

[206]:
```
continuos_variable = ["Income","Limit","Rating","Balance","Age"]

for i in continuos_variable:
    fig, axes = plt.subplots(1,1, figsize=(10,6))
    fig.suptitle("Univariate Analysis of Continuous variable")
    sns.histplot(ax = axes,data=df,x=i)
    axes.set_title(f"{i}")
```

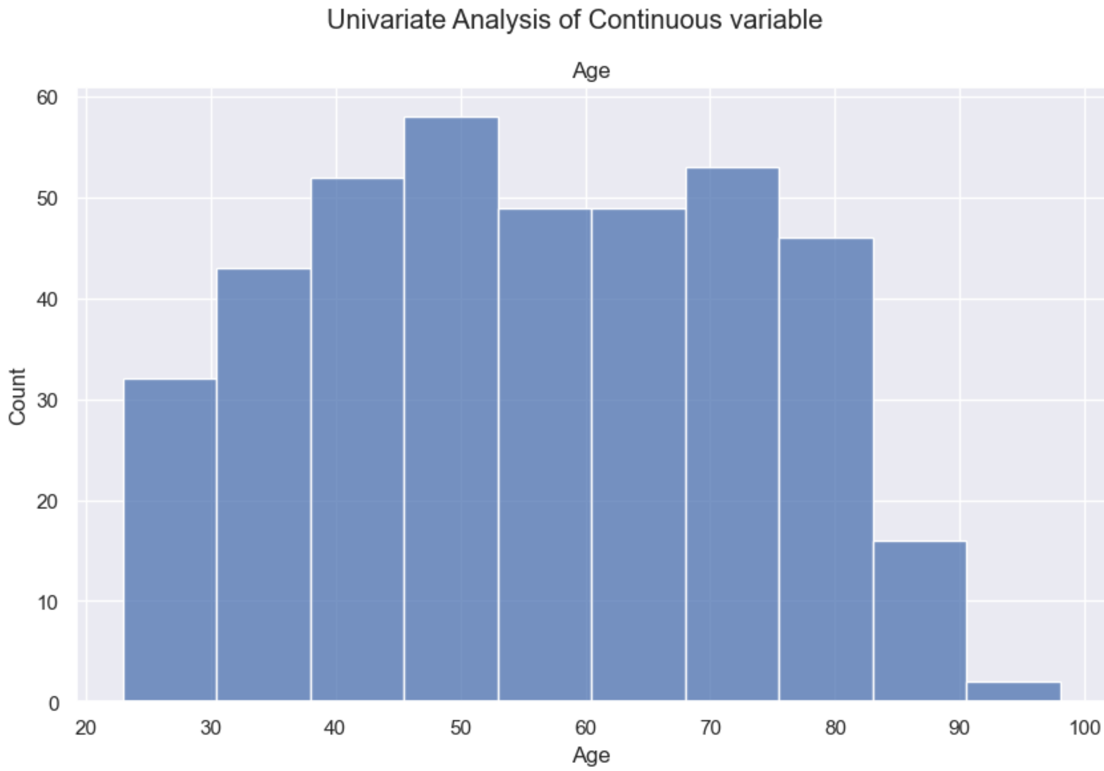



Univariate Analysis of Continuous variable
Limit
Count
0
10
20
30
40
50
60
2000
4000
6000
8000
10000
12000
14000
Limit


Univariate Analysis of Continuous variable
Rating
Count
0
10
20
30
40
50
60
70
200
400
600
800
1000
Rating



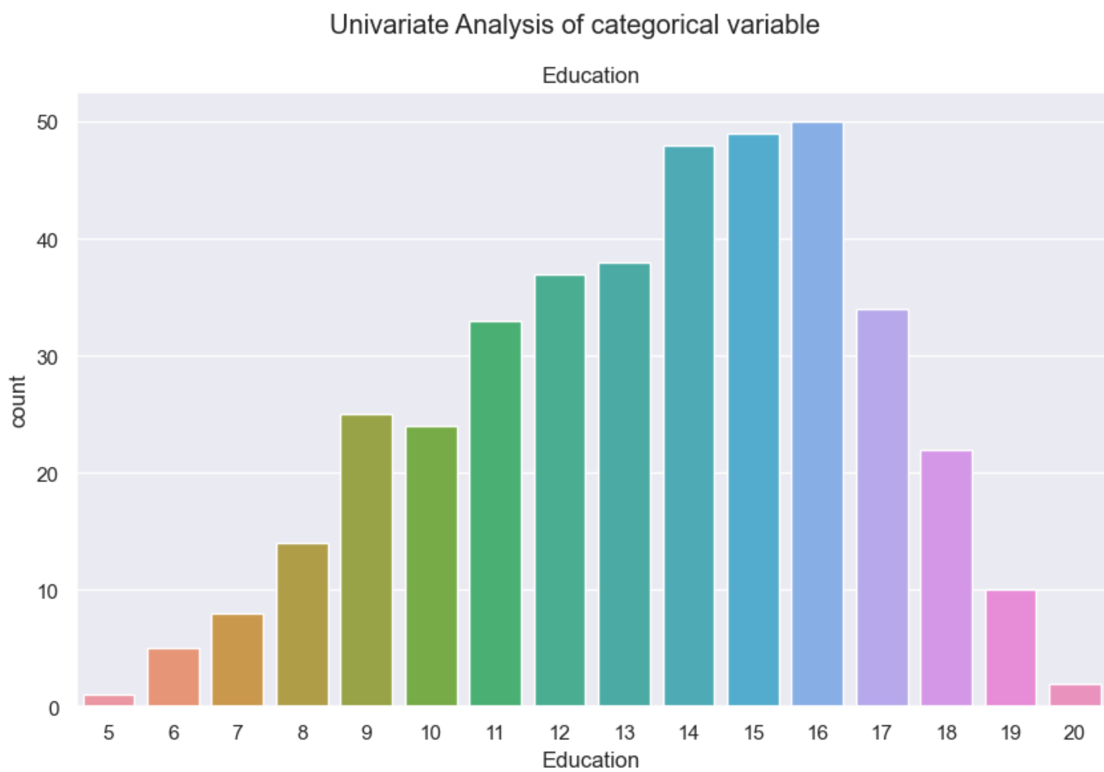
Univariate Analysis of Continuous variable
Balance
Count
140
120
100
80
60
40
20
0
0
250
500
750
1000
1250
1500
1750
2000
Balance

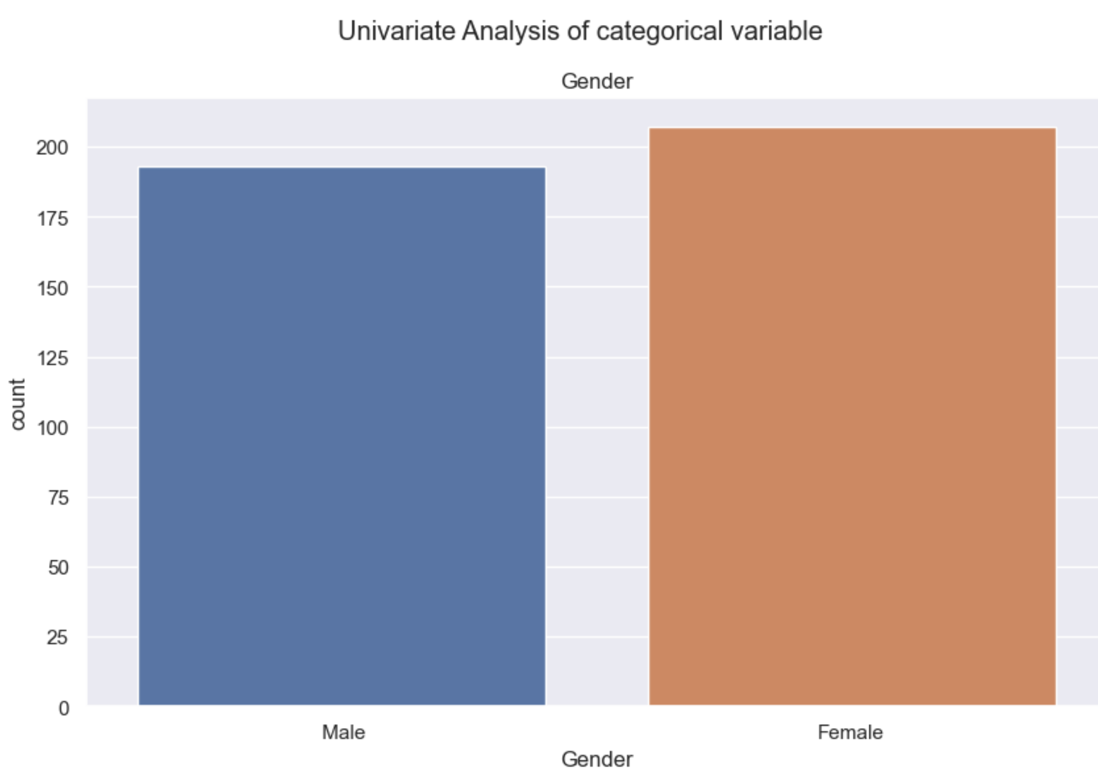


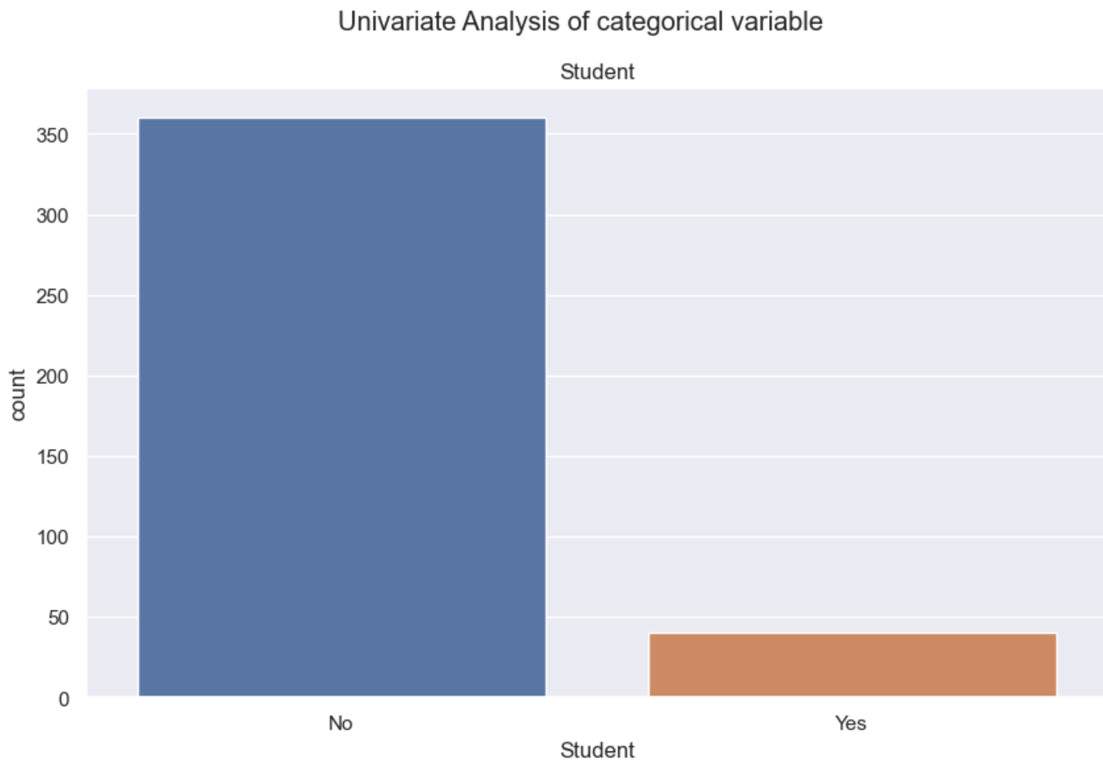
[207]:
```
df.columns
```

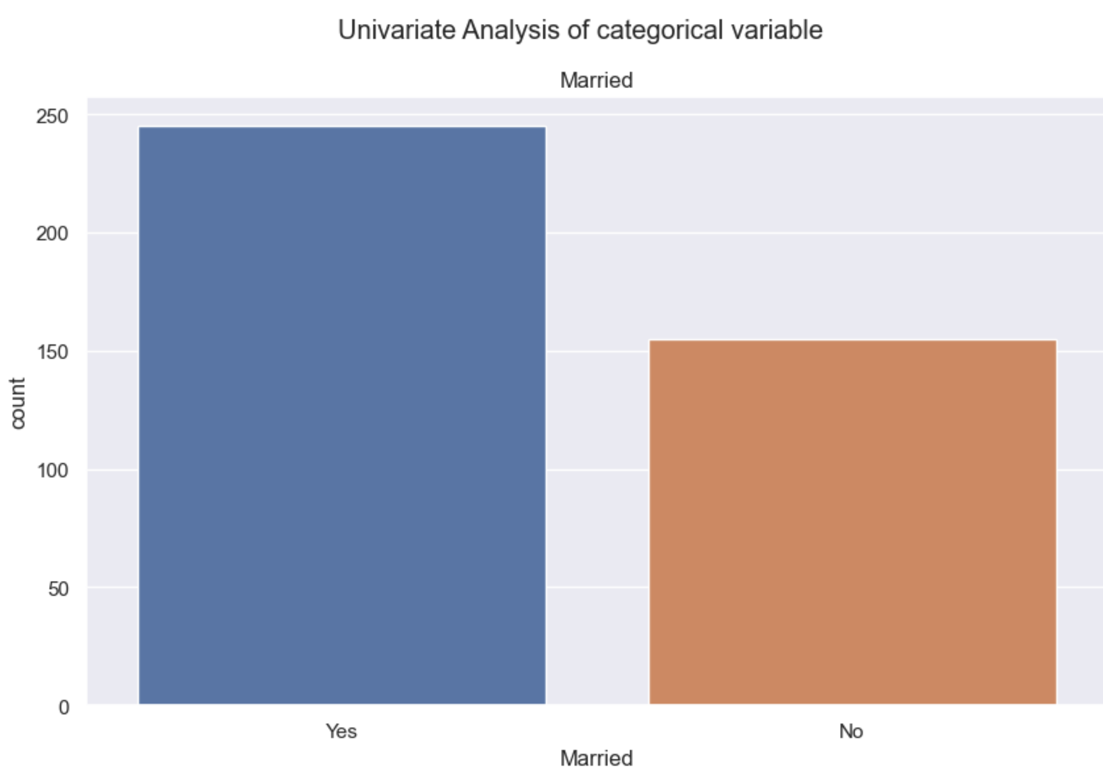
[207]:
```
Index(['Income', 'Limit', 'Rating', 'Cards', 'Age', 'Education', 'Gender',
       'Student', 'Married', 'Ethnicity', 'Balance'],
      dtype='object')
```

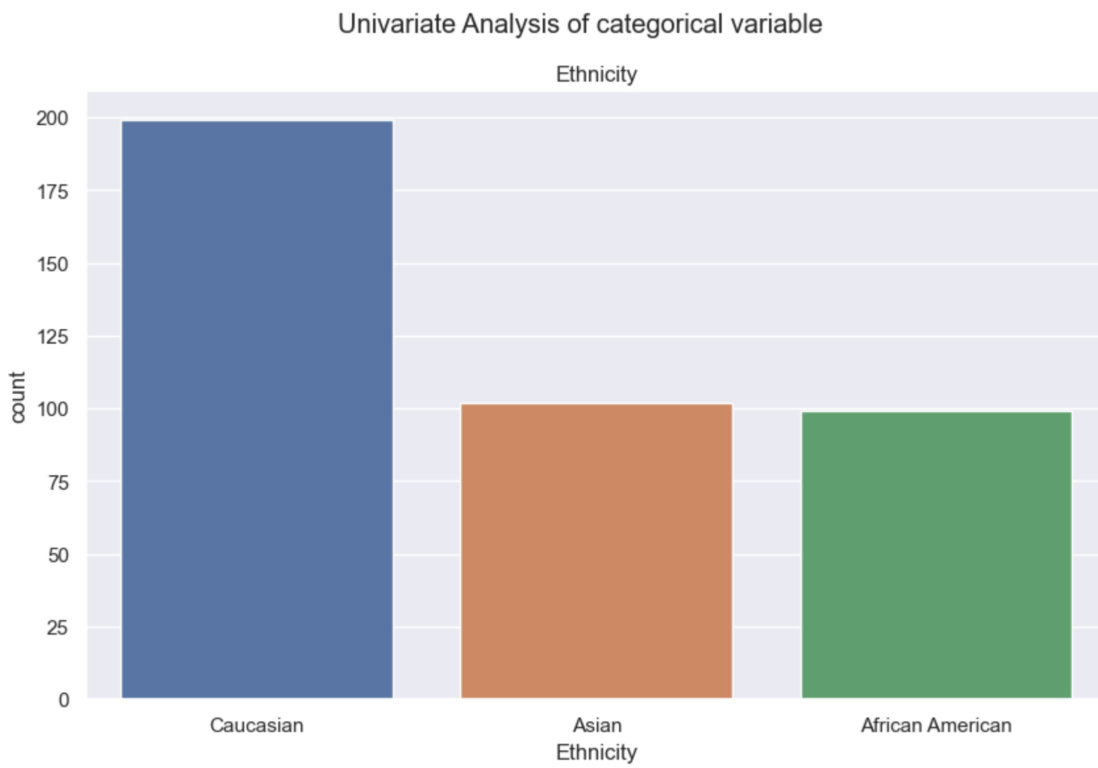
[208]:
```
categorical_variable = ['Education', 'Gender',
       'Student', 'Married', 'Ethnicity', 'Cards']

for i in categorical_variable:
    fig, axes = plt.subplots(1,1, figsize=(10,6))
    fig.suptitle("Univariate Analysis of categorical variable")
    sns.countplot(ax = axes,data=df,x=i)
    axes.set_title(f"{i}")
```

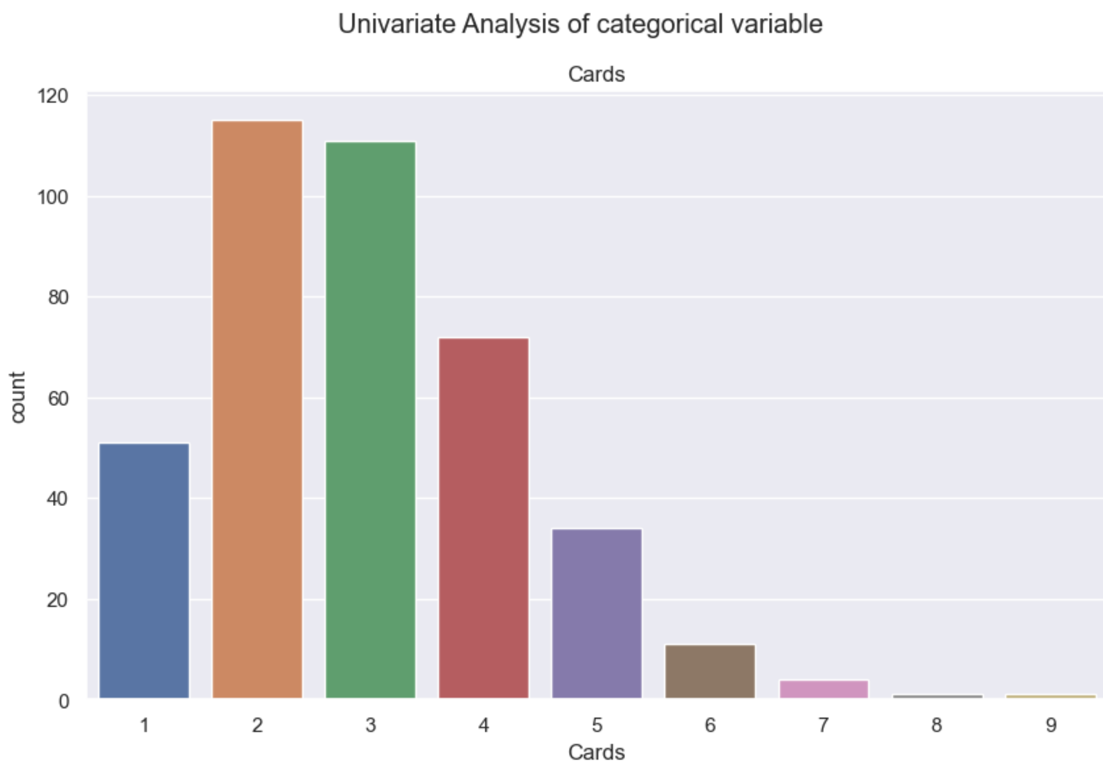

Univariate Analysis of categorical variable
Education
count
0
10
20
30
40
50
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
Education


Univariate Analysis of categorical variable
Gender
count
0
25
50
75
100
125
150
175
200
Male
Female
Gender


Univariate Analysis of categorical variable
Student
count
350
300
250
200
150
100
50
0
No
Yes
Student


Univariate Analysis of categorical variable
Married
count
250
200
150
100
50
0
Yes
No
Married


Univariate Analysis of categorical variable
Ethnicity
count
200
175
150
125
100
75
50
25
0
Caucasian
Asian
African American
Ethnicity



[209]:
```
df.columns
```

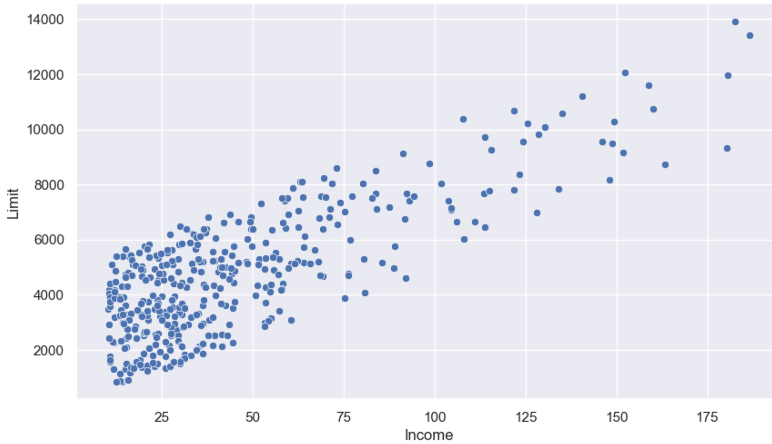
[209]:
```
Index(['Income', 'Limit', 'Rating', 'Cards', 'Age', 'Education', 'Gender',
       'Student', 'Married', 'Ethnicity', 'Balance'],
      dtype='object')
```

Important Variable - Income - Limit - Age - Balance - Rating

[210]:
```
fig, axes = plt.subplots(4, figsize=(10,26))
sns.scatterplot(ax=axes[0],data=df,x="Income",y="Limit")
axes[0].set_title("Income Vs Limit")
sns.scatterplot(ax=axes[1],data=df,x="Age",y="Income")
axes[1].set_title("Age Vs Income")
sns.scatterplot(ax=axes[2],data=df,x="Income",y="Balance")
axes[2].set_title("Income Vs Balance")
sns.scatterplot(ax=axes[3],data=df,x="Rating",y="Income")
axes[2].set_title("Rating Vs Income")
```

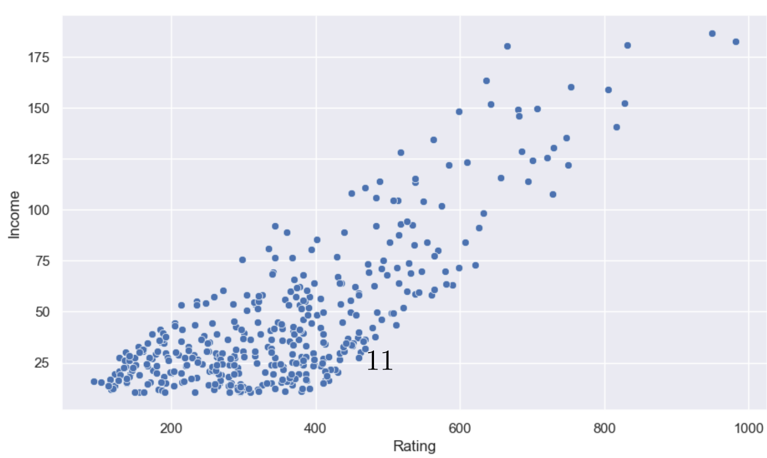
[210]:
```
Text(0.5, 1.0, 'Rating Vs Income')
```

Income Vs Limit



Age Vs Income

Rating Vs Income



Data Preprocessing - Missing value treatment

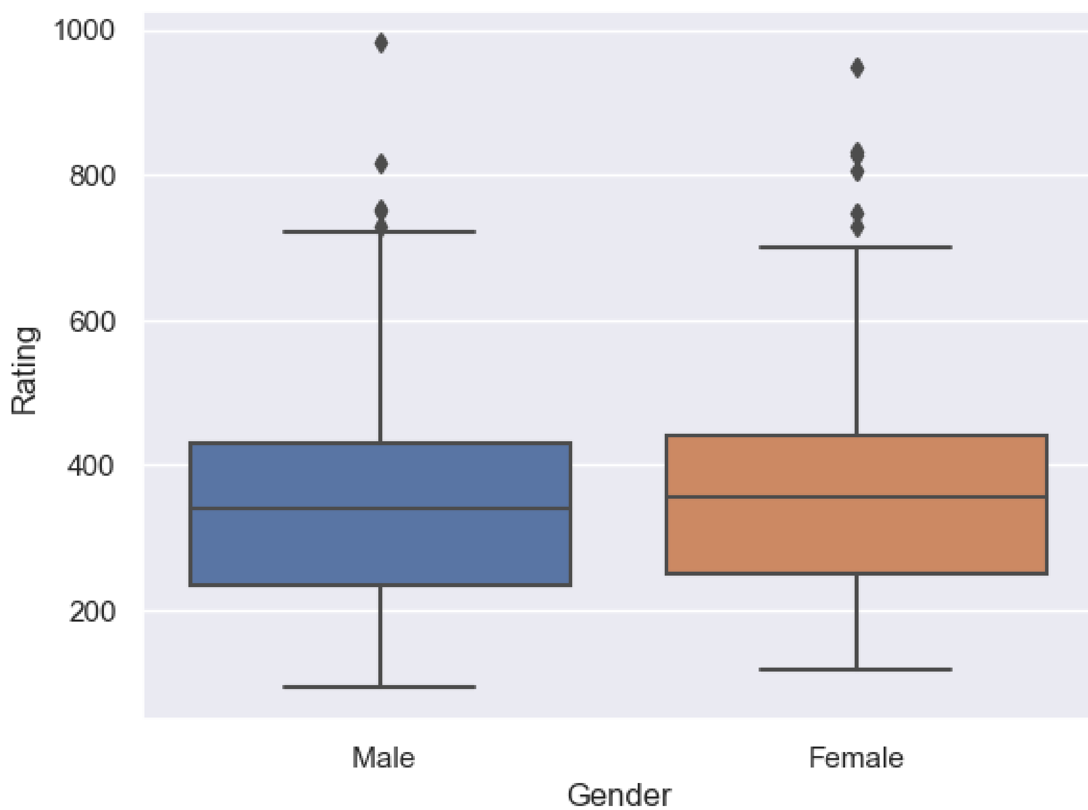
[211]:
```
df.isnull().sum()
```

[211]:
```
Income       0
Limit        0
Rating       0
Cards        0
Age          0
Education    0
Gender       0
Student      0
Married      0
Ethnicity    0
Balance      0
dtype: int64
```

- Outlier treatment

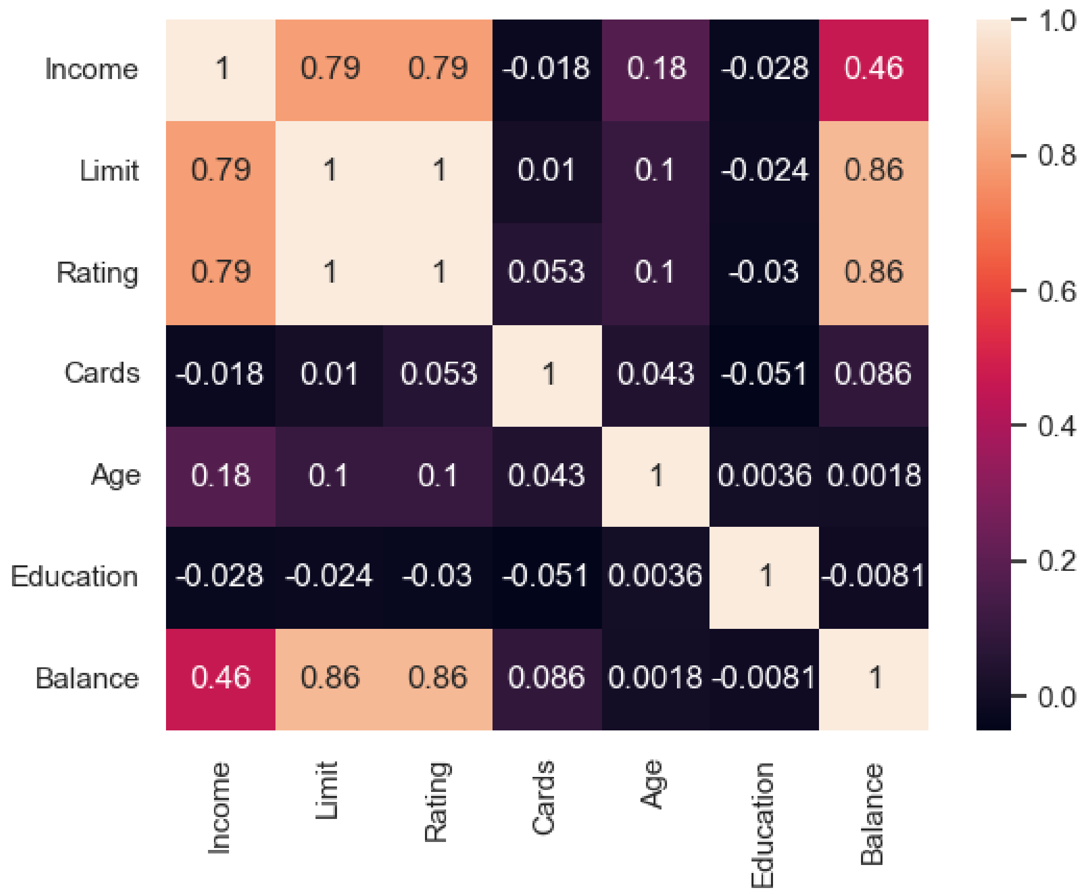
[212]:
```
sns.boxplot(data=df,x="Gender",y="Rating")
```

[212]:
```
<AxesSubplot:xlabel='Gender', ylabel='Rating'>
```



[213]:
```
sns.heatmap(df.corr(),annot=True)
```

[213]: <AxesSubplot:>



- Data preparation for modeling

[214]:
```
df.groupby("Ethnicity")["Income"].mean()
```

[214]:
```
Ethnicity
African American    47.682101
Asian               44.187833
Caucasian           44.521945
Name: Income, dtype: float64
```

[215]:
```
df["Gender"].replace({" Male":1,"Female":0},inplace=True)
df["Student"].replace({"No":0,"Yes":1},inplace=True)
df["Married"].replace({"No":0,"Yes":1},inplace=True)
```

```python
df["Ethnicity"].replace({
    "African American":0,
    "Asian":1,
    "Caucasian":2
},inplace=True)
```

[216]:
```python
x = df.drop("Rating",axis=1)
y = df["Rating"]
```

Model building

- Linear Regression Model

[217]:
```python
x_sm = sm.add_constant(x)
sm_model = sm.OLS(y,x_sm).fit()
print(sm_model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Rating   R-squared:                       0.996
Model:                            OLS   Adj. R-squared:                  0.996
Method:                 Least Squares   F-statistic:                     9205.
Date:                Fri, 28 Jul 2023   Prob (F-statistic):               0.00
Time:                        19:57:49   Log-Likelihood:                -1489.6
No. Observations:                 400   AIC:                             3001.
Df Residuals:                     389   BIC:                             3045.
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         31.3413      4.135      7.579      0.000      23.211      39.472
Income         0.1207      0.047      2.573      0.010       0.028       0.213
Limit          0.0633      0.001     44.361      0.000       0.060       0.066
Cards          4.5976      0.392     11.737      0.000       3.827       5.368
Age            0.0150      0.030      0.495      0.621      -0.045       0.075
Education     -0.2395      0.164     -1.460      0.145      -0.562       0.083
Gender        -0.1797      1.021     -0.176      0.860      -2.187       1.828
Student       -1.9979      2.810     -0.711      0.478      -7.523       3.527
Married        2.2162      1.055      2.101      0.036       0.142       4.290
Ethnicity      0.0254      0.618      0.041      0.967      -1.189       1.240
Balance        0.0116      0.005      2.248      0.025       0.001       0.022
==============================================================================
Omnibus:                        6.286   Durbin-Watson:                   2.071
Prob(Omnibus):                  0.043   Jarque-Bera (JB):                5.628
Skew:                           0.225   Prob(JB):                       0.0600
Kurtosis:                       2.633   Cond. No.                     4.65e+04
==============================================================================
```

```
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 4.65e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

- Multicollinearity check by VIF score

[218]:
```python
vif = pd.DataFrame()
X_t = x
vif['Features'] = X_t.columns
vif['VIF'] = np.round([variance_inflation_factor(X_t.values, i) for i in␣
 ↪range(X_t.shape[1])],2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

[218]:

	Features	VIF
1	Limit	137.36
9	Balance	33.73
0	Income	19.56
4	Education	14.66
3	Age	10.83
2	Cards	5.14
8	Ethnicity	3.14
7	Married	2.59
6	Student	2.34
5	Gender	1.90

[219]:
```python
x1 = x.drop("Education",axis=1)
x1_sm = sm.add_constant(x1)
sm1_model = sm.OLS(y,x1_sm).fit()
print(sm1_model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Rating   R-squared:                       0.996
Model:                            OLS   Adj. R-squared:                  0.996
Method:                 Least Squares   F-statistic:                 1.020e+04
Date:                Fri, 28 Jul 2023   Prob (F-statistic):               0.00
Time:                        19:57:49   Log-Likelihood:                -1490.7
No. Observations:                 400   AIC:                             3001.
Df Residuals:                     390   BIC:                             3041.
Df Model:                           9
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         28.2526      3.558      7.940      0.000      21.257      35.249
Income         0.1241      0.047      2.645      0.008       0.032       0.216
```

```
Limit           0.0632      0.001     44.275      0.000       0.060       0.066
Cards           4.6175      0.392     11.778      0.000       3.847       5.388
Age             0.0144      0.030      0.475      0.635      -0.045       0.074
Gender         -0.2005      1.023     -0.196      0.845      -2.211       1.810
Student        -2.3279      2.805     -0.830      0.407      -7.843       3.187
Married         2.1275      1.055      2.017      0.044       0.054       4.201
Ethnicity       0.0531      0.618      0.086      0.932      -1.163       1.269
Balance         0.0120      0.005      2.309      0.021       0.002       0.022
==============================================================================
Omnibus:                        6.649   Durbin-Watson:                   2.071
Prob(Omnibus):                  0.036   Jarque-Bera (JB):                5.781
Skew:                           0.221   Prob(JB):                       0.0556
Kurtosis:                       2.612   Cond. No.                     4.29e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 4.29e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[220]:
```python
x2 = x1.drop("Age",axis=1)
x2_sm = sm.add_constant(x2)
sm2_model = sm.OLS(y,x2_sm).fit()
print(sm2_model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Rating   R-squared:                       0.996
Model:                            OLS   Adj. R-squared:                  0.996
Method:                 Least Squares   F-statistic:                 1.150e+04
Date:                Fri, 28 Jul 2023   Prob (F-statistic):               0.00
Time:                        19:57:49   Log-Likelihood:                -1490.8
No. Observations:                 400   AIC:                             3000.
Df Residuals:                     391   BIC:                             3036.
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         28.9082      3.277      8.823      0.000      22.466      35.350
Income         0.1239      0.047      2.644      0.009       0.032       0.216
Limit          0.0633      0.001     44.526      0.000       0.060       0.066
Cards          4.6322      0.390     11.864      0.000       3.865       5.400
Gender        -0.2038      1.022     -0.199      0.842      -2.212       1.805
Student       -2.2495      2.797     -0.804      0.422      -7.749       3.250
Married        2.0852      1.050      1.986      0.048       0.021       4.149
Ethnicity      0.0476      0.618      0.077      0.939      -1.167       1.262
```

```
Balance         0.0117      0.005      2.274      0.024       0.002       0.022
==============================================================================
Omnibus:                        6.541   Durbin-Watson:                   2.069
Prob(Omnibus):                  0.038   Jarque-Bera (JB):                5.698
Skew:                           0.220   Prob(JB):                       0.0579
Kurtosis:                       2.614   Cond. No.                     4.16e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 4.16e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[222]:
```
x3 = x2.drop("Balance",axis=1)
x3_sm = sm.add_constant(x3)
sm3_model = sm.OLS(y,x3_sm).fit()
print(sm3_model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Rating   R-squared:                       0.996
Model:                            OLS   Adj. R-squared:                  0.996
Method:                 Least Squares   F-statistic:                 1.300e+04
Date:                Fri, 28 Jul 2023   Prob (F-statistic):               0.00
Time:                        19:58:13   Log-Likelihood:                -1493.4
No. Observations:                 400   AIC:                             3003.
Df Residuals:                     392   BIC:                             3035.
Df Model:                           7
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         22.9547      1.980     11.592      0.000      19.062      26.848
Income         0.0320      0.024      1.342      0.180      -0.015       0.079
Limit          0.0664      0.000    182.414      0.000       0.066       0.067
Cards          4.9026      0.374     13.114      0.000       4.168       5.638
Gender        -0.0806      1.026     -0.079      0.937      -2.097       1.936
Student        2.7924      1.714      1.629      0.104      -0.578       6.163
Married        2.0411      1.055      1.934      0.054      -0.033       4.116
Ethnicity      0.0998      0.621      0.161      0.872      -1.120       1.320
==============================================================================
Omnibus:                        7.256   Durbin-Watson:                   2.070
Prob(Omnibus):                  0.027   Jarque-Bera (JB):                6.214
Skew:                           0.229   Prob(JB):                       0.0447
Kurtosis:                       2.596   Cond. No.                     2.17e+04
==============================================================================
```

```
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 2.17e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[223]:
```
vif = pd.DataFrame()
X_t = x3
vif['Features'] = X_t.columns
vif['VIF'] = np.round([variance_inflation_factor(X_t.values, i) for i in␣
 ↪range(X_t.shape[1])],2)
vif = vif.sort_values(by = "VIF", ascending = False)
vif
```

[223]:

	Features	VIF
1	Limit	11.54
0	Income	7.03
2	Cards	3.97
6	Ethnicity	2.82
5	Married	2.37
3	Gender	1.81
4	Student	1.10

[ ]:
```
standard = StandardScaler()
x = standard.fit_transform(df.drop("Rating",axis=1))
y = standard.fit_transform(df["Rating"].values.reshape(-1,1))
```

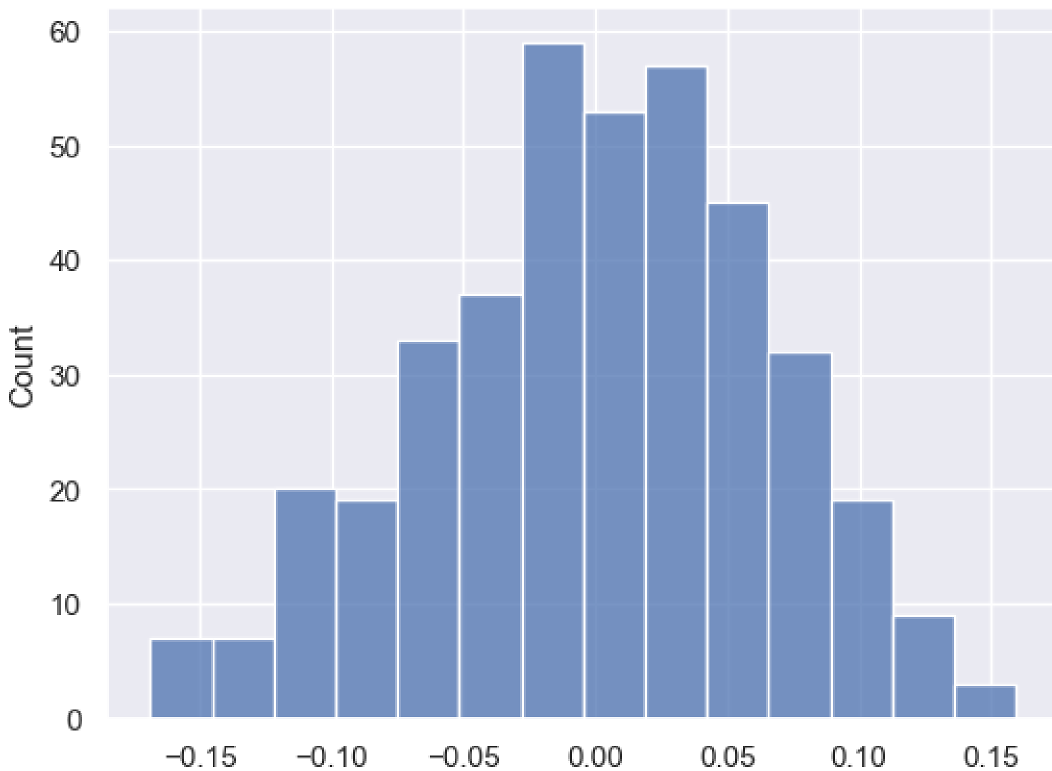
[ ]:
```
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.2,shuffle=True)
```

[ ]:
```
model = LinearRegression()
model.fit(x,y)
y_hat = np.dot(x,model.coef_.T) + model.intercept_
error = y_hat - y
error.mean()
```

[ ]:
```
LinearRegression()
```

[228]:
```
sns.histplot(error.squeeze())
```

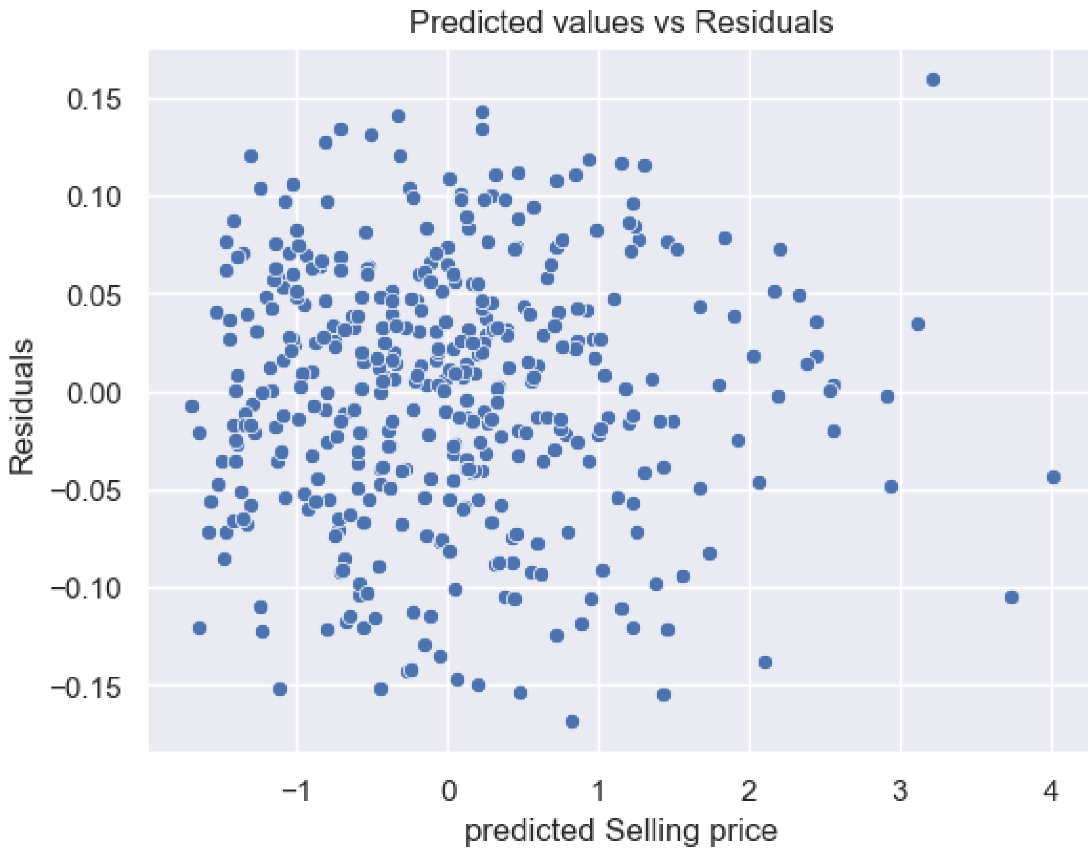
[228]:
```
<AxesSubplot:ylabel='Count'>
```



[ ]:
```
sns.scatterplot(y_hat.squeeze(),error.squeeze())
plt.xlabel("predicted Selling price")
plt.ylabel("Residuals")
plt.title("Predicted values vs Residuals")
```

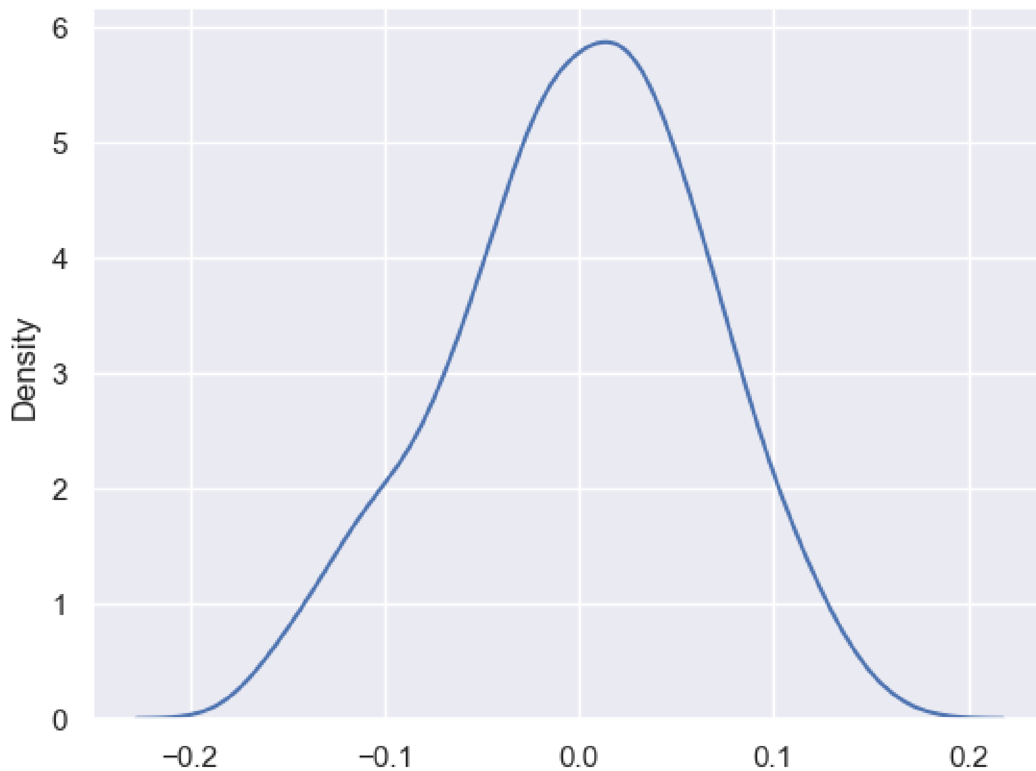
```
c:\Users\revan\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
FutureWarning: Pass the following variables as keyword args: x, y. From version
0.12, the only valid positional argument will be `data`, and passing other
arguments without an explicit keyword will result in an error or
misinterpretation.
  warnings.warn(
```

[ ]: Text(0.5, 1.0, 'Predicted values vs Residuals')



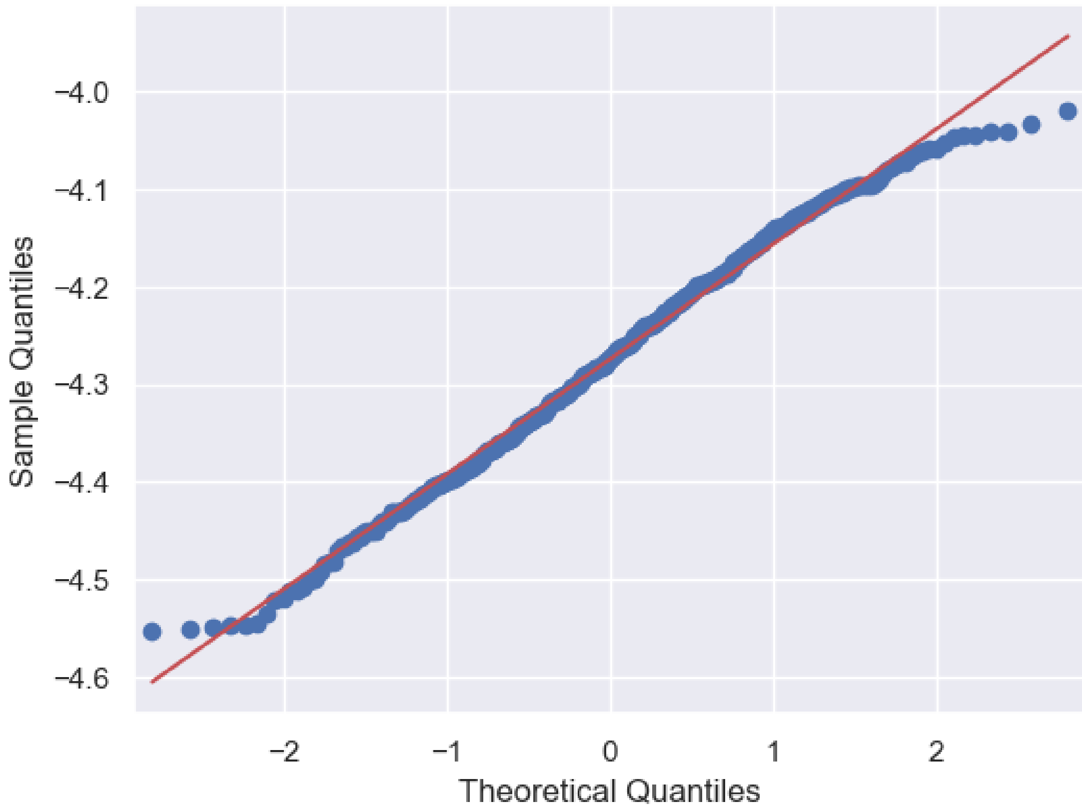
[ ]:
```
sns.kdeplot(error.squeeze())
```

[ ]: <AxesSubplot:ylabel='Density'>

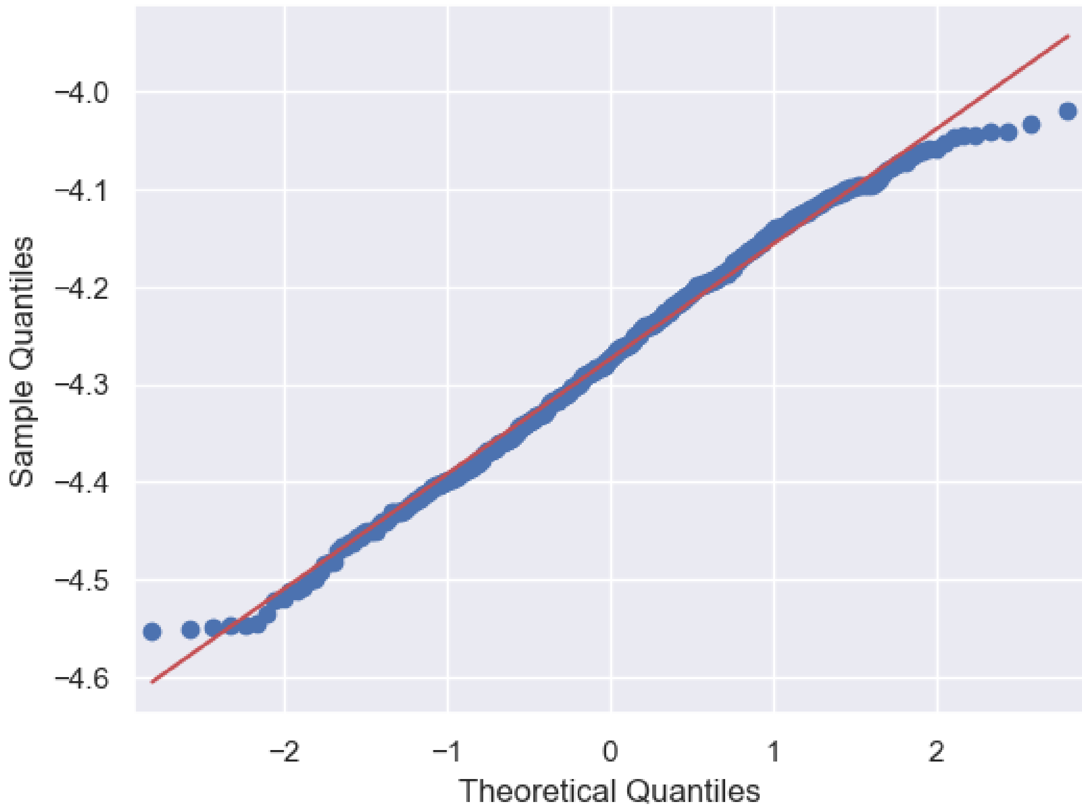


[231]:
```
qqplot(stats.boxcox(np.square(error.squeeze())/len(error.
↪squeeze()))[0],line="s")
```

[231]:



−4.0
−4.1
−4.2
−4.3
−4.4
−4.5
−4.6
Sample Quantiles
−2
−1
0
1
2
Theoretical Quantiles


−4.0
−4.1
−4.2
−4.3
−4.4
−4.5
−4.6
Sample Quantiles
−2
−1
0
1
2
Theoretical Quantiles

```
[ ]:
```