In [1]:
```
import numpy as np
import pandas as pd


import os
for dirname, _, filenames in os.walk(''):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
```

In [3]:
```
test_loan_data = pd.read_csv('data.csv')
train_loan_data = pd.read_csv('data.csv')
```

In [4]:
```
train_loan_data.shape
```

Out[4]:
```
(367, 12)
```

In [5]:
```
train_loan_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 367 entries, 0 to 366
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Loan_ID            367 non-null    object 
 1   Gender             356 non-null    object 
 2   Married            367 non-null    object 
 3   Dependents         357 non-null    object 
 4   Education          367 non-null    object 
 5   Self_Employed      344 non-null    object 
 6   ApplicantIncome    367 non-null    int64  
 7   CoapplicantIncome  367 non-null    int64  
 8   LoanAmount         362 non-null    float64
 9   Loan_Amount_Term   361 non-null    float64
 10  Credit_History     338 non-null    float64
 11  Property_Area      367 non-null    object 
dtypes: float64(3), int64(2), object(7)
memory usage: 34.5+ KB
```

In [6]:
```
train_loan_data.dtypes
```

Out[6]:
```
Loan_ID               object
Gender                object
Married               object
Dependents            object
Education             object
Self_Employed         object
ApplicantIncome        int64
CoapplicantIncome      int64
LoanAmount           float64
Loan_Amount_Term     float64
Credit_History       float64
Property_Area         object
dtype: object
```

In [7]:
```python
train_loan_data.columns
```

Out[7]:
```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area'],
      dtype='object')
```

In [8]:
```python
test_loan_data.columns
```

Out[8]:
```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome', 'LoanAmount',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area'],
      dtype='object')
```

In [9]:
```python
train_loan_data.head(5)
```

Out[9]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	Loa
0	LP001015	Male	Yes	0	Graduate	No	5720	0	
1	LP001022	Male	Yes	1	Graduate	No	3076	1500	
2	LP001031	Male	Yes	2	Graduate	No	5000	1800	
3	LP001035	Male	Yes	2	Graduate	No	2340	2546	
4	LP001051	Male	No	0	Not Graduate	No	3276	0	

In [10]:
```python
train_loan_data.tail(5)
```

Out[10]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	L
362	LP002971	Male	Yes	3+	Not Graduate	Yes	4009	1777	
363	LP002975	Male	Yes	0	Graduate	No	4158	709	
364	LP002980	Male	No	0	Graduate	No	3250	1993	
365	LP002986	Male	Yes	0	Graduate	No	5000	2393	
366	LP002989	Male	No	0	Graduate	Yes	9200	0	

In [11]:
```python
train_loan_data.isnull().sum()
```

Out[11]:
```
Loan_ID               0
Gender               11
Married               0
Dependents           10
Education             0
Self_Employed        23
ApplicantIncome       0
CoapplicantIncome     0
LoanAmount            5
Loan_Amount_Term      6
Credit_History       29
Property_Area         0
dtype: int64
```

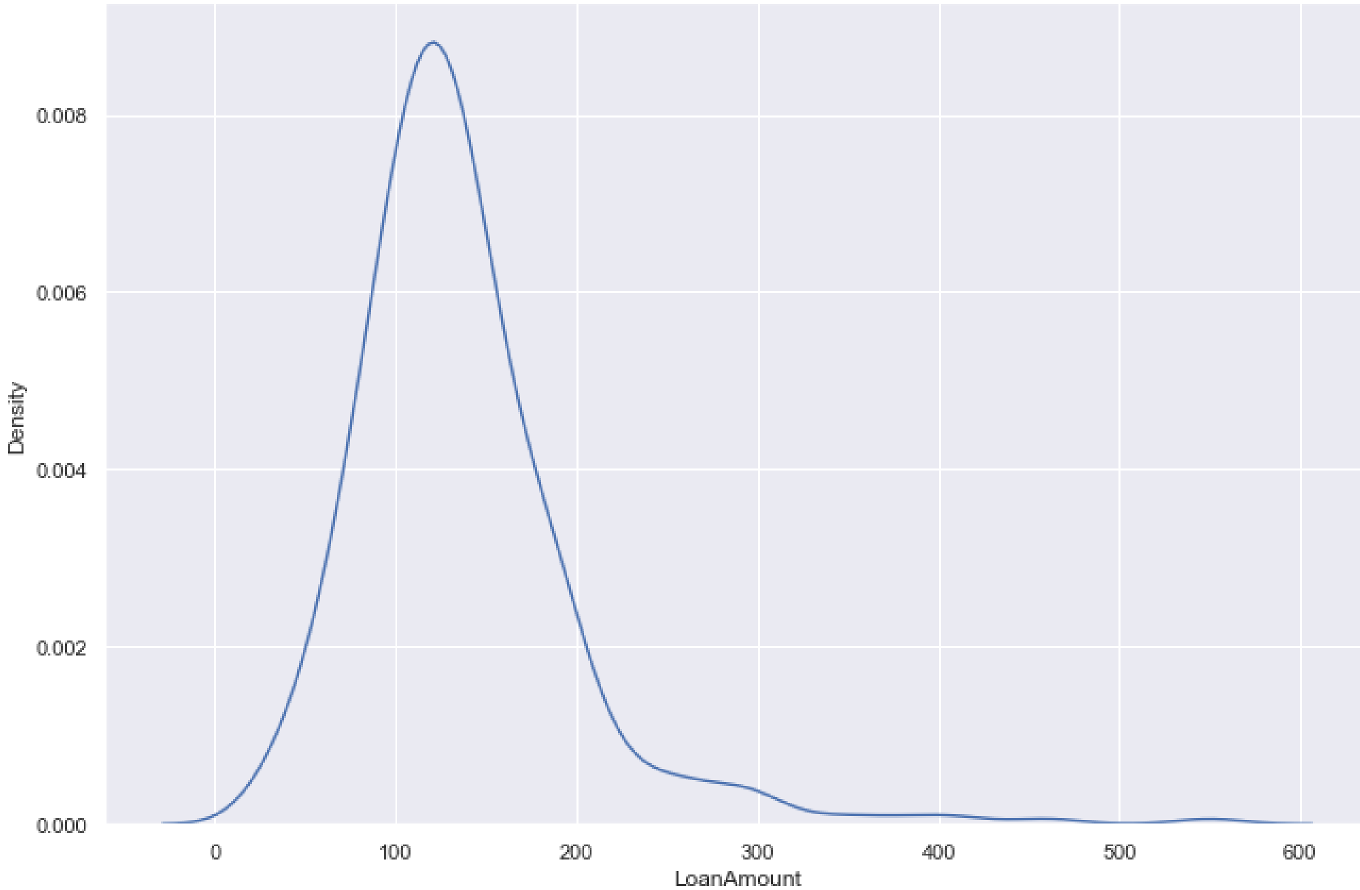
In [12]:
```python
x_data = train_loan_data.drop('LoanAmount', axis='columns')
```

In [13]:
```python
y_data = train_loan_data['LoanAmount']
```

In [14]:
```python
plt.figure(figsize=(12,8))
```

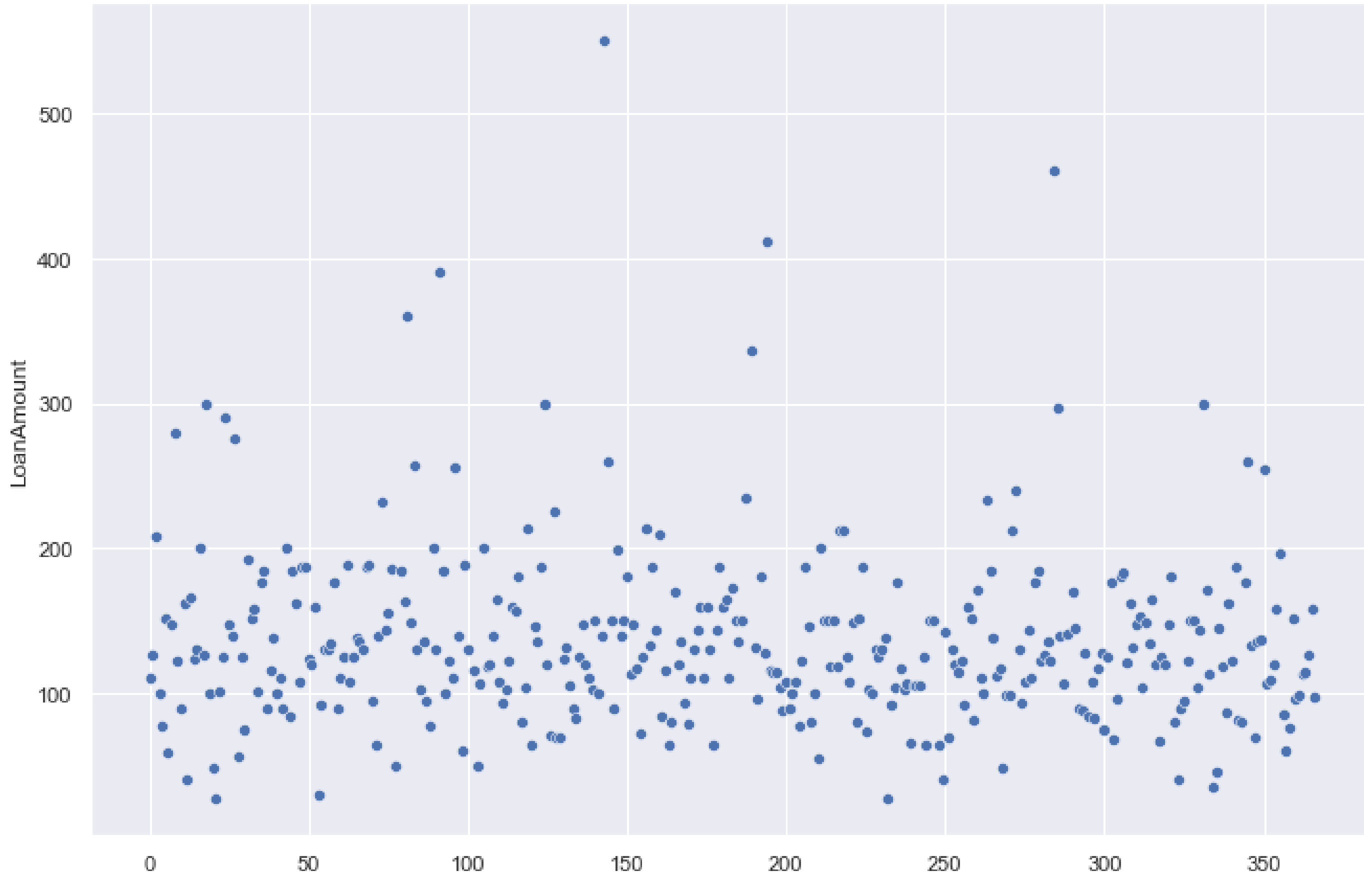
```
sns.set(style='darkgrid')
sns.kdeplot(y_data)
```

Out[14]: <AxesSubplot:xlabel='LoanAmount', ylabel='Density'>



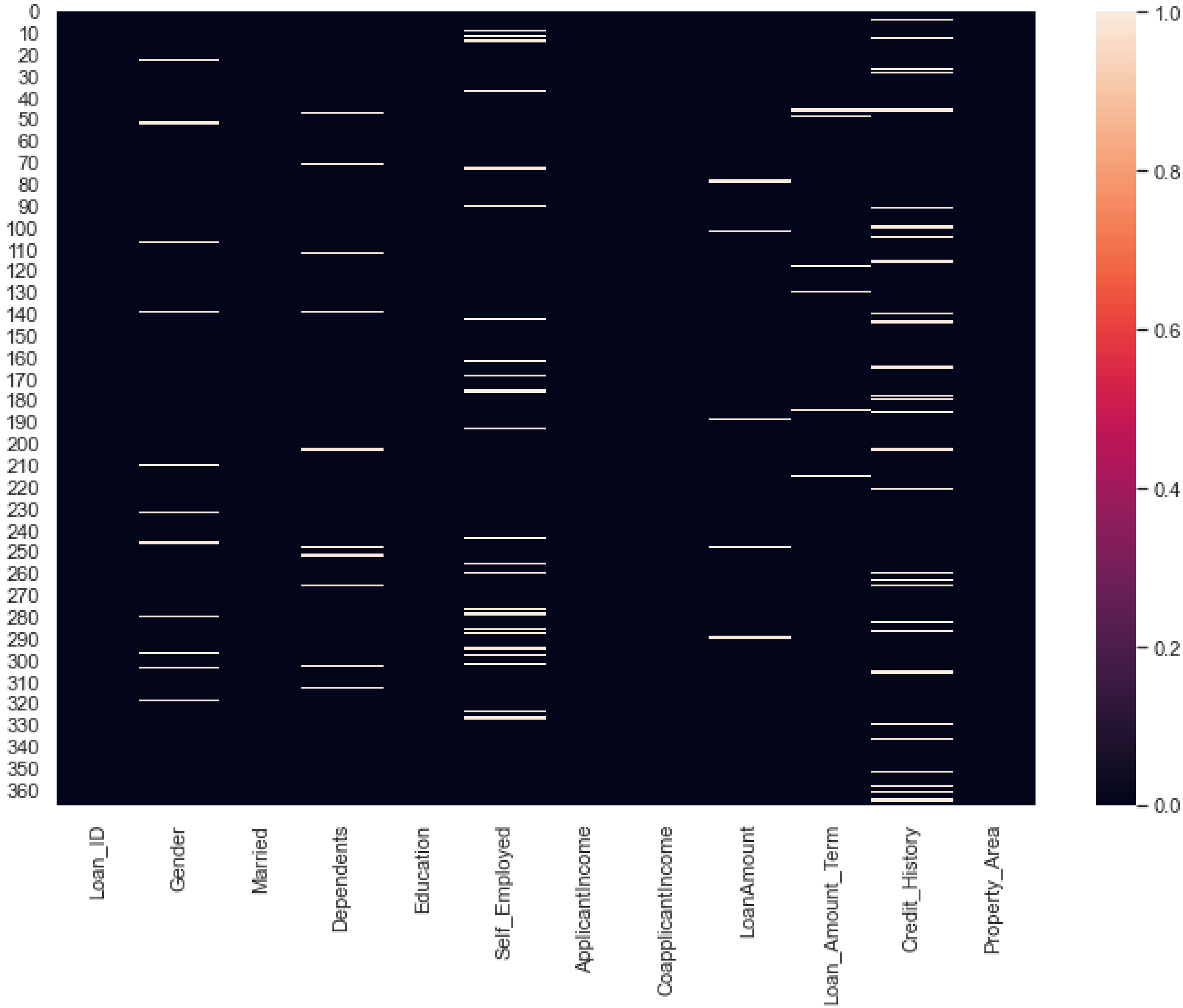
In [15]:
```
plt.figure(figsize=(12,8))
sns.scatterplot(data=y_data)
```

Out[15]: <AxesSubplot:ylabel='LoanAmount'>



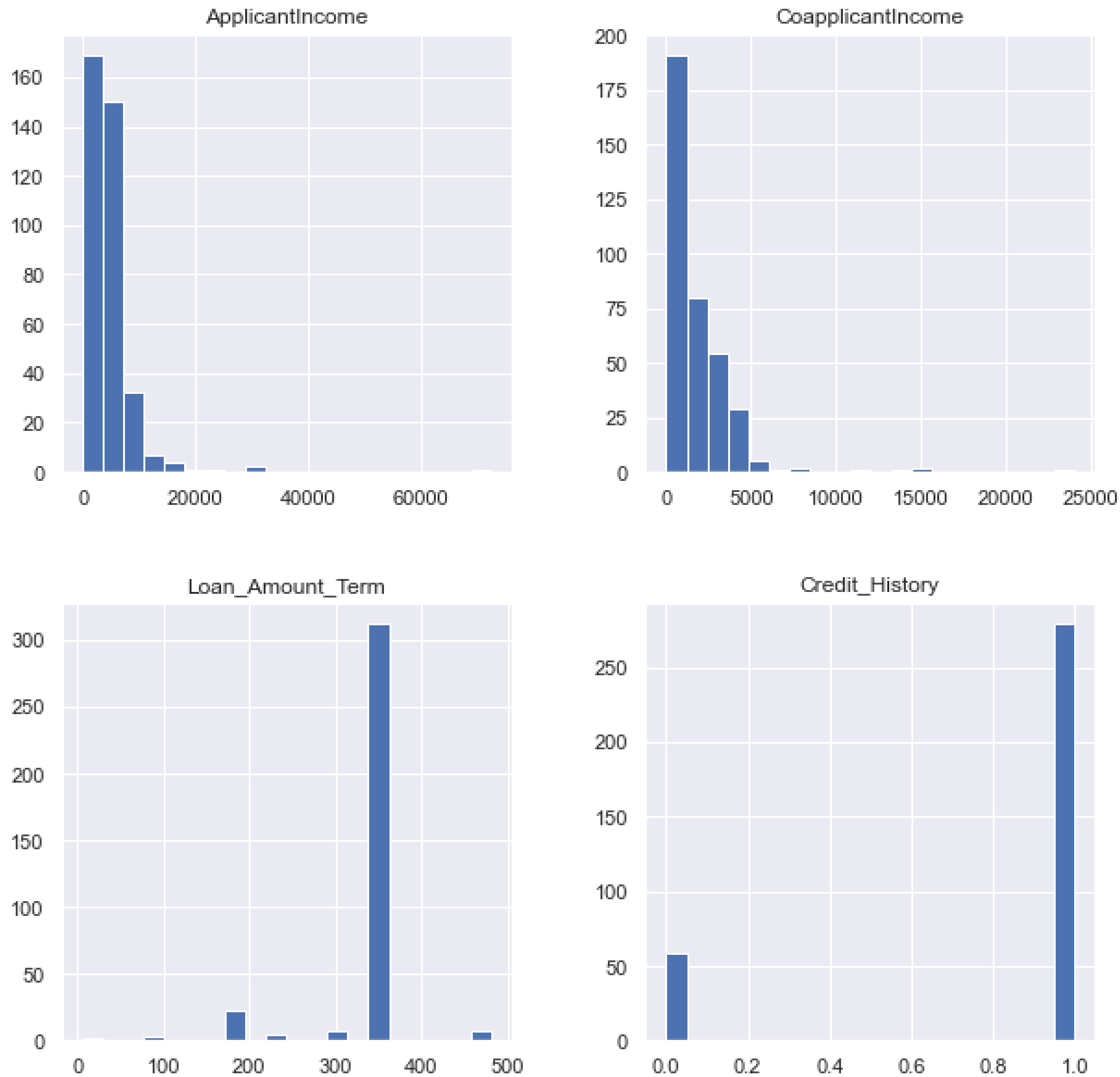
In [16]:
```
plt.figure(figsize=(12,8))
sns.heatmap(train_loan_data.isnull())
```

Out[16]:
```
<AxesSubplot:>
```



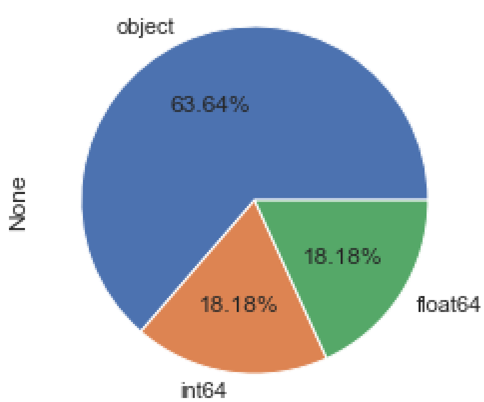
In [17]:

```
x_data.hist(figsize = (10, 10), bins = 20, legend = False)
plt.show()
```



In [18]:
```
x_data.dtypes.value_counts().plot.pie(autopct='%0.2f%%')
```

Out[18]: <AxesSubplot:ylabel='None'>



In [19]:
```
obj_col = x_data.select_dtypes(include='object').columns
int_col = x_data.select_dtypes(include='int').columns
flt_col = x_data.select_dtypes(include='float').columns
```

In [20]:
```
le = LabelEncoder()
```

In [21]:
```
for obj in obj_col:
```

```
    x_data[obj] = le.fit_transform(x_data[obj].astype(str))
```

In [22]:
```
for nt in int_col:
    x_data[nt] = le.fit_transform(x_data[nt].astype(int))
```

In [23]:
```
for flt in flt_col:
    x_data[flt] = le.fit_transform(x_data[flt].astype(float))
```

In [24]:
```
x_data
```

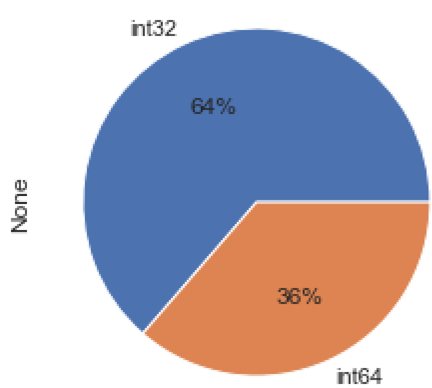
Out[24]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	Lo
0	0	1	1	0	0	0	251	0	
1	1	1	1	1	0	0	102	43	
2	2	1	1	2	0	0	230	67	
3	3	1	1	2	0	0	43	110	
4	4	1	0	0	1	0	120	0	
...	...	...	...	...	...	...	...	...	
362	362	1	1	3	1	1	177	65	
363	363	1	1	0	0	0	186	12	
364	364	1	0	0	0	0	117	75	
365	365	1	1	0	0	0	230	101	
366	366	1	0	0	0	1	292	0	

367 rows × 11 columns

In [25]:
```
x_data.dtypes.value_counts().plot.pie(autopct='%2.0f%%')
```

Out[25]: <AxesSubplot:ylabel='None'>

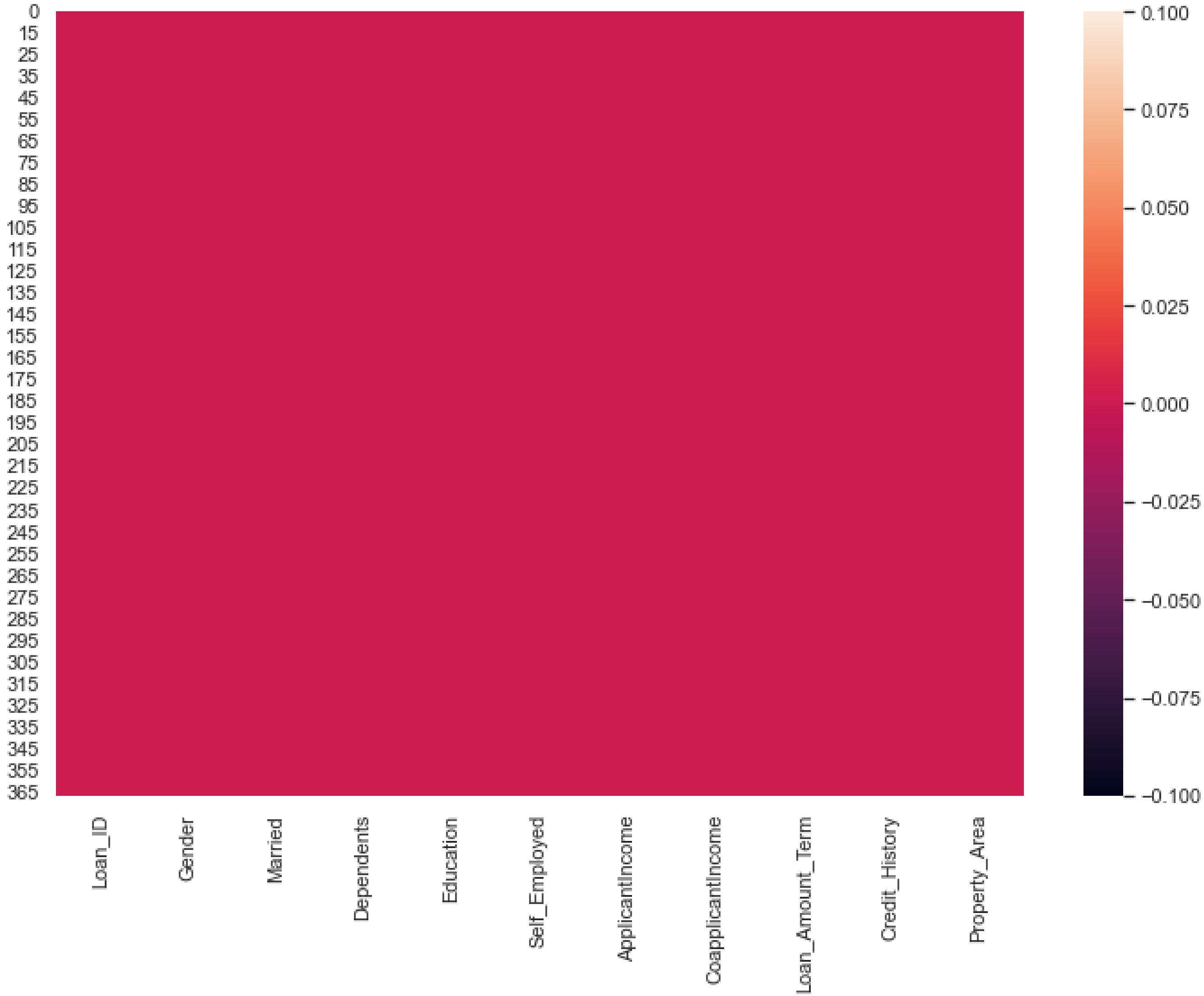


In [26]:
```
x_data = x_data.dropna()
y_data = y_data.dropna()
```

In [27]:
```
x_data = x_data.drop([5,6,1,4,7])
```

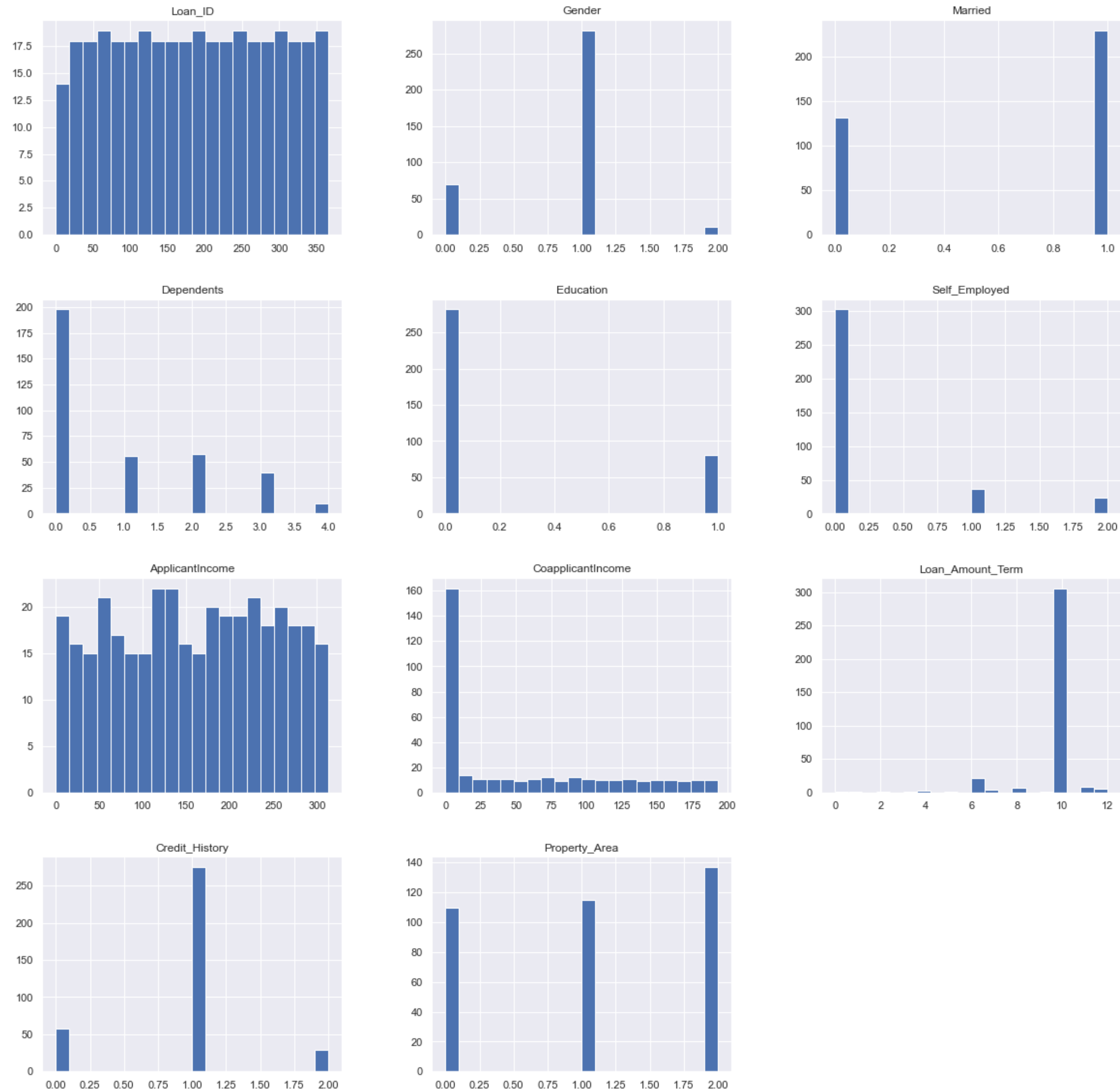
In [28]:
```
plt.figure(figsize=(12,8))
sns.heatmap(x_data.isnull())
```

Out[28]: <AxesSubplot:>



In [29]:

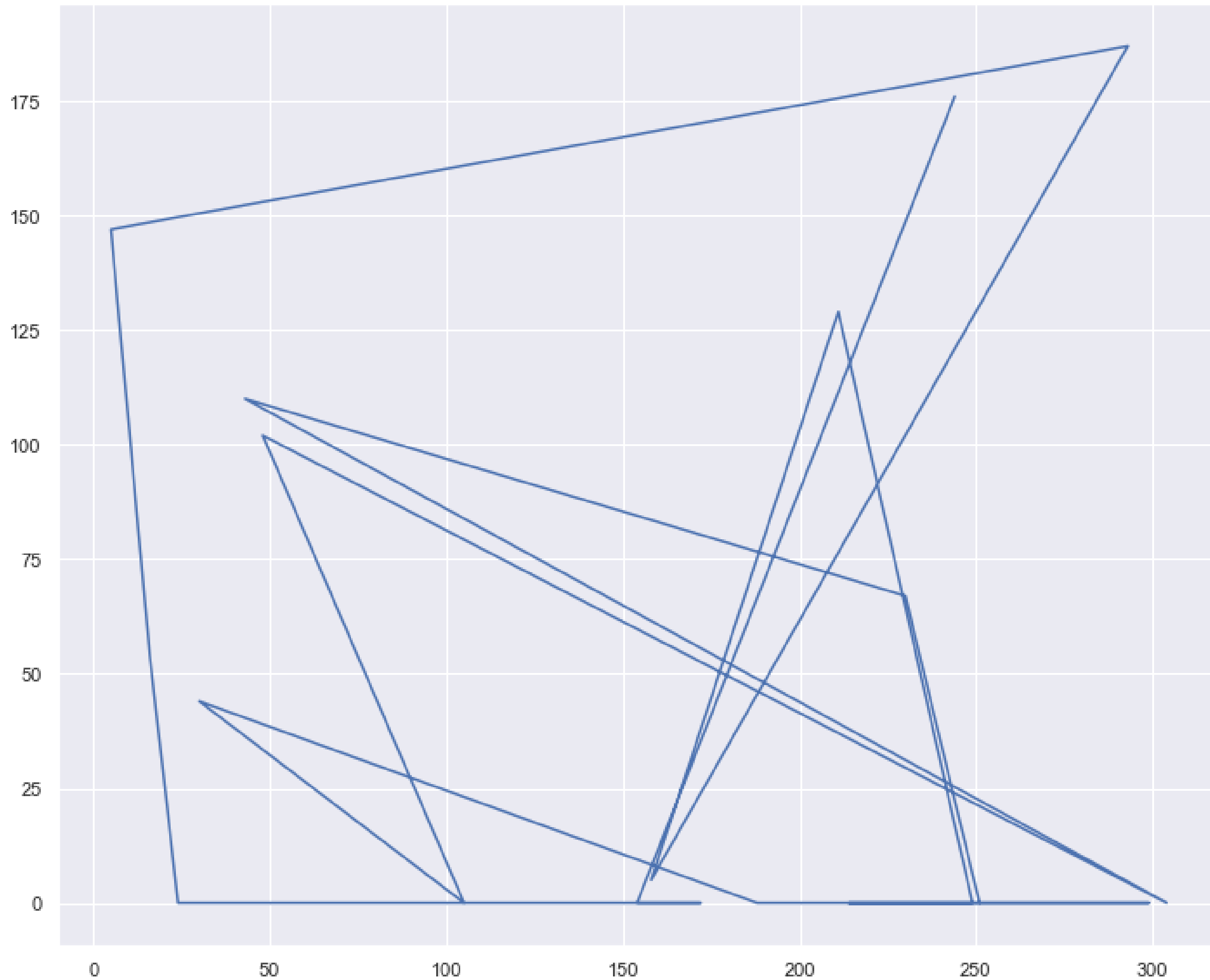
```
x_data.hist(figsize = (20, 20), bins = 20, legend = False)
plt.show()
```



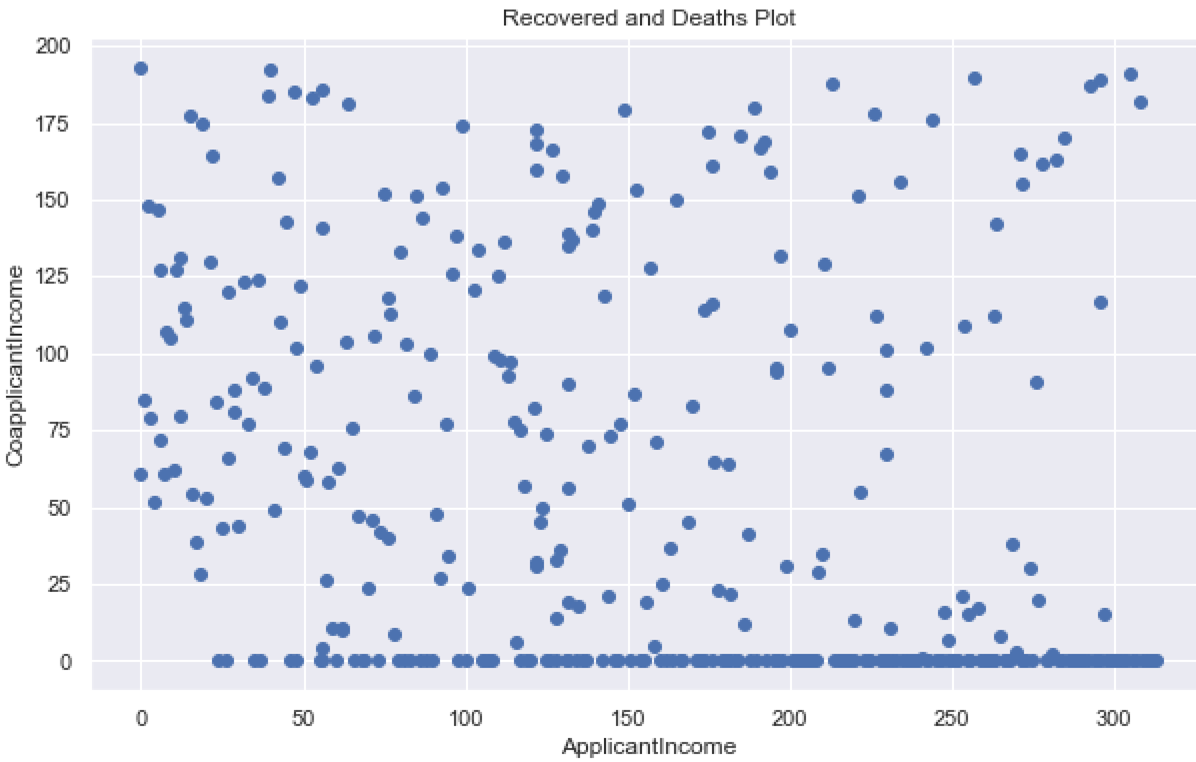
In [30]:
```
x_data.columns
```

Out[30]:
```
Index(['Loan_ID', 'Gender', 'Married', 'Dependents', 'Education',
       'Self_Employed', 'ApplicantIncome', 'CoapplicantIncome',
       'Loan_Amount_Term', 'Credit_History', 'Property_Area'],
      dtype='object')
```

In [31]:
```
plt.figure(figsize=(12,10))
plt.plot(x_data['ApplicantIncome'][0:20], x_data['CoapplicantIncome'][0:20])
plt.show()
```



In [32]:
```
plt.figure(figsize=(10, 6))
plt.plot(x_data['ApplicantIncome'], x_data['CoapplicantIncome'], 'o')
plt.title("Recovered and Deaths Plot")
plt.xlabel("ApplicantIncome")
plt.ylabel("CoapplicantIncome")
plt.show()
```



In [33]:
```
x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_size=0.1)
```

In [34]:
```
rmr = RandomForestRegressor()
```

In [35]:
```
x_test
```

Out[35]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	Loa
178	178	1	1	2	0	0	235	0	
126	126	1	0	0	0	0	263	0	
25	25	1	0	0	0	0	0	193	
262	262	1	0	0	0	0	128	14	
12	12	1	0	3	0	0	188	0	
203	203	0	1	0	0	0	109	99	
111	111	1	1	4	0	0	180	0	
293	293	0	1	0	0	0	33	77	
106	106	2	0	0	0	0	7	61	
261	261	1	1	0	0	0	91	48	
133	133	0	0	1	0	0	142	0	
154	154	1	1	2	0	0	225	0	
365	365	1	1	0	0	0	230	101	
280	280	1	0	0	0	0	29	81	
157	157	1	1	0	0	0	196	94	
128	128	1	0	0	0	0	100	0	
10	10	1	0	0	1	0	105	0	
277	277	0	0	0	0	0	252	0	

135	135	1	1	0	0	0	193	0
328	328	1	1	3	0	1	269	38
62	62	0	0	2	0	0	228	0
80	80	1	1	3	0	0	161	25
243	243	1	1	0	0	2	58	58
116	116	0	0	0	1	0	0	61
345	345	1	1	3	0	0	283	0
282	282	0	0	0	0	0	246	0
190	190	0	1	1	0	0	196	0
232	232	1	1	0	0	0	4	52
29	29	1	0	0	0	0	89	100
117	117	1	1	1	0	0	6	127
225	225	0	0	0	0	0	60	0
184	184	1	1	3	0	0	297	15
242	242	1	1	0	0	0	135	18
21	21	0	0	3	1	0	24	0
359	359	1	0	0	0	0	176	116
89	89	0	0	0	0	2	232	0
255	255	1	0	0	0	2	79	0

In [36]:
```
rmr.fit(x_train, y_train)
```
Out[36]:
```
RandomForestRegressor()
```

In [37]:
```
rmr.predict(x_test)
```
Out[37]:
```
array([123.97, 152.11, 195.43, 124.46,  99.38, 109.8 , 125.94, 101.39,
       140.74, 137.7 , 138.75, 155.12, 122.57, 112.94, 130.84, 154.76,
       201.32, 144.5 , 140.06, 174.09, 147.28, 129.55, 118.39, 152.9 ,
       226.  , 134.81, 169.18,  84.73, 115.06, 119.62, 115.18, 142.26,
       103.32, 163.09, 124.  , 170.28, 100.74])
```

In [38]:
```
test_loan_data
```
Out[38]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	Lo
0	LP001015	Male	Yes	0	Graduate	No	5720	0	
1	LP001022	Male	Yes	1	Graduate	No	3076	1500	
2	LP001031	Male	Yes	2	Graduate	No	5000	1800	
3	LP001035	Male	Yes	2	Graduate	No	2340	2546	
4	LP001051	Male	No	0	Not Graduate	No	3276	0	
...	...	...	...	...	...	...	...	...	
362	LP002971	Male	Yes	3+	Not Graduate	Yes	4009	1777	

363	LP002975	Male	Yes	0	Graduate	No	4158	709
364	LP002980	Male	No	0	Graduate	No	3250	1993
365	LP002986	Male	Yes	0	Graduate	No	5000	2393
366	LP002989	Male	No	0	Graduate	Yes	9200	0

367 rows × 12 columns

In [ ]: