13_Outlier

13.1 Detecting Outlier

In [3]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
dataset = pd.read_csv('loan.csv')
dataset.head(3)
```

Out[2]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	C
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	

In [4]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            614 non-null    object
 1   Gender             601 non-null    object
 2   Married            611 non-null    object
 3   Dependents         599 non-null    object
 4   Education          614 non-null    object
 5   Self_Employed      582 non-null    object
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         592 non-null    float64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object
 12  Loan_Status        614 non-null    object
dtypes: float64(4), int64(1), object(8)
memory usage: 62.5+ KB
```

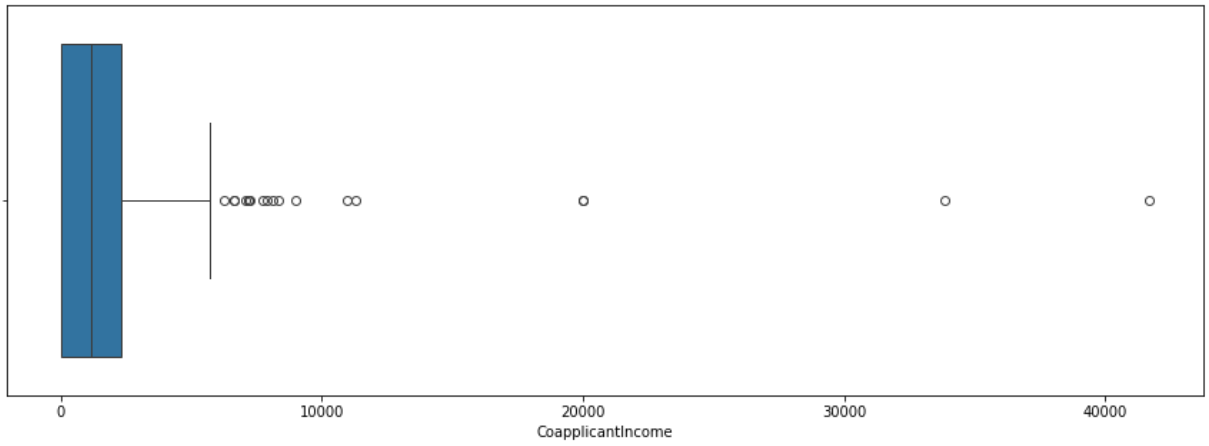
In [6]:
```
dataset.describe()
```

Out[6]:

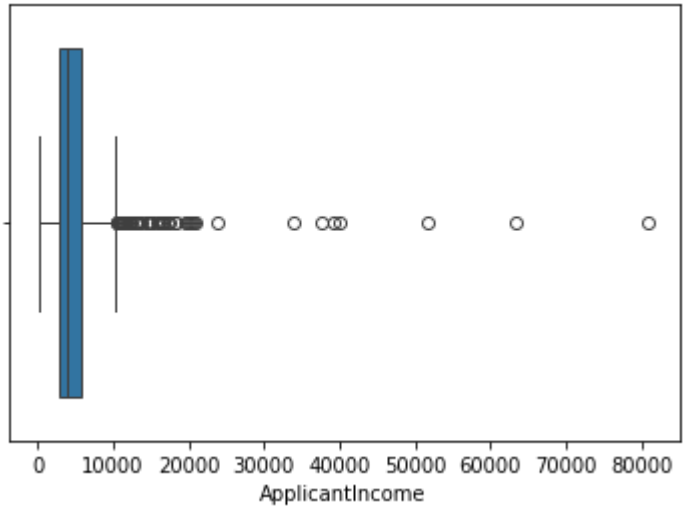
	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_Histo
count	614.000000	614.000000	592.000000	600.00000	564.0000
mean	5403.459283	1621.245798	146.412162	342.00000	0.8421
std	6109.041673	2926.248369	85.587325	65.12041	0.3648
min	150.000000	0.000000	9.000000	12.00000	0.0000
25%	2877.500000	0.000000	100.000000	360.00000	1.0000
50%	3812.500000	1188.500000	128.000000	360.00000	1.0000
75%	5795.000000	2297.250000	168.000000	360.00000	1.0000
max	81000.000000	41667.000000	700.000000	480.00000	1.0000

Detect Outlier through Boxplot

In [16]:
```
plt.figure(figsize=(15,5))
sns.boxplot(x='CoapplicantIncome', data=dataset)
plt.show()
```

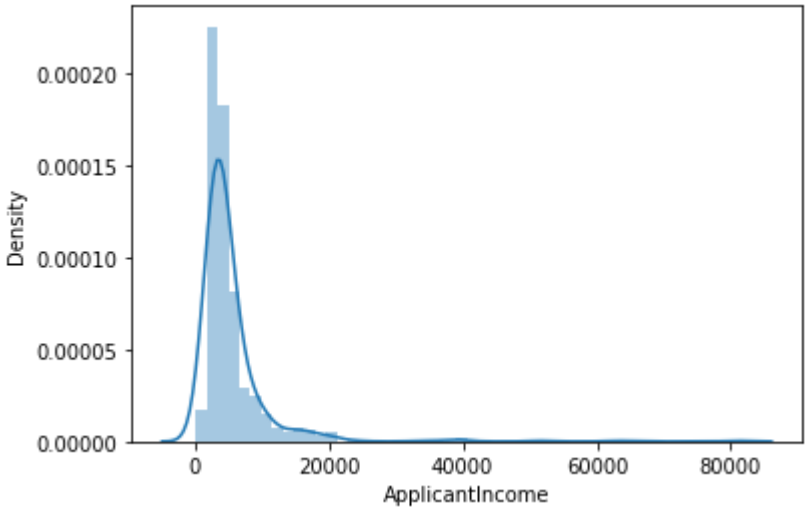


In [8]:
```
sns.boxplot(x='ApplicantIncome', data=dataset)
plt.show()
```



In [9]:
```
sns.distplot(dataset['ApplicantIncome'])
plt.show()
```

```
C:\Users\rashi\AppData\Local\Temp/ipykernel_8588/1976060950.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(dataset['ApplicantIncome'])
```



You can see that tail is too long, so definitely outlier is persent in this

13.2 Removing Outlier

There are two methods for removing outlier:

1. IQR (Inter Quartile Range) method

2. Z-Score method

13.2.1 Removing Outlier through IQR Method

In [11]:
```
dataset.shape
```

Out[11]: (614, 13)

In [13]:
```
q1 = dataset['CoapplicantIncome'].quantile(0.25)
q3 = dataset['CoapplicantIncome'].quantile(0.75)
q1, q3
```

Out[13]: (0.0, 2297.25)

In [14]:
```
IQR = q3 - q1
IQR
```

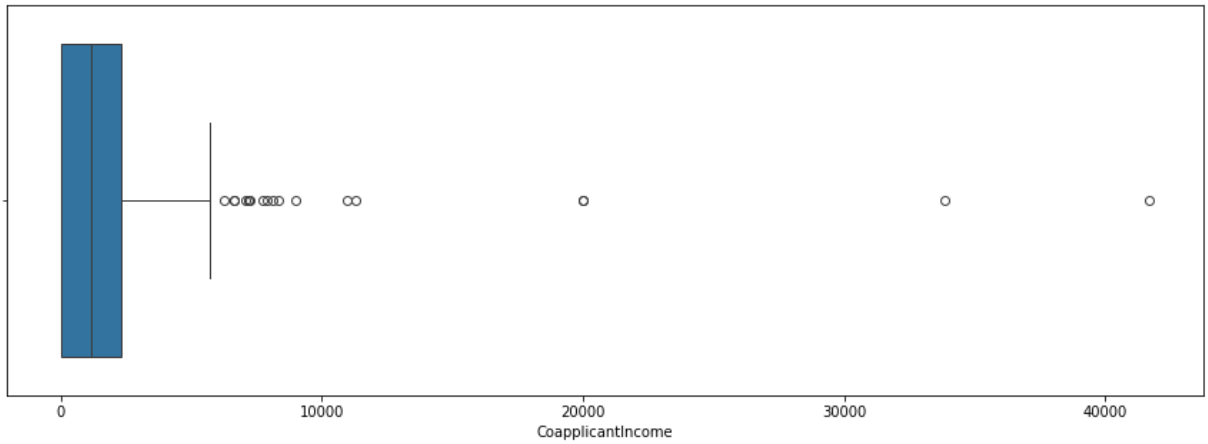
Out[14]: 2297.25

In [15]:
```
min_range = q1 - (1.5*IQR)
max_range = q3 + (1.5*IQR)
min_range, max_range
```

Out[15]: (-3445.875, 5743.125)

We will discard min_range as it is in negative while our data does not contain negative value. max_range is about 5000 as evident in graph below

In [17]:
```
plt.figure(figsize=(15,5))
sns.boxplot(x='CoapplicantIncome', data=dataset)
plt.show()
```



So now we will remove the outlier from the data

In [18]:
```
dataset.head(3)
```

Out[18]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	C
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	

In [19]:
```
dataset['CoapplicantIncome'] < max_range
```

Out[19]:
```
0      True
1      True
2      True
3      True
4      True
       ...
609    True
610    True
611    True
612    True
613    True
Name: CoapplicantIncome, Length: 614, dtype: bool
```

In [23]:
```
dataset.shape
```

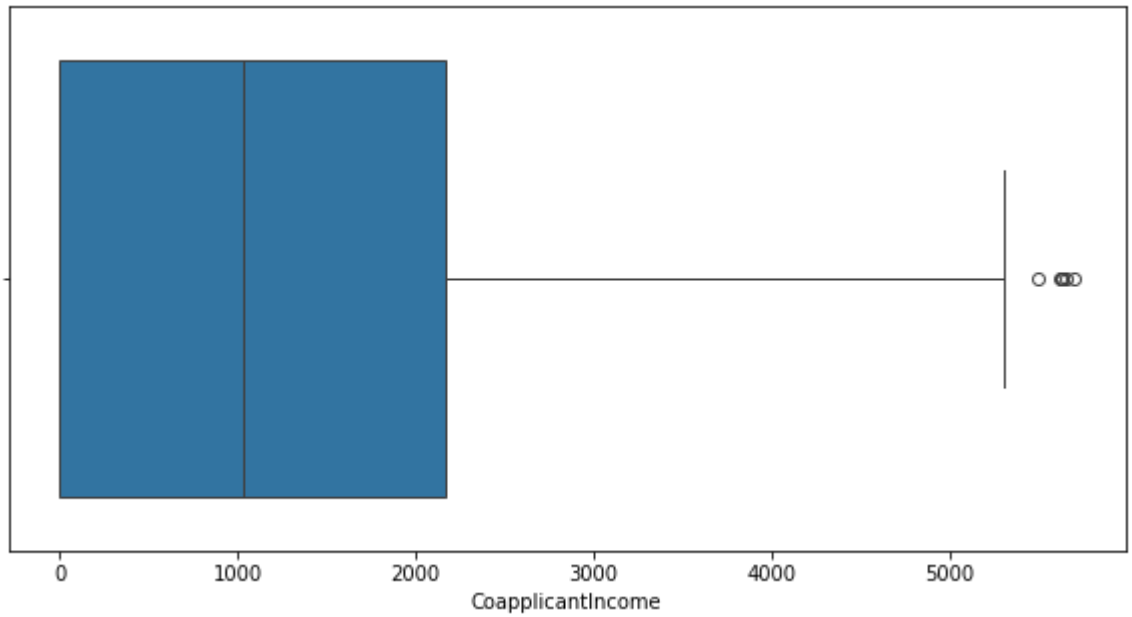
Out[23]:
```
(614, 13)
```

In [22]:
```
new_dataset = dataset[dataset['CoapplicantIncome'] < max_range]
new_dataset.shape
```

Out[22]:
```
(596, 13)
```

It means that 18 rows are removed which were containing outlier in new_dataset

In [26]:
```
plt.figure(figsize=(10,5))
sns.boxplot(x='CoapplicantIncome', data=new_dataset)
plt.show()
```



So number of outliers have been decreased significantly

Outliers may contain essential data so be careful in removing outliter. ML methods like decision tree is not affected by outlier, so you may keep outlier when using decision tree. Linear regression is very affected by outlier so you should remove outlier when using linear regression, but be careful you must not lose essential data

13.2.2 Removing Outlier through Z-Score Method 1

In [28]:
```
dataset.isnull().sum()
```

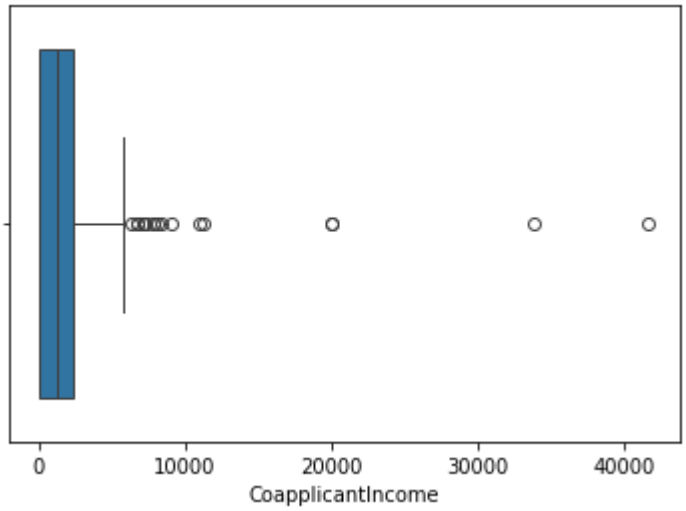
Out[28]:
```
Loan_ID              0
Gender              13
Married              3
Dependents          15
Education            0
Self_Employed       32
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount          22
Loan_Amount_Term    14
Credit_History      50
Property_Area        0
Loan_Status          0
dtype: int64
```

In [29]:
```
dataset.describe()
```

Out[29]:

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_Histo
count	614.000000	614.000000	592.000000	600.00000	564.0000
mean	5403.459283	1621.245798	146.412162	342.00000	0.8421
std	6109.041673	2926.248369	85.587325	65.12041	0.3648
min	150.000000	0.000000	9.000000	12.00000	0.0000
25%	2877.500000	0.000000	100.000000	360.00000	1.0000
50%	3812.500000	1188.500000	128.000000	360.00000	1.0000
75%	5795.000000	2297.250000	168.000000	360.00000	1.0000
max	81000.000000	41667.000000	700.000000	480.00000	1.0000

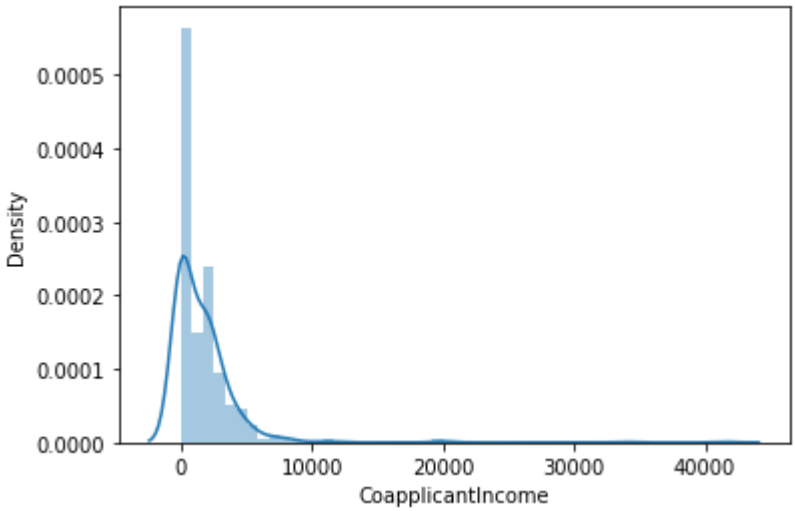
In [30]:
```
sns.boxplot(x='CoapplicantIncome', data=dataset)
plt.show()
```



In [31]:
```
sns.distplot(dataset['CoapplicantIncome'])
```

```
C:\Users\rashi\AppData\Local\Temp/ipykernel_8588/4274022579.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(dataset['CoapplicantIncome'])
```

Out[31]:
```
<Axes: xlabel='CoapplicantIncome', ylabel='Density'>
```



In [36]:
```
min_range = dataset['CoapplicantIncome'].mean() - (3*dataset['CoapplicantIncome'].s
max_range = dataset['CoapplicantIncome'].mean() + (3*dataset['CoapplicantIncome'].s
min_range, max_range
```

Out[36]: (-7157.4993096454655, 10399.990905699668)

- So will ignore min_range b/c its value is negative and our data doesn't contain any -ve value, so will ignore it
- We will take max_range and remove the data greater than this

In [43]:
```
new_dataset_z = dataset[dataset['CoapplicantIncome'] <= max_range]
```

In [44]:
```
dataset.shape, new_dataset_z.shape
```

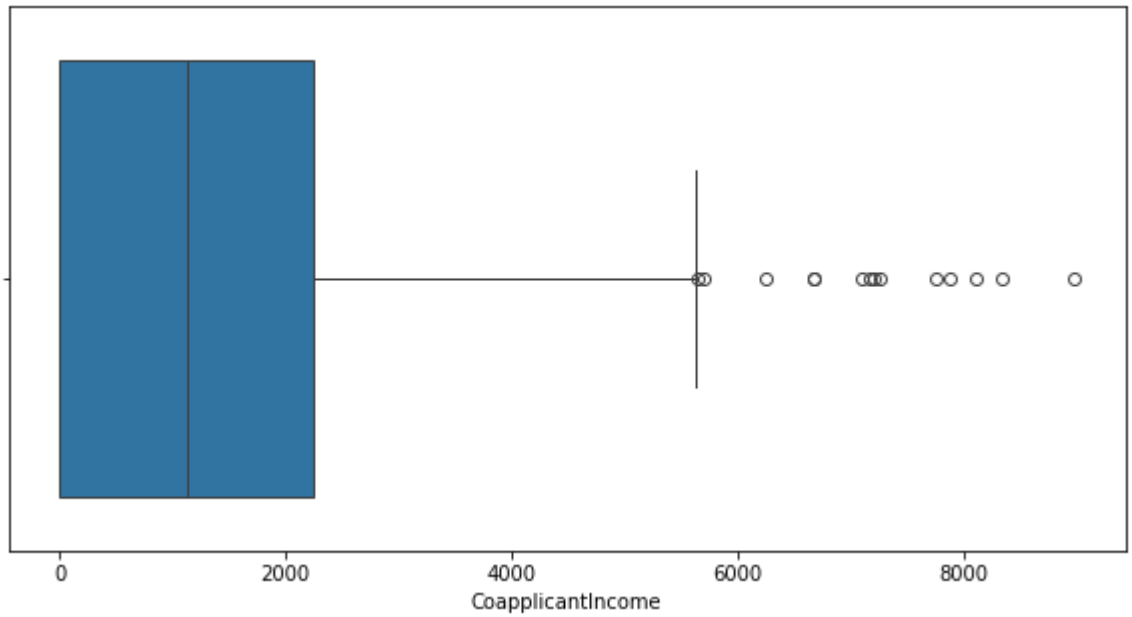
Out[44]: ((614, 13), (608, 13))

In [45]:
```
614-608
```

Out[45]: 6

So You can see 6 rows are deleted containing outliers

In [46]:
```
plt.figure(figsize=(10,5))
sns.boxplot(x='CoapplicantIncome', data=new_dataset_z)
plt.show()
```



13.2.3 Removing Outlier through Z-Score Method 2

In [48]:
```
# Formula of z_score
z_score = (dataset['CoapplicantIncome'] - dataset['CoapplicantIncome'].mean())/data
z_score
```

Out[48]:
```
0     -0.554036
1     -0.038700
2     -0.554036
3      0.251774
4     -0.554036
         ...
609   -0.554036
610   -0.554036
611   -0.472019
612   -0.554036
613   -0.554036
Name: CoapplicantIncome, Length: 614, dtype: float64
```

In [52]:
```
dataset['Z_score'] = z_score
dataset.head(3)
```

Out[52]:

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	C
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	

In [59]:
```
dataset['Z_score']
```

Out[59]:
```
0     -0.554036
1     -0.038700
2     -0.554036
3      0.251774
4     -0.554036
         ...
609   -0.554036
610   -0.554036
611   -0.472019
612   -0.554036
613   -0.554036
Name: Z_score, Length: 614, dtype: float64
```

In [60]:
```
# new_dataset_z = dataset[dataset['CoapplicantIncome'] <= max_range]
new_dataset_z_2 = dataset[dataset['Z_score'] < 3]
new_dataset_z_2.shape
```

Out[60]:
```
(608, 14)
```

So both method 1 and method 2 for removing outlier by z-score are equal

In [ ]:
```

```