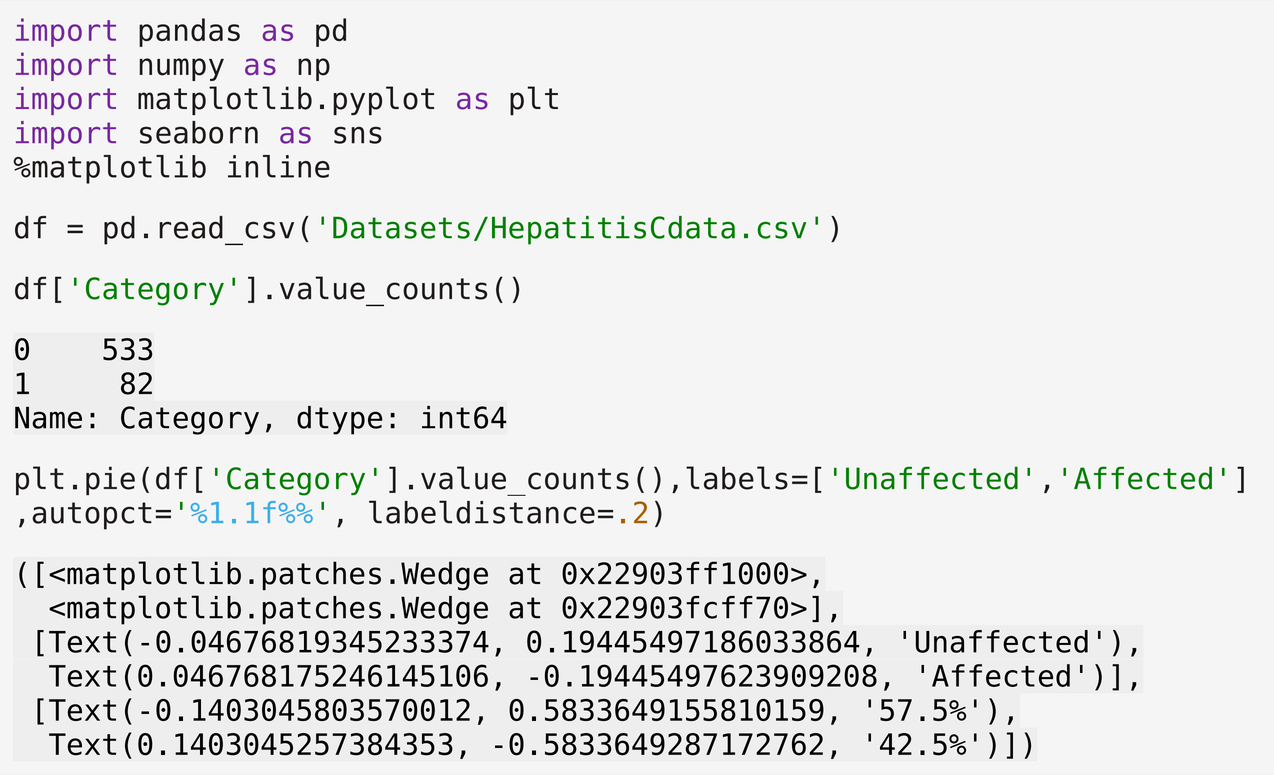
Day - 17 ____________________ #100DaysOfML



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

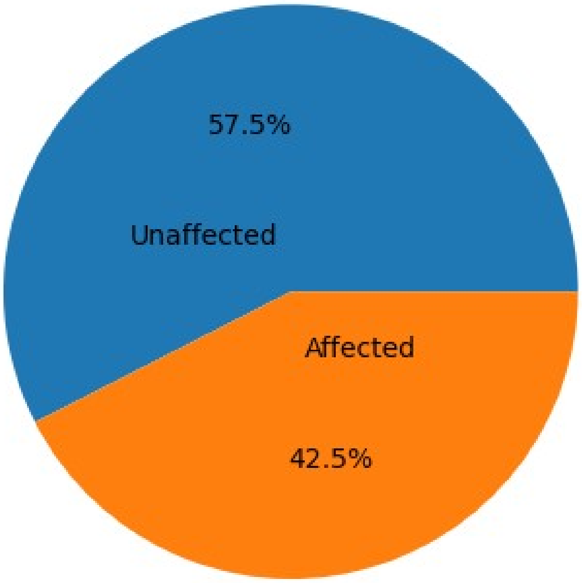
```
df = pd.read_csv('Datasets/HepatitisCdata.csv')
```

```
df['Category'].value_counts()
```

```
0    533
1     82
Name: Category, dtype: int64
```

```
plt.pie(df['Category'].value_counts(),labels=['Unaffected','Affected']
,autopct='%1.1f%%', labeldistance=.2)
```

```
([<matplotlib.patches.Wedge at 0x22903ff1000>,
  <matplotlib.patches.Wedge at 0x22903fcff70>],
 [Text(-0.046768193452333374, 0.19445497186033864, 'Unaffected'),
  Text(0.046768175246145106, -0.19445497623909208, 'Affected')],
 [Text(-0.1403045803570012, 0.5833649155810159, '57.5%'),
  Text(0.1403045257384353, -0.5833649287172762, '42.5%')])
```



```
df['Category'].value_counts()
```

```
0    533
1     82
Name: Category, dtype: int64
```

```
df['Category'] = df['Category'].map({'0=Blood Donor':0, '0s=suspect Blood Donor':1, '1=Hepatitis':1,'2=Fibrosis':1, '3=Cirrhosis':1})
```

```
df1 = pd.read_csv('Datasets/Hepatitis_C___Advanced_Liver_Disease_Disparities_Dashboard.csvdf['Sex'].map({'m':1,'f':0})')
```

```
df['Sex'] = df['Sex'].map({'m':1,'f':0})
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import tr
```

```
X.isnull().sum()
```

```
Age      0
Sex      0
ALB      0
ALP      0
ALT      0
AST      0
BIL      0
CHE      0
CHOL     0
CREA     0
GGT      0
PROT     0
dtype: int64
```

```
df['ALP'] = df['ALP'].fillna(df['ALP'].mean())
```

```
df['ALP'].mean()
```

```
68.28391959798995
```

```
df['CHOL'] = df['CHOL'].fillna(df['CHOL'].mean())
```

```
df['PROT'] = df['PROT'].fillna(df['PROT'].mean())
```

```
df.to_csv('HepatitisCdata_cleaned.csv')
```

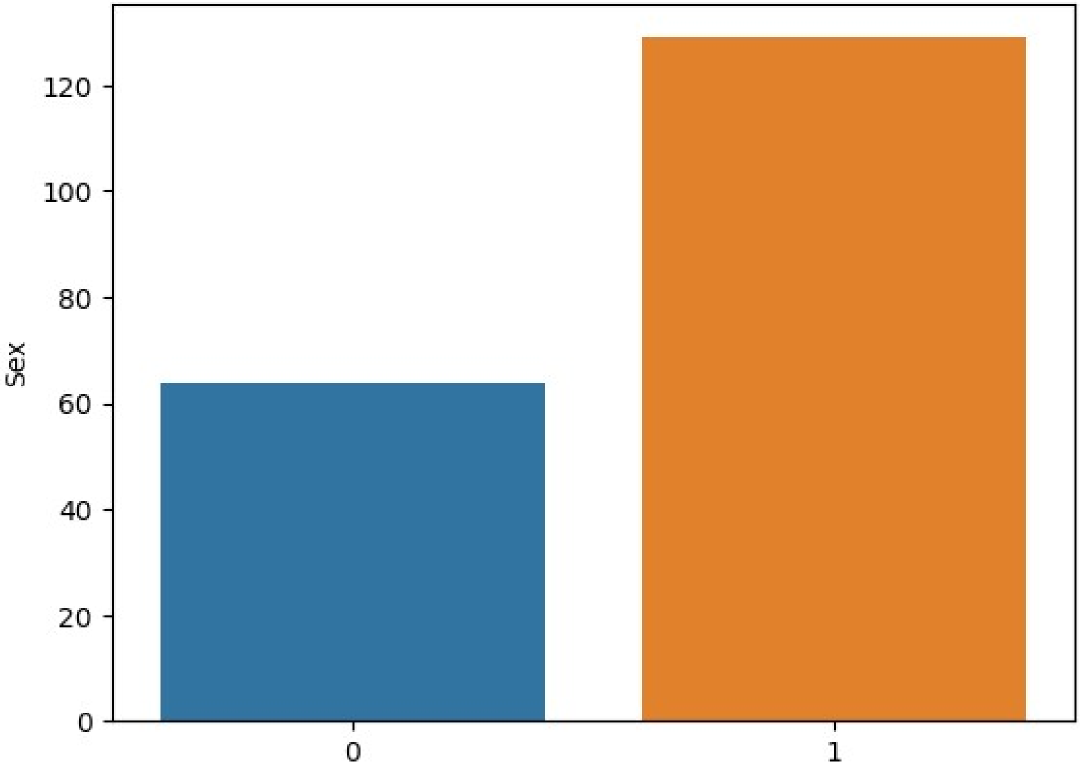
```
df['Sex'].val
```

```
0      1
1      1
2      1
3      1
4      1
```

```
      ..
610    0
611    0
612    0
613    0
614    0
Name: Sex, Length: 615, dtype: int64
```

```
# plt.bar(df['Sex'].unique(),df['Sex'].value_counts(),hue =
df['Category'])
plt.xlabel = ['Female','Male']
sns.barplot(y= df['Sex'].value_counts(),x=df['Sex'].unique())
```

```
<Axes: ylabel='Sex'>
```



```
df['Sex'].value_counts()
```

```
1    377
0    238
Name: Sex, dtype: int64
```

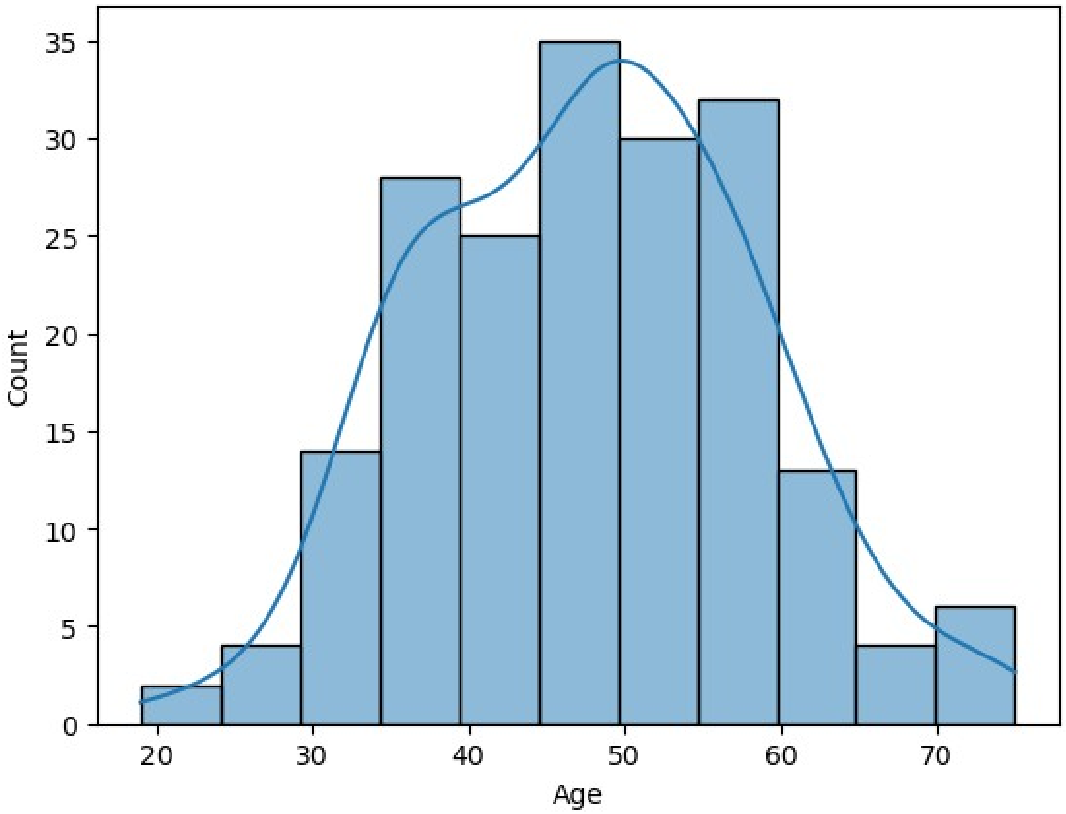
```
df
```

```
     Unnamed: 0  Category  Age  Sex   ALB        ALP    ALT    AST
BIL  \
0             1         0   32    1  38.5   52.50000    7.7   22.1
```

```
7.5
1             2         0   32     1  38.5   70.30000   18.0   24.7
3.9
2             3         0   32     1  46.9   74.70000   36.2   52.6
6.1
3             4         0   32     1  43.2   52.00000   30.6   22.6
18.9
4             5         0   32     1  39.2   74.10000   32.6   24.8
9.6
..          ...       ...  ...  ...   ...        ...    ...    ...   .
..
610         611         1   62     0  32.0  416.60000    5.9  110.3
50.0
611         612         1   64     0  24.0  102.80000    2.9   44.4
20.0
612         613         1   64     0  29.0   87.30000    3.5   99.0
48.0
613         614         1   46     0  33.0   68.28392   39.0   62.0
20.0
614         615         1   59     0  36.0   68.28392  100.0   80.0
12.0

       CHE  CHOL   CREA    GGT  PROT
0     6.93  3.23  106.0   12.1  69.0
1    11.17  4.80   74.0   15.6  76.5
2     8.84  5.20   86.0   33.2  79.3
3     7.33  4.74   80.0   33.8  75.7
4     9.15  4.32   76.0   29.9  68.7
..     ...   ...    ...    ...   ...
610   5.57  6.30   55.7  650.9  68.5
611   1.54  3.02   63.0   35.9  71.3
612   1.66  3.63   66.7   64.2  82.0
613   3.56  4.20   52.0   50.0  71.0
614   9.07  5.30   67.0   34.0  68.0

[615 rows x 14 columns]
```

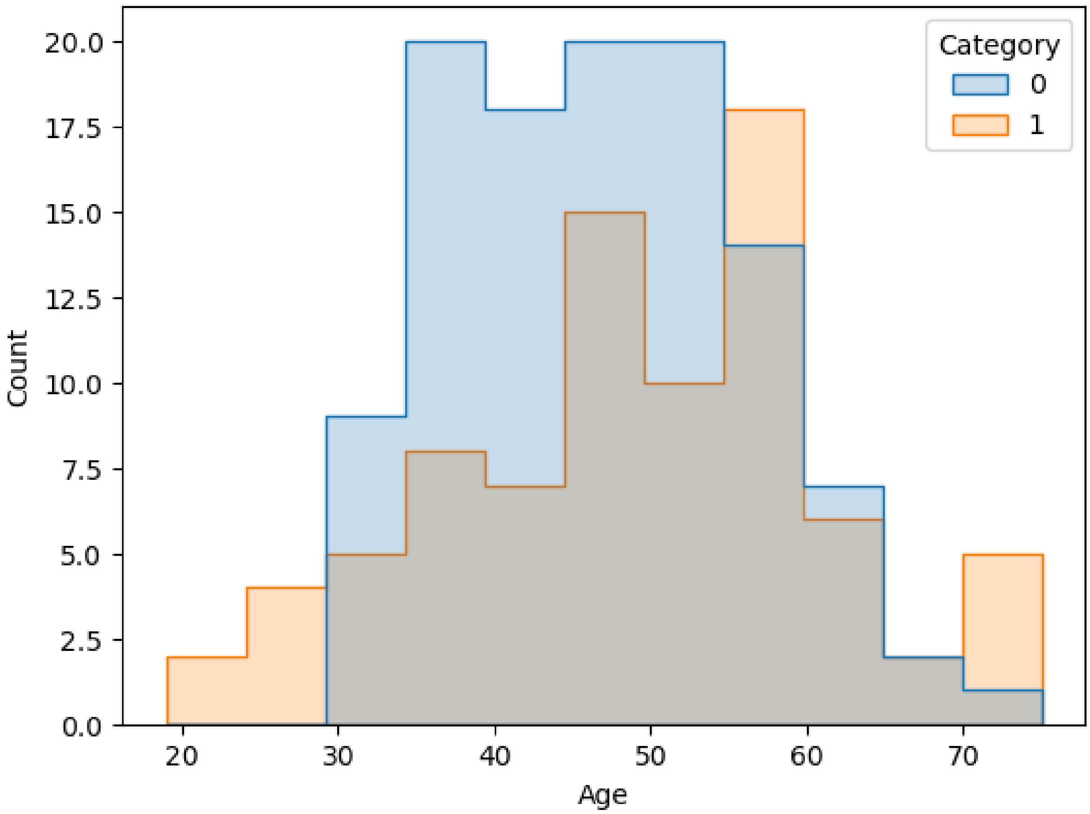
```python
sns.histplot(df['Age'],kde=True)
```

```
<Axes: xlabel='Age', ylabel='Count'>
```



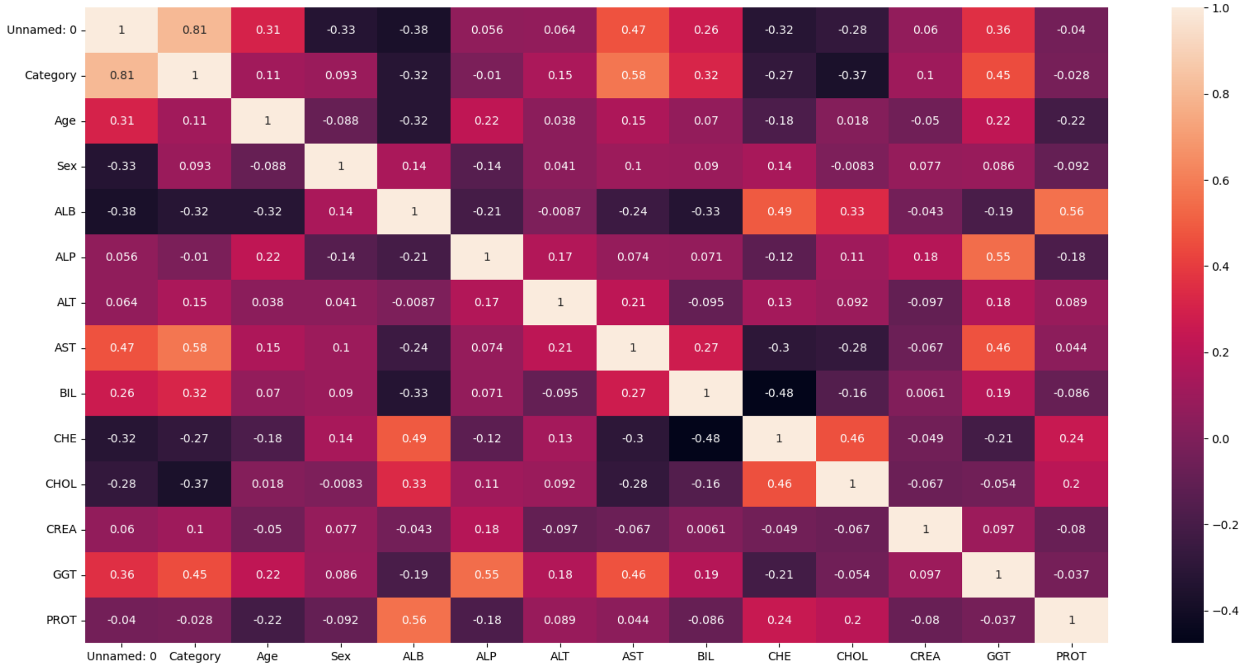
```
sns.histplot(data=df, x='Age', hue='Category', element='step',
stat='count', common_norm=False)
```

```
<Axes: xlabel='Age', ylabel='Count'>
```



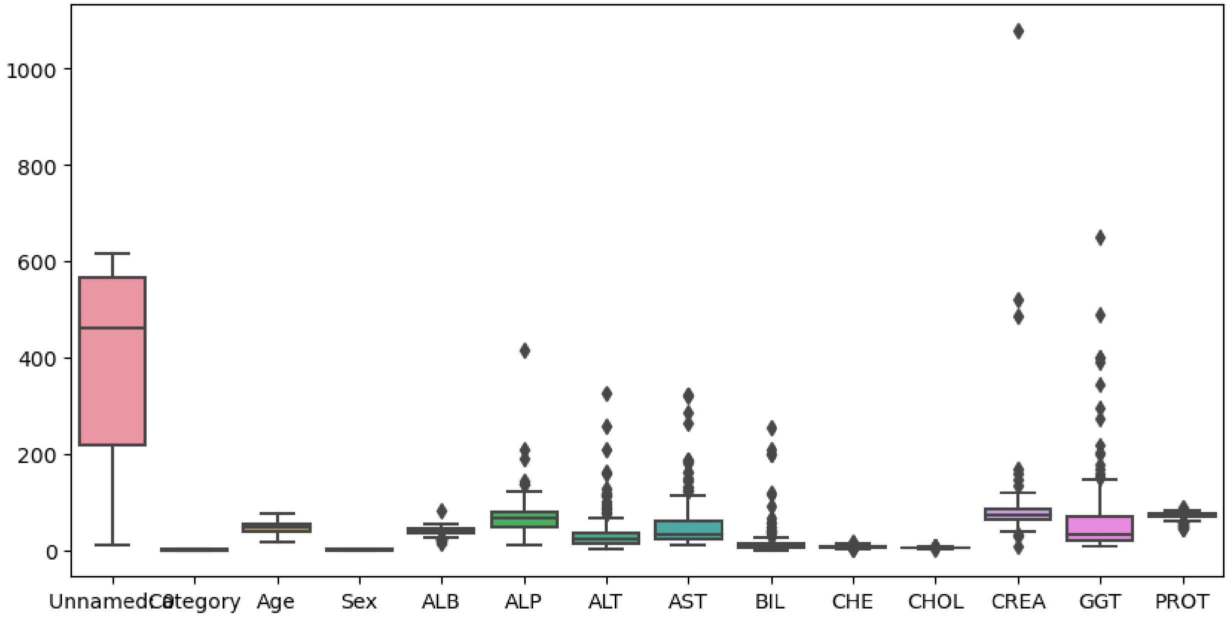
```
plt.figure(figsize=(20,10))
sns.heatmap(df.corr(),annot=True)
```

```
<Axes: >
```



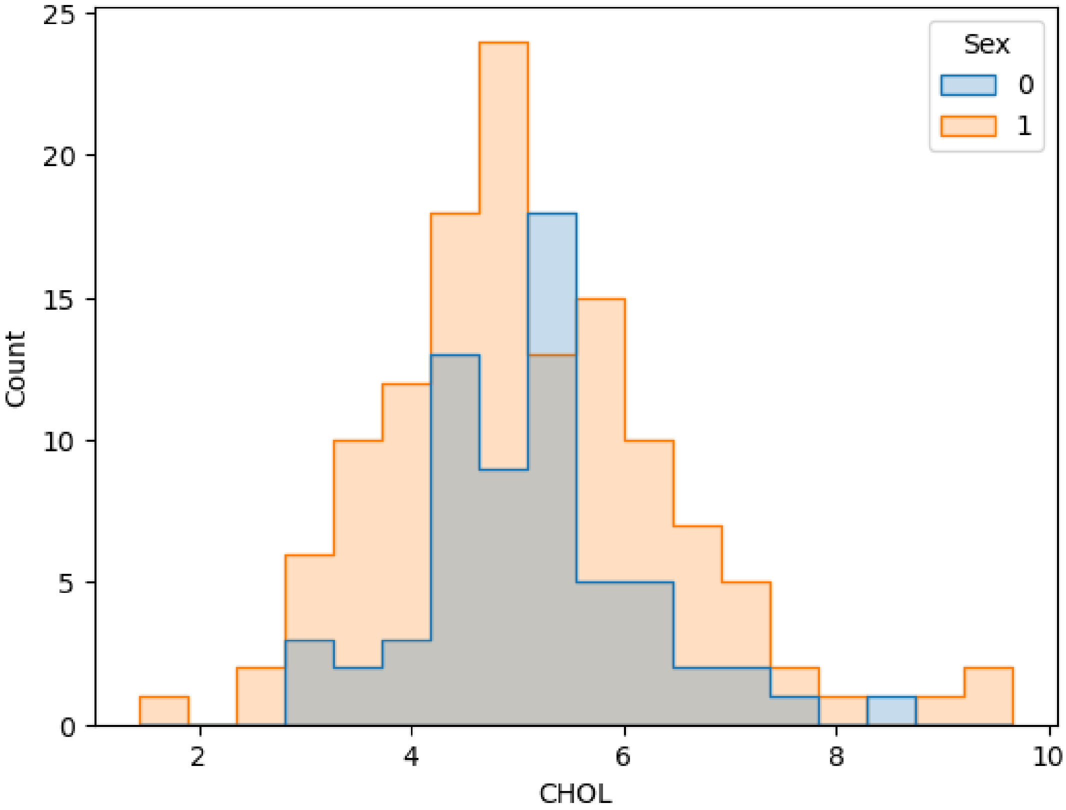
```
plt.figure(figsize=(10,5))
sns.boxplot(df)
```

```
<Axes: >
```



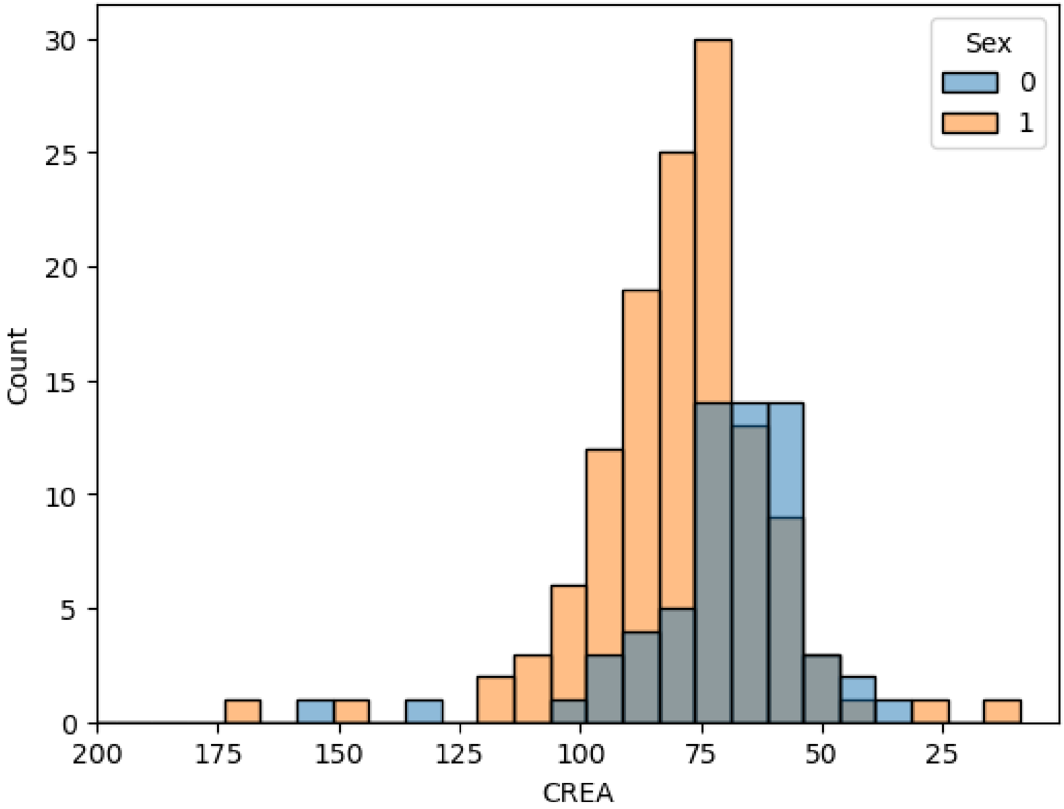
```
sns.histplot(data = df,x =
'CHOL',hue='Sex',element='step',stat='count',common_norm=False)
```

```
<Axes: xlabel='CHOL', ylabel='Count'>
```



```
plt.xlim(200)
sns.histplot(data = df,x = 'CREA',hue = 'Sex')
```

```
<Axes: xlabel='CREA', ylabel='Count'>
```

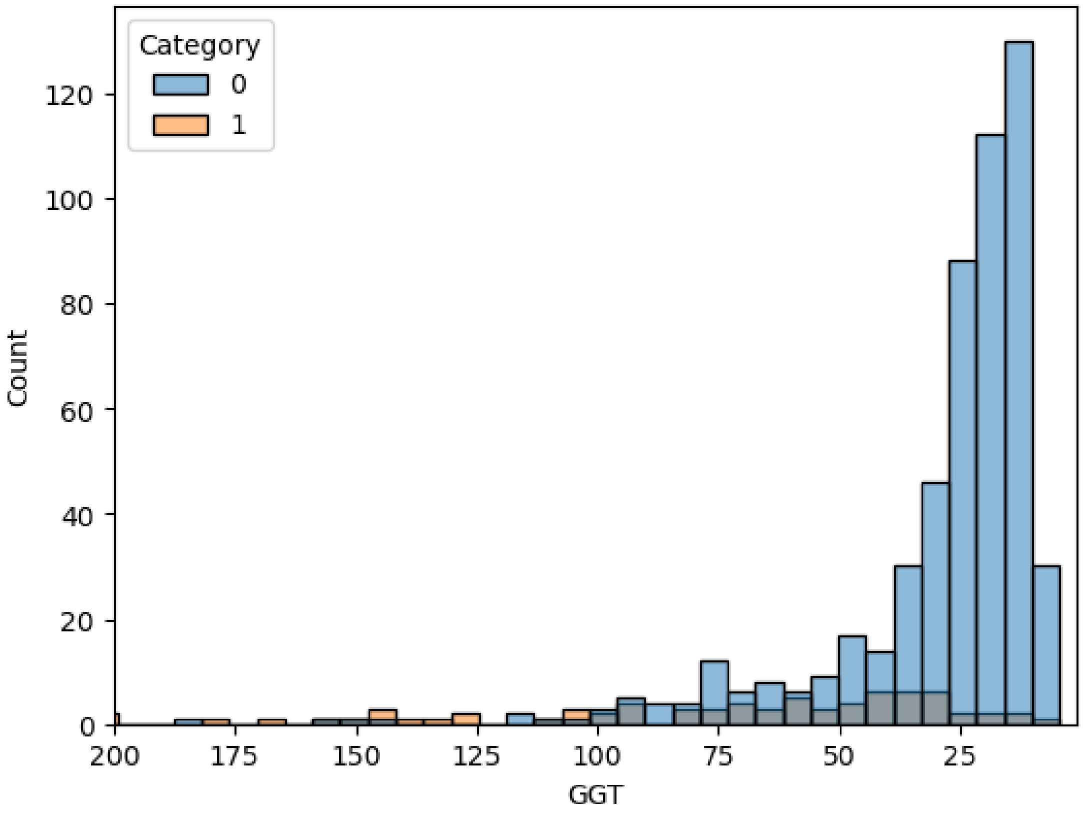


```
df

     Unnamed: 0  Category  Age  Sex   ALB        ALP    ALT    AST
BIL  \
0             1         0   32    1  38.5   52.50000    7.7   22.1
7.5
1             2         0   32    1  38.5   70.30000   18.0   24.7
3.9
2             3         0   32    1  46.9   74.70000   36.2   52.6
6.1
3             4         0   32    1  43.2   52.00000   30.6   22.6
18.9
4             5         0   32    1  39.2   74.10000   32.6   24.8
9.6
..          ...       ...  ...  ...   ...        ...    ...    ...   .
..
610         611         1   62    0  32.0  416.60000    5.9  110.3
50.0
611         612         1   64    0  24.0  102.80000    2.9   44.4
20.0
612         613         1   64    0  29.0   87.30000    3.5   99.0
48.0
613         614         1   46    0  33.0   68.28392   39.0   62.0
20.0
```

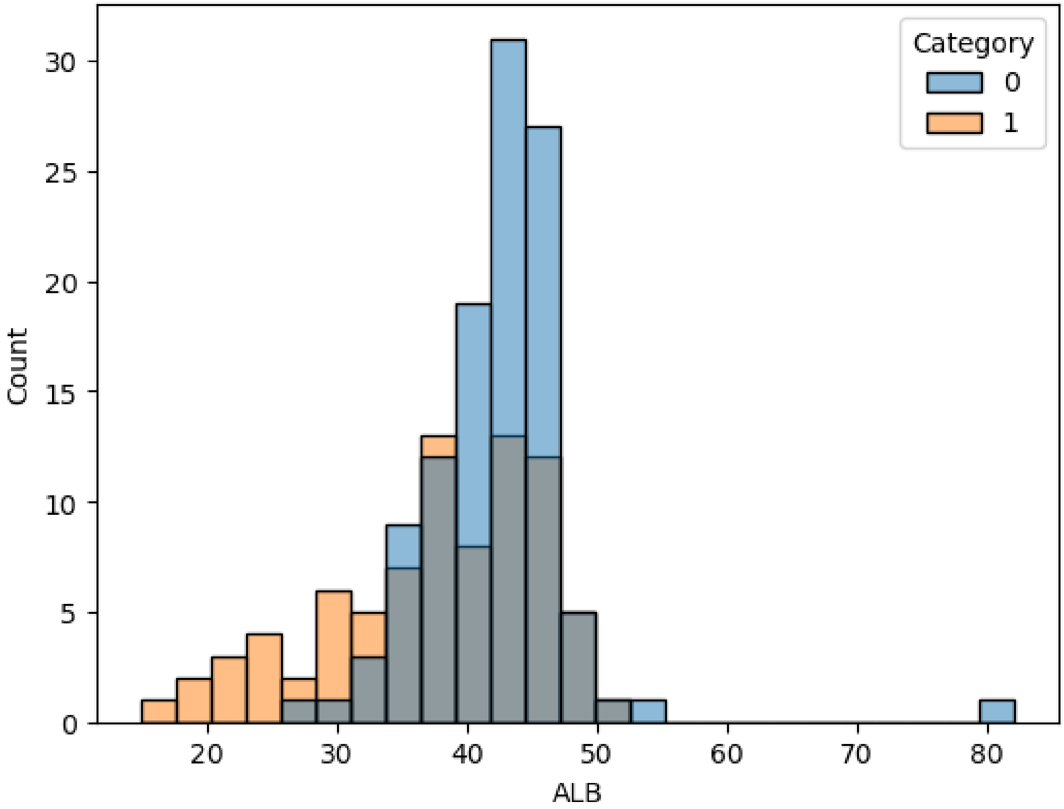
```
614          615          1   59    0  36.0   68.28392  100.0   80.0
12.0

      CHE  CHOL   CREA    GGT  PROT
0    6.93  3.23  106.0   12.1  69.0
1   11.17  4.80   74.0   15.6  76.5
2    8.84  5.20   86.0   33.2  79.3
3    7.33  4.74   80.0   33.8  75.7
4    9.15  4.32   76.0   29.9  68.7
..    ...   ...    ...    ...   ...
610  5.57  6.30   55.7  650.9  68.5
611  1.54  3.02   63.0   35.9  71.3
612  1.66  3.63   66.7   64.2  82.0
613  3.56  4.20   52.0   50.0  71.0
614  9.07  5.30   67.0   34.0  68.0

[615 rows x 14 columns]
```

```
# Gamma-glutamyl Transferase GGT is an enzyme found throughout the
body,
# but it is mostly found in the liver. When the liver is damaged,
# High GGT = may be a sign of liver disease or damage to the bile
ducts

plt.xlim(200)
sns.histplot(data = df, x = 'GGT',hue ='Category')
# females have more GGT Harmone
```

```
<Axes: xlabel='GGT', ylabel='Count'>
```



```
# ALB (Albumin):

# Albumin is a protein made by the liver. It helps maintain blood
volume and pressure and
# prevents fluid from leaking out of blood vessels.

sns.histplot(data = df, x = 'ALB',hue = 'Category')
# Lower Albumin lavels may lead to Hepatitis

<Axes: xlabel='ALB', ylabel='Count'>
```



```
# rows_to_remove = df[df['Category'] == 0].sample(frac=0.2)  # Adjust
frac as needed (e.g., 0.2 means remove 20% of the rows)
# df = df.drop(rows_to_remove.index)

df['Category'].value_counts()
```

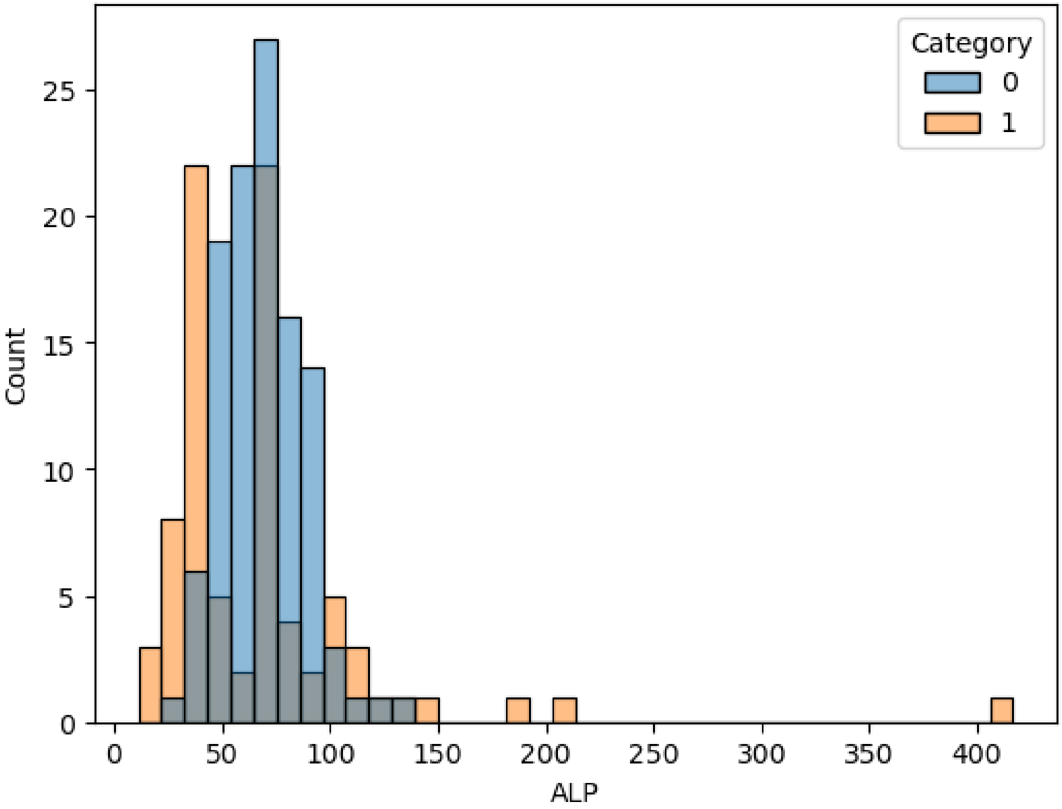
```
0    111
1     82
Name: Category, dtype: int64
```

ALP (Alkaline Phosphatase):

Alkaline Phosphatase is an enzyme found in the liver, bones, and other tissues.

```
# plt.xlim(200)
sns.histplot(data = df,x = 'ALP',hue = 'Category')
# Elevated levels may indicate liver or bone disease.
```

```
<Axes: xlabel='ALP', ylabel='Count'>
```

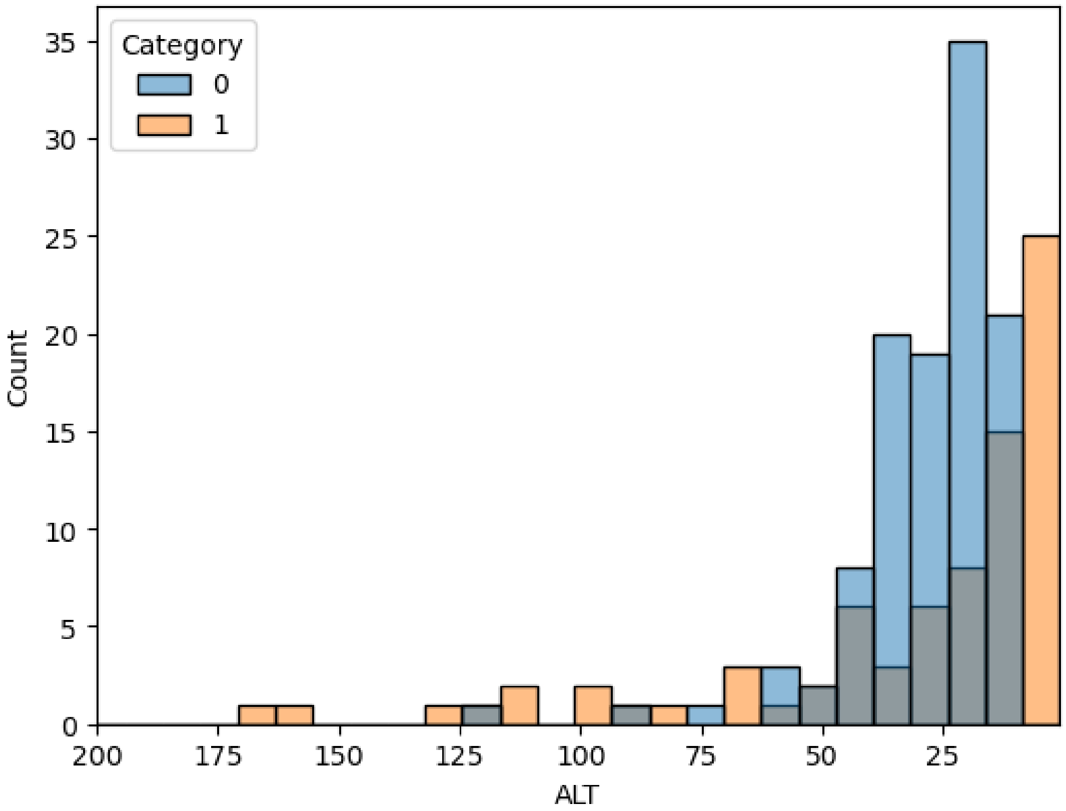


ALT (Alanine Aminotransferase):

Alanine Aminotransferase is an enzyme found mainly in the liver.

```
plt.xlim(200)
sns.histplot(data = df , x = 'ALT',hue = 'Category')
# Elevated levels may indicate liver damage or disease.
```

```
<Axes: xlabel='ALT', ylabel='Count'>
```


Category
0
1
Count
35
30
25
20
15
10
5
0
200
175
150
125
100
75
50
25
ALT