outlier-detection-and-removal

July 10, 2023

1 Problem Statement

Outlier detection and removal using:

1. Z-score
2. Percentile
3. IQR

2 Importing libraries

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

import matplotlib.pyplot as plt
import seaborn as sns
```

3 Dataset Description

[2]:
```
df1 = pd.read_csv('/kaggle/input/climate-insights-dataset/climate_change_data.
↪csv')
```

[3]:
```
df1.head()
```

[3]:

	Date	Location	Country	Temperature	CO2 Emissions	Sea Level Rise	Precipitation	Humidity
0	2000-01-01 00:00:00.000000000	New Williamtown	Latvia	10.688986	403.118903	0.717506	13.835237	23.631256
1	2000-01-01 20:09:43.258325832	North Rachel	South Africa	13.814430	396.663499	1.205715	40.974084	43.982946
2	2000-01-02 16:19:26.516651665	West Williamland	French Guiana	27.323718	451.553155	-0.160783	42.697931	96.652600
3	2000-01-03 12:29:09.774977497	South David	Vietnam	12.309581	422.404983	-0.475931	5.193341	47.467938
4	2000-01-04 08:38:53.033303330	New Scottburgh	Moldova	13.210885	410.472999	1.135757	78.695280	61.789672

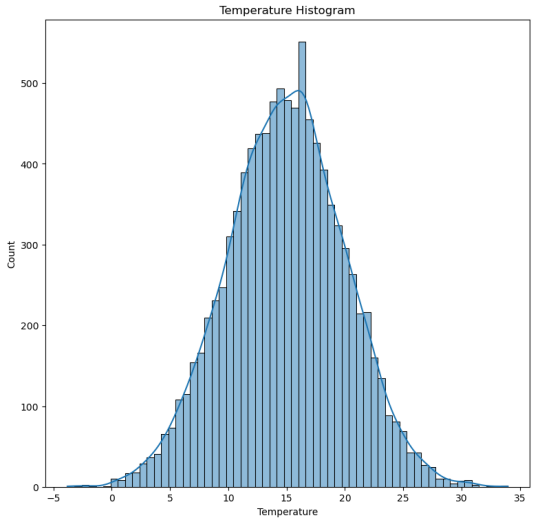
```
   Wind Speed
0   18.492026
1   34.249300
2   34.124261
3    8.554563
4    8.001164
```

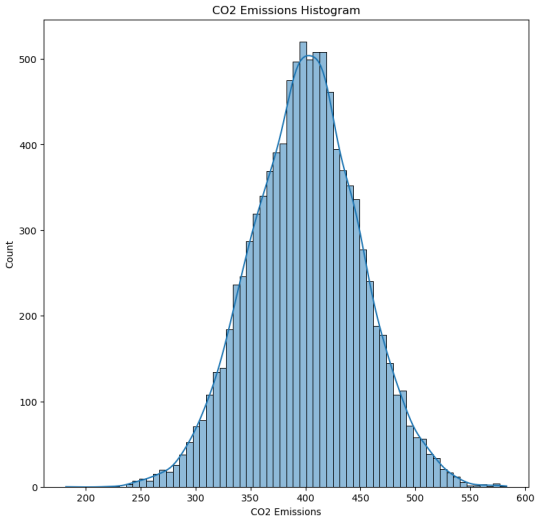
Numerical columns

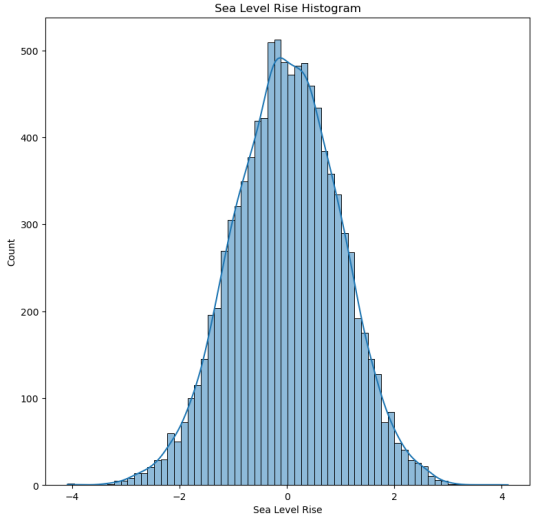
[4]:
```
num_cols = []
for col in df1.columns:
    if df1[col].dtypes != 'O':
        num_cols.append(col)
```

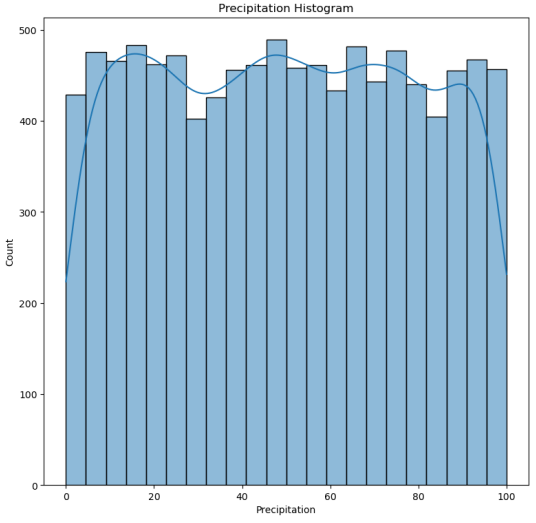
Distribution of numerical columns

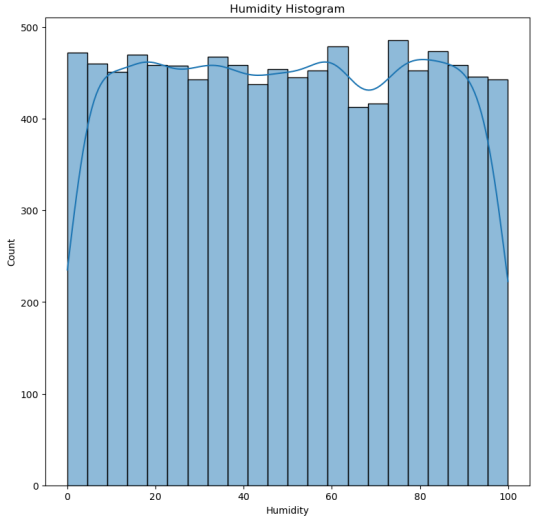
[5]:
```
plt.figure(figsize = (20, 30))
for i, col in enumerate(num_cols):
    ax = plt.subplot(3, 2, i + 1)
    sns.histplot(df1[col], kde = True)
    ax.set_title(col + " Histogram")
plt.show()
```

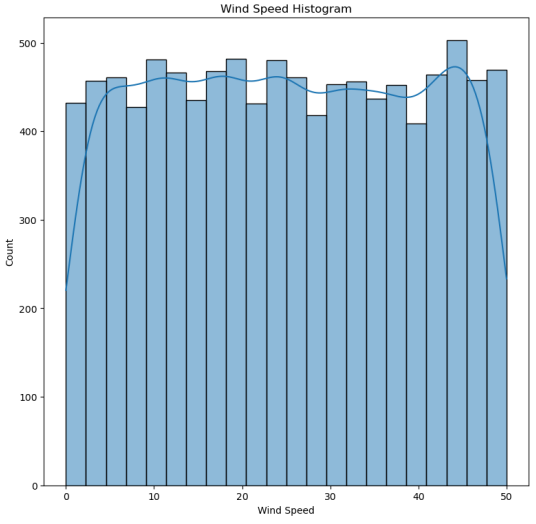

Temperature Histogram
Count
Temperature


CO2 Emissions Histogram
Count
CO2 Emissions


Sea Level Rise Histogram
Count
Sea Level Rise


Precipitation Histogram
Count
Precipitation


Humidity Histogram
Count
Humidity


Wind Speed Histogram
Count
Wind Speed

4 Z-score method

Z-score method can only be applied on columns with normal or almost normal distribution.
Here, If a certain value falls outside of 3 standard deviations we can say it an outlier.
Temperature, CO2 Emissions, Sea Level Rise are almost normal distribution. So, we will choose z-score method for it.

4.0.1 Trimming

Simply removing the outliers

[6]:
```
mean = df1['Temperature'].mean()
std = df1['Temperature'].std()
upper_limit = mean + 3 * std
lower_limit = mean - 3 * std
```

[7]:
```
df1.shape
```

[7]:
```
(10000, 9)
```

[8]:
```
# outliers
len(df1[(df1['Temperature'] < lower_limit) | (df1['Temperature'] >␣
 ↪upper_limit)])
```

[8]:
```
28
```

[9]:
```
new_df1_1 = df1[(df1['Temperature'] >= lower_limit) & (df1['Temperature'] <=␣
 ↪upper_limit)]
```

[10]:
```
new_df1_1.shape
```

[10]:
```
(9972, 9)
```

4.0.2 Capping

Setting the outliers value to upper and lower limit

[11]:
```
new_df1_2 = df1.copy()
```

[12]:
```
new_df1_2['Temperature'] = np.where(df1['Temperature'] > upper_limit,
        upper_limit,
        np.where(df1['Temperature'] < lower_limit,
                 lower_limit,
                 df1['Temperature']
                )
       )
```

[13]:
```
new_df1_1.head()
```

[13]:

	Date	Location	Country	Temperature	CO2 Emissions	Sea Level Rise	Precipitation	Humidity	Wind Speed
0	2000-01-01 00:00:00.000000000	New Williamtown	Latvia	10.688986	403.118903	0.717506	13.835237	23.631256	18.492026
1	2000-01-01 20:09:43.258325832	North Rachel	South Africa	13.814430	396.663499	1.205715	40.974084	43.982946	34.249300
2	2000-01-02 16:19:26.516651665	West Williamland	French Guiana	27.323718	451.553155	-0.160783	42.697931	96.652600	34.124261
3	2000-01-03 12:29:09.774977497	South David	Vietnam	12.309581	422.404983	-0.475931	5.193341	47.467938	8.554563
4	2000-01-04 08:38:53.033303330	New Scottburgh	Moldova	13.210885	410.472999	1.135757	78.695280	61.789672	8.001164

5 Percentile Method

Percentile - describes how a compare to other scores from the same set.
If a value is in kth percentile, it is greater than k percent of the total values.

In percentile method, if a value is greater than 99/95 percentile(depends upon the problem statement) or less than 1/5 percentile than it is consider an outlier.

[14]:
```
df2 = df1.copy()
```

[15]:
```
df2.head()
```

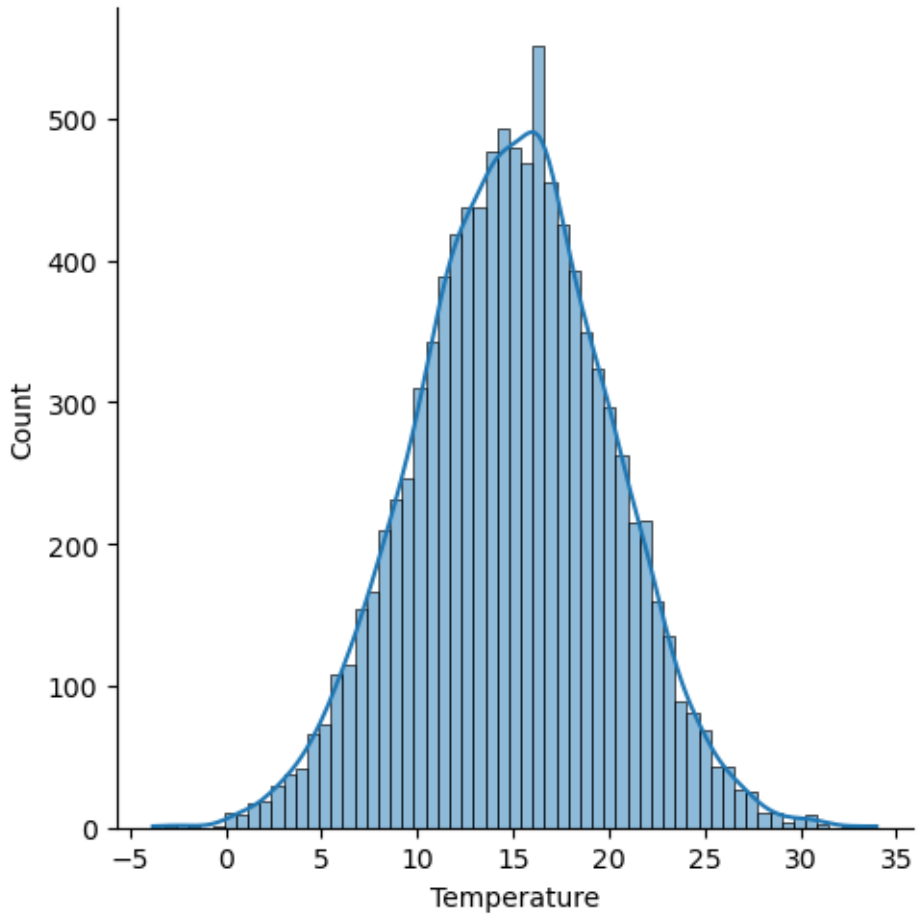
[15]:

	Date	Location	Country	Temperature	CO2 Emissions	Sea Level Rise	Precipitation	Humidity
0	2000-01-01 00:00:00.000000000	New Williamtown	Latvia	10.688986	403.118903	0.717506	13.835237	23.631256
1	2000-01-01 20:09:43.258325832	North Rachel	South Africa	13.814430	396.663499	1.205715	40.974084	43.982946
2	2000-01-02 16:19:26.516651665	West Williamland	French Guiana	27.323718	451.553155	-0.160783	42.697931	96.652600
3	2000-01-03 12:29:09.774977497	South David	Vietnam	12.309581	422.404983	-0.475931	5.193341	47.467938
4	2000-01-04 08:38:53.033303330	New Scottburgh	Moldova					

```
4    13.210885     410.472999         1.135757      78.695280   61.789672

   Wind Speed
0   18.492026
1   34.249300
2   34.124261
3    8.554563
4    8.001164
```

[16]:
```
sns.displot(df2['Temperature'], kde = True)
```

[16]:
```
<seaborn.axisgrid.FacetGrid at 0x7c7457df1180>
```



[17]:
```
#The value with 99th percentile
upper_limit = df2['Temperature'].quantile(0.99)
upper_limit
```

[17]:
```
26.54418440413302
```

[18]:
```
#The value with 1th percentile
lower_limit = df2['Temperature'].quantile(0.01)
lower_limit
```

[18]:
```
3.158667894296705
```

5.0.1 Trimming

[19]:
```
df2.shape
```

[19]:
```
(10000, 9)
```

[20]:
```
len(df2[(df2['Temperature'] > upper_limit) | (df2['Temperature'] <␣
↪lower_limit)])
```

[20]:
```
200
```

[21]:
```
new_df2_1 = df2[(df2['Temperature'] <= upper_limit) & (df2['Temperature'] >=␣
↪lower_limit)]
```

[22]:
```
new_df2_1.shape
```

[22]:
```
(9800, 9)
```

5.0.2 Capping

Capping using percentile method is called winsorization technique.

[23]:
```
new_df2_2 = df2.copy()
```

[24]:
```
new_df2_2['Temperature'] = np.where(df2['Temperature'] > upper_limit,
         upper_limit,
         np.where(df2['Temperature'] < lower_limit,
                  lower_limit,
                  df2['Temperature']
                 )
        )
```

[25]:
```
new_df2_2.head()
```

[25]:

	Date	Location	Country	\
0	2000-01-01 00:00:00.000000000	New Williamtown	Latvia	
1	2000-01-01 20:09:43.258325832	North Rachel	South Africa	
2	2000-01-02 16:19:26.516651665	West Williamland	French Guiana	
3	2000-01-03 12:29:09.774977497	South David	Vietnam	
4	2000-01-04 08:38:53.033303330	New Scottburgh	Moldova	

```
   Temperature  CO2 Emissions  Sea Level Rise  Precipitation   Humidity  \
0    10.688986     403.118903        0.717506      13.835237  23.631256
1    13.814430     396.663499        1.205715      40.974084  43.982946
2    26.544184     451.553155       -0.160783      42.697931  96.652600
3    12.309581     422.404983       -0.475931       5.193341  47.467938
4    13.210885     410.472999        1.135757      78.695280  61.789672

   Wind Speed
0   18.492026
1   34.249300
2   34.124261
3    8.554563
4    8.001164
```

6 IQR Method

IQR - Inter Quartile Range
Q1 - 25th percentile
Q2 - 50th percentile (Median)
Q3 - 75th percentile

Used for skewed data.
In this method, we calculate the minimum and maximum value. If any value is less than minimum value or greater than maximum value, then it is considered as an outlier.
IQR = Q3 - Q1
Minimum = Q1 - 1.5 * IQR
Maximum = Q3 + 1.5 * IQR

[26]:
```
df3 = pd.read_csv('/kaggle/input/titanic/train.csv')
```

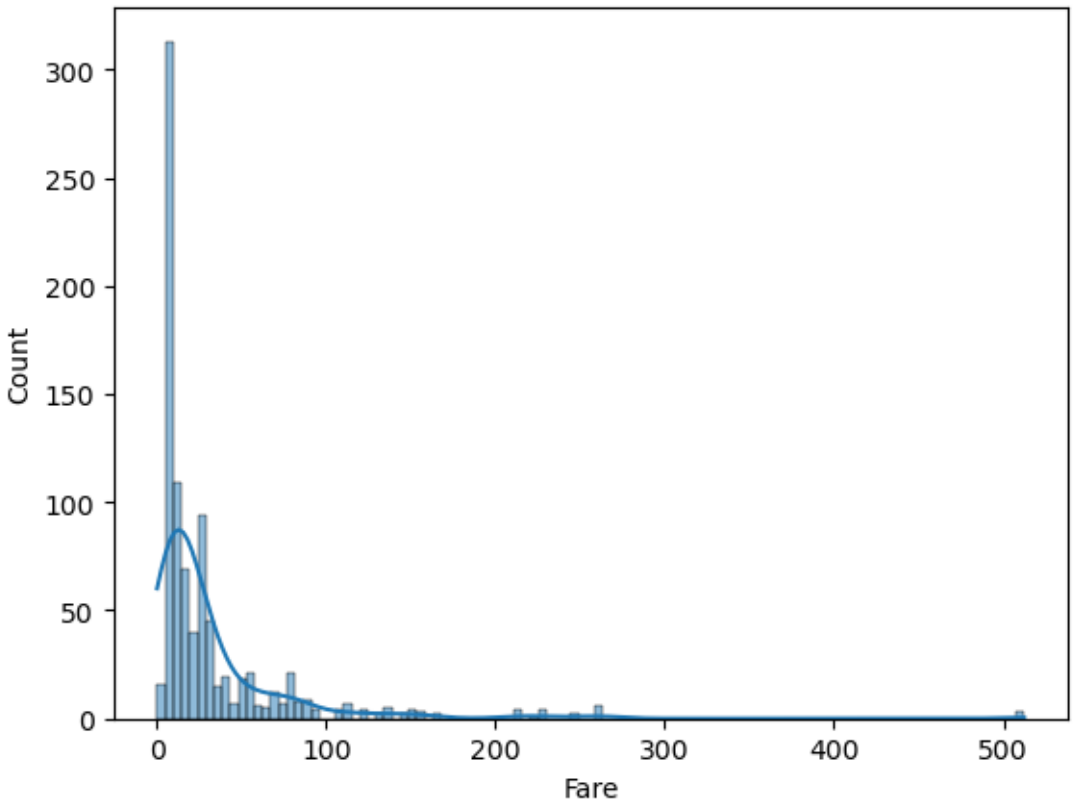
[27]:
```
df3.head()
```

[27]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin Embarked
```

```
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

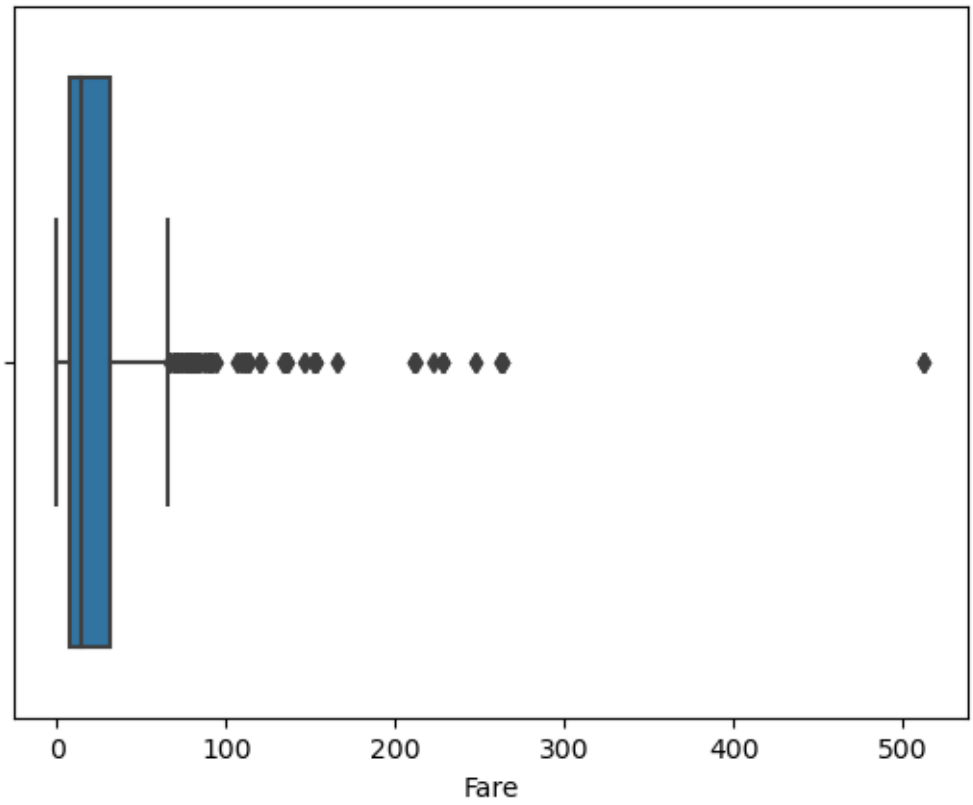
[28]:
```
sns.histplot(df3['Fare'], kde = True)
#we have positively skewed data
```

[28]: <Axes: xlabel='Fare', ylabel='Count'>



[29]:
```
sns.boxplot(x = df3['Fare'])
```

[29]: <Axes: xlabel='Fare'>



[30]:
```
q1 = df3['Fare'].quantile(0.25)
q3 = df3['Fare'].quantile(0.75)
iqr = q3 - q1
```

[31]:
```
min_val = q1 - (1.5 * iqr)
max_val = q3 + (1.5 * iqr)
```

[32]:
```
len(df3[(df3['Fare'] > max_val) | (df3['Fare'] < min_val)])
```

[32]: 116

6.0.1 Trimming

[33]:
```
df3.shape
```

[33]: (891, 12)

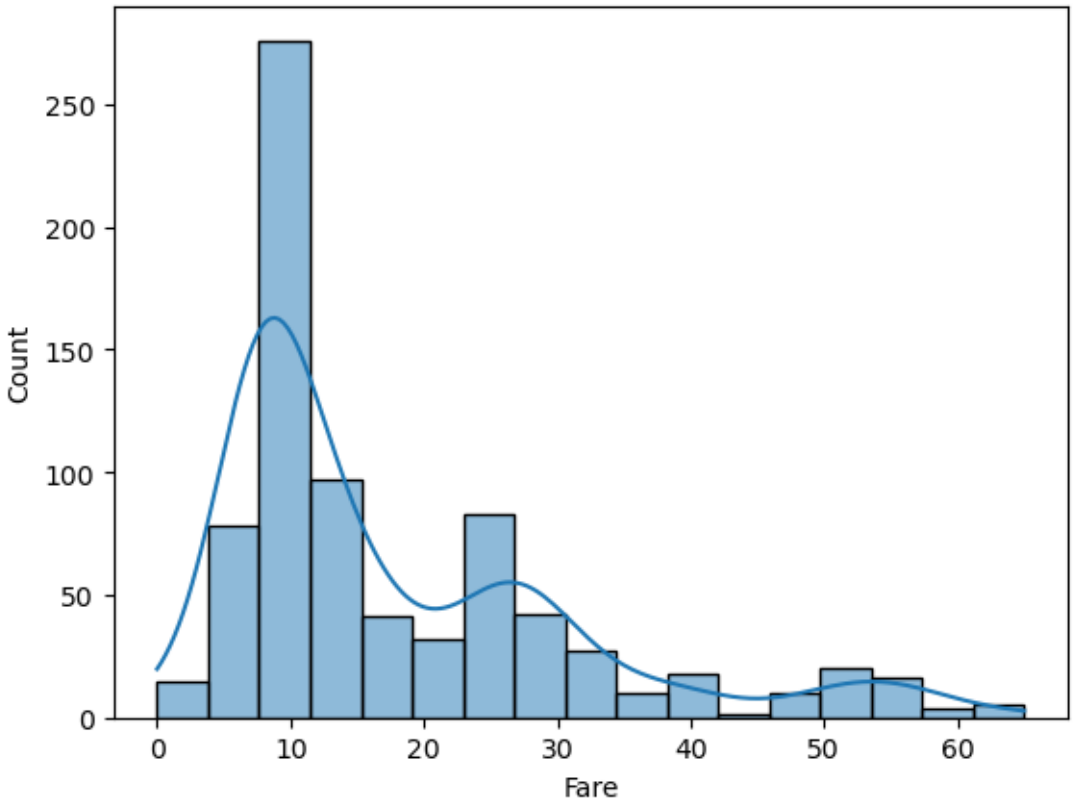
[34]:
```
new_df3_1 = df3[(df3['Fare'] < max_val) & (df3['Fare'] > min_val)]
```

[35]:
```
new_df3_1.shape
```

[35]: (775, 12)

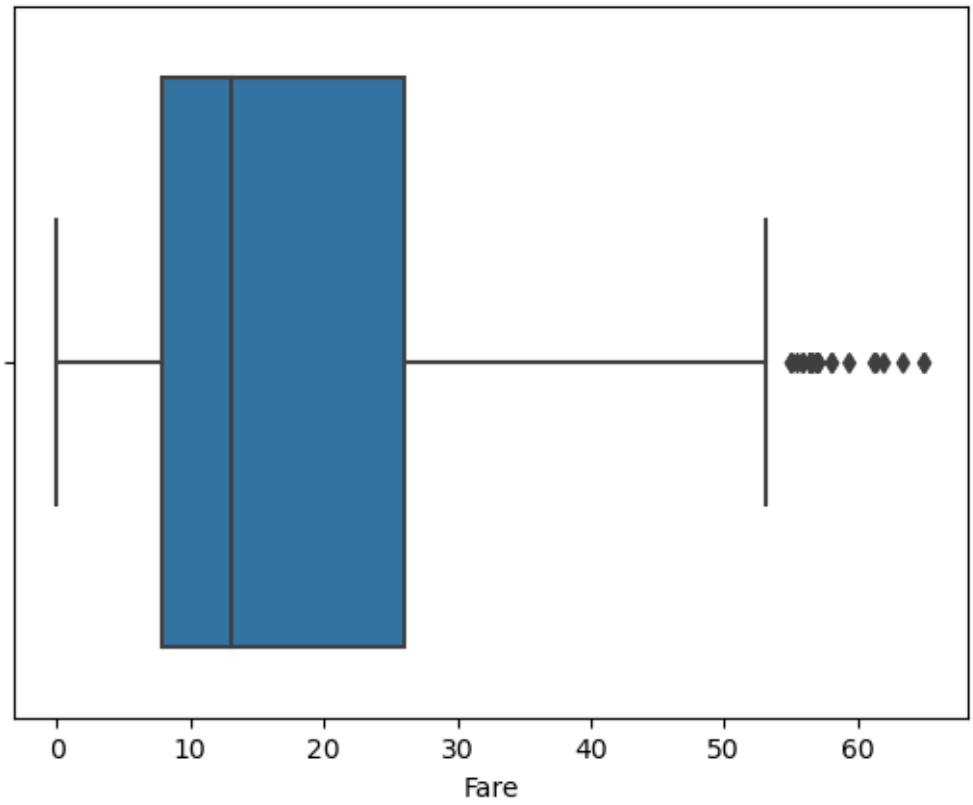
[36]:
```
sns.histplot(new_df3_1['Fare'], kde = True)
```

[36]: <Axes: xlabel='Fare', ylabel='Count'>



[37]:
```
sns.boxplot(x = new_df3_1['Fare'])
#selfnote - why still outliers are present in the box plot
```

[37]: <Axes: xlabel='Fare'>



6.0.2 Capping

[38]:
```
new_df3_2 = df3.copy()
```

[39]:
```
new_df3_2['Fare'] = np.where(df3['Fare'] > max_val,
         max_val,
         np.where(df3['Fare'] < min_val,
                  min_val,
                  df3['Fare']
                 )
        )
```

[40]:
```
new_df3_2.sample(5)
```

[40]:

	PassengerId	Survived	Pclass	Name	Sex	Age	\
346	347	1	2	Smith, Miss. Marion Elsie	female	40.00	
544	545	0	1	Douglas, Mr. Walter Donald	male	50.00	
755	756	1	2	Hamalainen, Master. Viljo	male	0.67	
393	394	1	1	Newell, Miss. Marjorie	female	23.00	
364	365	0	3	O'Brien, Mr. Thomas	male	NaN	

	SibSp	Parch	Ticket	Fare	Cabin	Embarked
346	0	0	31418	13.0000	NaN	S
544	1	0	PC 17761	65.6344	C86	C
755	1	1	250649	14.5000	NaN	S
393	1	0	35273	65.6344	D36	C
364	1	0	370365	15.5000	NaN	Q