In [24]:
```
#imports
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
```

In [25]:
```
#Create dataset for student performance
# Create a dictionary with the data
data = {
    'name': pd.Series(['Alice', 'Bob', 'Charlie', 'David', 'Emma', 'Frank', 'G
                       'Katie', 'Liam', 'Mia', 'Nate', 'Olivia', 'Peter', 'Qui
    'division': pd.Series(['A', 'B', 'A', 'C', 'B', 'A', 'B', 'C', 'B', 'A', '
                           'B', 'A', 'C']),
    'marks1': pd.Series([70, 80, 85, 90, 95, 65, 75, 60, 50, 85, np.nan, 55, 8
    'marks2': pd.Series([60, 70, 75, 80, 85, 55, 65, 50, 40, 75, 80, 45, 70, 6
    'marks3': pd.Series([5, 60, 65, 70, 75, 45, 55, 40, 30, 65, 70, 35, 60, 50
}

# Create the dataframe
df = pd.DataFrame(data)
```

In [26]:
```
df
```

Out[26]:

	name	division	marks1	marks2	marks3
0	Alice	A	70.0	60.0	5.0
1	Bob	B	80.0	70.0	60.0
2	Charlie	A	85.0	75.0	65.0
3	David	C	90.0	80.0	70.0
4	Emma	B	95.0	85.0	75.0
5	Frank	A	65.0	55.0	45.0
6	Grace	B	75.0	65.0	55.0
7	Henry	C	60.0	50.0	40.0
8	Ivy	B	50.0	40.0	30.0
9	Jack	A	85.0	75.0	65.0
10	Katie	C	NaN	80.0	70.0
11	Liam	B	55.0	45.0	35.0
12	Mia	A	80.0	70.0	60.0
13	Nate	C	70.0	60.0	50.0
14	Olivia	B	75.0	NaN	55.0
15	Peter	A	40.0	30.0	20.0
16	Quinn	C	90.0	80.0	70.0
17	Rachel	B	80.0	70.0	60.0
18	Sam	A	85.0	75.0	NaN
19	Tyler	C	65.0	55.0	45.0

In [27]:
```
df.head()
```

Out[27]:

	name	division	marks1	marks2	marks3
0	Alice	A	70.0	60.0	5.0
1	Bob	B	80.0	70.0	60.0
2	Charlie	A	85.0	75.0	65.0
3	David	C	90.0	80.0	70.0
4	Emma	B	95.0	85.0	75.0

In [28]:
```
df.head(10)
```

Out[28]:

	name	division	marks1	marks2	marks3
0	Alice	A	70.0	60.0	5.0
1	Bob	B	80.0	70.0	60.0
2	Charlie	A	85.0	75.0	65.0
3	David	C	90.0	80.0	70.0
4	Emma	B	95.0	85.0	75.0
5	Frank	A	65.0	55.0	45.0
6	Grace	B	75.0	65.0	55.0
7	Henry	C	60.0	50.0	40.0
8	Ivy	B	50.0	40.0	30.0
9	Jack	A	85.0	75.0	65.0

In [29]:
```
df.tail()
```

Out[29]:

	name	division	marks1	marks2	marks3
15	Peter	A	40.0	30.0	20.0
16	Quinn	C	90.0	80.0	70.0
17	Rachel	B	80.0	70.0	60.0
18	Sam	A	85.0	75.0	NaN
19	Tyler	C	65.0	55.0	45.0

In [30]:
```
df.tail(10)
```

Out[30]:

	name	division	marks1	marks2	marks3
10	Katie	C	NaN	80.0	70.0
11	Liam	B	55.0	45.0	35.0
12	Mia	A	80.0	70.0	60.0
13	Nate	C	70.0	60.0	50.0
14	Olivia	B	75.0	NaN	55.0
15	Peter	A	40.0	30.0	20.0
16	Quinn	C	90.0	80.0	70.0
17	Rachel	B	80.0	70.0	60.0
18	Sam	A	85.0	75.0	NaN
19	Tyler	C	65.0	55.0	45.0

In [31]:
```
# Data Preprocessing

#display information about data set
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20 entries, 0 to 19
Data columns (total 5 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   name      20 non-null     object
 1   division  20 non-null     object
 2   marks1    19 non-null     float64
 3   marks2    19 non-null     float64
 4   marks3    19 non-null     float64
dtypes: float64(3), object(2)
memory usage: 928.0+ bytes
```

In [32]:
```
df.columns
```

Out[32]:
```
Index(['name', 'division', 'marks1', 'marks2', 'marks3'], dtype='object')
```

In [33]:
```
df.shape
```

Out[33]:
```
(20, 5)
```

In [34]:
```
df.index
```

Out[34]:
```
RangeIndex(start=0, stop=20, step=1)
```

In [35]:
```
df.isnull()
```

Out[35]:

	name	division	marks1	marks2	marks3
0	False	False	False	False	False
1	False	False	False	False	False
2	False	False	False	False	False
3	False	False	False	False	False
4	False	False	False	False	False
5	False	False	False	False	False
6	False	False	False	False	False
7	False	False	False	False	False
8	False	False	False	False	False
9	False	False	False	False	False
10	False	False	True	False	False
11	False	False	False	False	False
12	False	False	False	False	False
13	False	False	False	False	False
14	False	False	False	True	False
15	False	False	False	False	False
16	False	False	False	False	False
17	False	False	False	False	False
18	False	False	False	False	True
19	False	False	False	False	False

In [36]:
```
df.isnull().any()
```

Out[36]:
```
name        False
division    False
marks1       True
marks2       True
marks3       True
dtype: bool
```

In [37]:
```
df.isnull().sum()
```

Out[37]:
```
name        0
division    0
marks1      1
marks2      1
marks3      1
dtype: int64
```

In [38]:
```
# Fill NaN values in marks1 with the mean of marks2 and marks3 for that row
df['marks1'].fillna(df[['marks2', 'marks3']].mean(axis=1), inplace=True)

# Fill NaN values in marks2 with the mean of marks1 and marks3 for that row
df['marks2'].fillna(df[['marks1', 'marks3']].mean(axis=1), inplace=True)

# Fill NaN values in marks3 with the mean of marks1 and marks2 for that row
df['marks3'].fillna(df[['marks1', 'marks2']].mean(axis=1), inplace=True)
```

In [39]:
```
df.isnull().sum()
```

Out[39]:
```
name        0
division    0
marks1      0
marks2      0
marks3      0
dtype: int64
```

In [40]:
```
df
```

Out[40]:

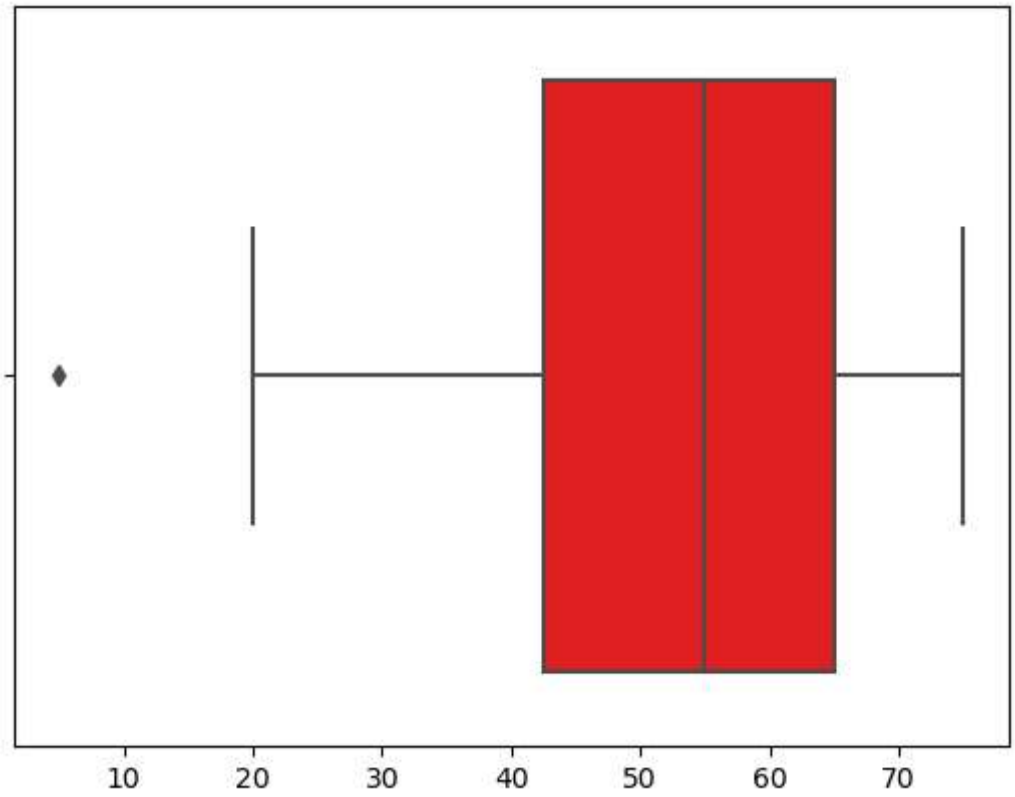
	name	division	marks1	marks2	marks3
0	Alice	A	70.0	60.0	5.0
1	Bob	B	80.0	70.0	60.0
2	Charlie	A	85.0	75.0	65.0
3	David	C	90.0	80.0	70.0
4	Emma	B	95.0	85.0	75.0
5	Frank	A	65.0	55.0	45.0
6	Grace	B	75.0	65.0	55.0
7	Henry	C	60.0	50.0	40.0
8	Ivy	B	50.0	40.0	30.0
9	Jack	A	85.0	75.0	65.0
10	Katie	C	75.0	80.0	70.0
11	Liam	B	55.0	45.0	35.0
12	Mia	A	80.0	70.0	60.0
13	Nate	C	70.0	60.0	50.0
14	Olivia	B	75.0	65.0	55.0
15	Peter	A	40.0	30.0	20.0
16	Quinn	C	90.0	80.0	70.0
17	Rachel	B	80.0	70.0	60.0
18	Sam	A	85.0	75.0	80.0
19	Tyler	C	65.0	55.0	45.0

In [56]:
```
df.describe()
```

Out[56]:

	marks1	marks2	marks3
count	2.000000e+01	20.000000	2.000000e+01
mean	1.665335e-17	0.000000	-2.775558e-18
std	1.025978e+00	1.025978	1.025978e+00
min	-2.397448e+00	-2.373741	-2.539005e+00
25%	-6.083078e-01	-0.641083	-4.785559e-01
50%	1.073484e-01	0.225245	2.525712e-01
75%	8.230047e-01	0.745043	7.178338e-01
max	1.538661e+00	1.438106	1.448961e+00

In [54]:
```
#Outlier present in 'marks3' can be visualized below
sns.boxplot(data= data, x='marks3', color= 'red');
```



In [46]:
```
def detect_outliers_iqr(data):
    Q1 = data.quantile(0.25)
    Q3 = data.quantile(0.75)

    IQR = Q3 - Q1
    # print(Q1,Q3,IQR)
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = data[(data < lower_bound) | (data > upper_bound)]
    return outliers
```

In [47]:
```
outliers = detect_outliers_iqr(data['marks3'])
outliers
```

Out[47]:
```
0    5.0
dtype: float64
```

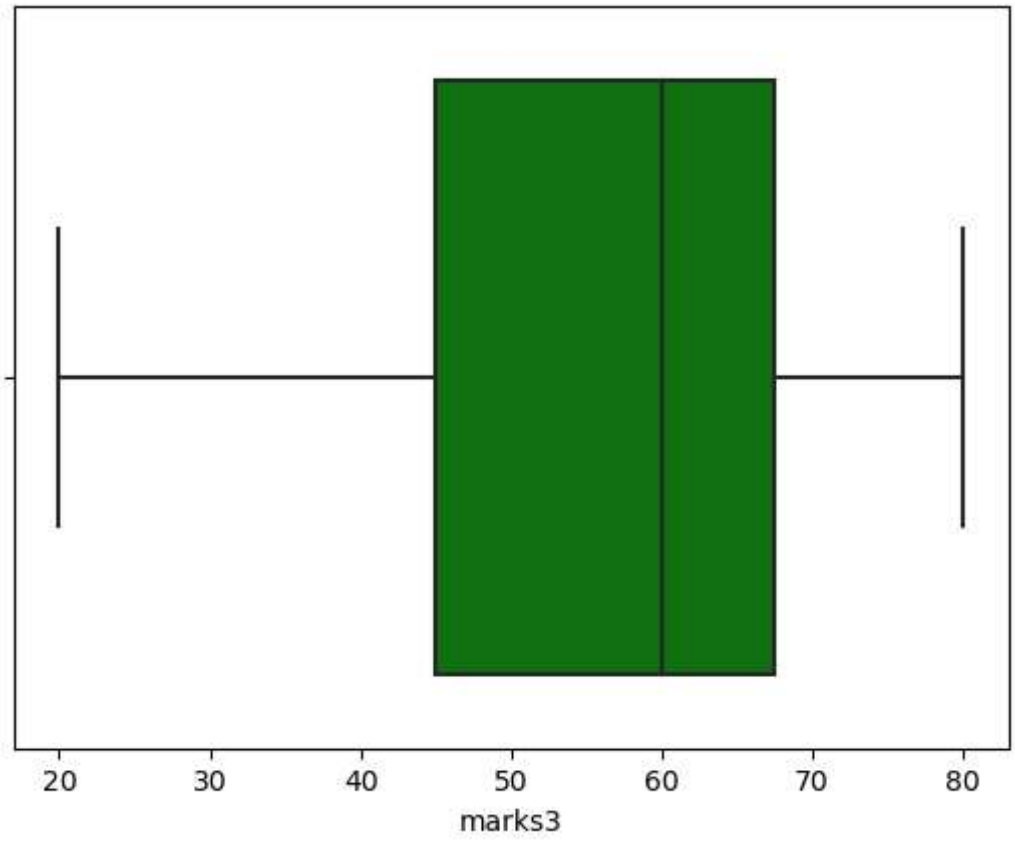
In [48]:
```
data_wo_outliers = df[~df['marks3'].isin(outliers)]
```

In [49]:
```
data_wo_outliers
```

Out[49]:

	name	division	marks1	marks2	marks3
1	Bob	B	80.0	70.0	60.0
2	Charlie	A	85.0	75.0	65.0
3	David	C	90.0	80.0	70.0
4	Emma	B	95.0	85.0	75.0
5	Frank	A	65.0	55.0	45.0
6	Grace	B	75.0	65.0	55.0
7	Henry	C	60.0	50.0	40.0
8	Ivy	B	50.0	40.0	30.0
9	Jack	A	85.0	75.0	65.0
10	Katie	C	75.0	80.0	70.0
11	Liam	B	55.0	45.0	35.0
12	Mia	A	80.0	70.0	60.0
13	Nate	C	70.0	60.0	50.0
14	Olivia	B	75.0	65.0	55.0
15	Peter	A	40.0	30.0	20.0
16	Quinn	C	90.0	80.0	70.0
17	Rachel	B	80.0	70.0	60.0
18	Sam	A	85.0	75.0	80.0
19	Tyler	C	65.0	55.0	45.0

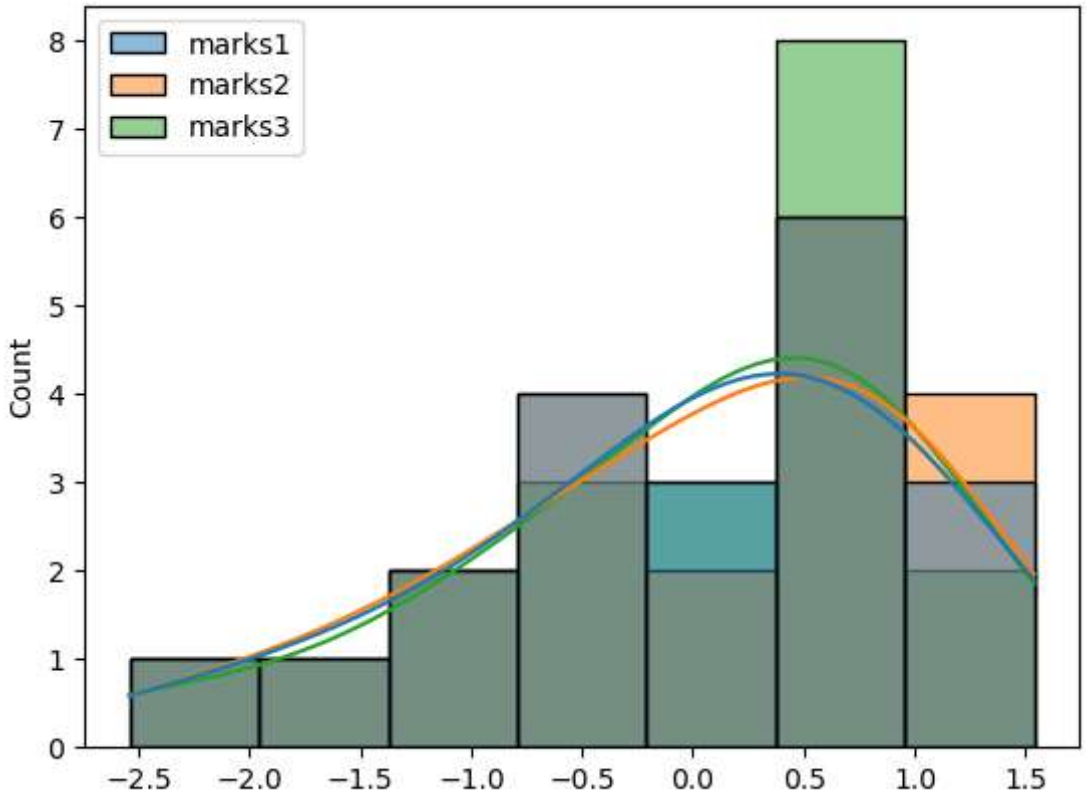
In [50]:
```
# The outlier has been removed and this can be visualized by the plot below
sns.boxplot(data= data_wo_outliers, x= 'marks3', color= 'green');
```



In [51]:
```
# Create a StandardScaler object
scaler = StandardScaler()

# Fit the scaler to the data and transform the data
df[['marks1', 'marks2', 'marks3']] = scaler.fit_transform(df[['marks1', 'marks
```

In [52]:
```
sns.histplot(df, kde=True);
```



In [ ]:
```

```

In [ ]:
```

```