In [1]:
```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
```

In [2]:
```
file_path = 'sample_kmeans_data.xlsx'  # Replace with your actual file path
df = pd.read_excel(file_path)
df.head(3)
```

Out[2]:

	Feature1	Feature2	Feature3
0	-2.936066	9.515734	5.798065
1	-10.33909	7.488545	1.166371
2	-8.461031	8.974873	0.695468

In [3]:
```
# Convert specific object columns to float safely
for col in df.columns:
    df[col] = pd.to_numeric(df[col], errors='coerce')
```

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
Feature1    19
Feature2    19
Feature3    19
dtype: int64
```

In [5]:
```
for col in df.columns:
    print(f"{col} type: {df[col].dtype}")
    print(df[col].head())
```

```
Feature1 type: float64
0    -2.936066
1   -10.339090
2    -8.461031
3    -1.927074
4     2.204123
Name: Feature1, dtype: float64
Feature2 type: float64
0    9.515734
1    7.488545
2    8.974873
3    9.902034
4   -7.688564
Name: Feature2, dtype: float64
Feature3 type: float64
0    5.798065
1    1.166371
2    0.695468
3    5.534211
4   -5.822686
Name: Feature3, dtype: float64
```

In [6]:
```
df = df.dropna()
```

In [7]:
```
scaler = StandardScaler()
scaled_data = scaler.fit_transform(df)
```

In [8]:
```
k = 3  # Set the number of clusters
kmeans = KMeans(n_clusters=k, random_state=42)
kmeans.fit(scaled_data)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1429: UserWarning: KMeans is known to have a memory leak on Windows with MKL, when there are less chunks than available threads. You can avoid it by setting the environment variable OMP_NUM_THREADS=1.
  warnings.warn(
```

Out[8]:

KMeans

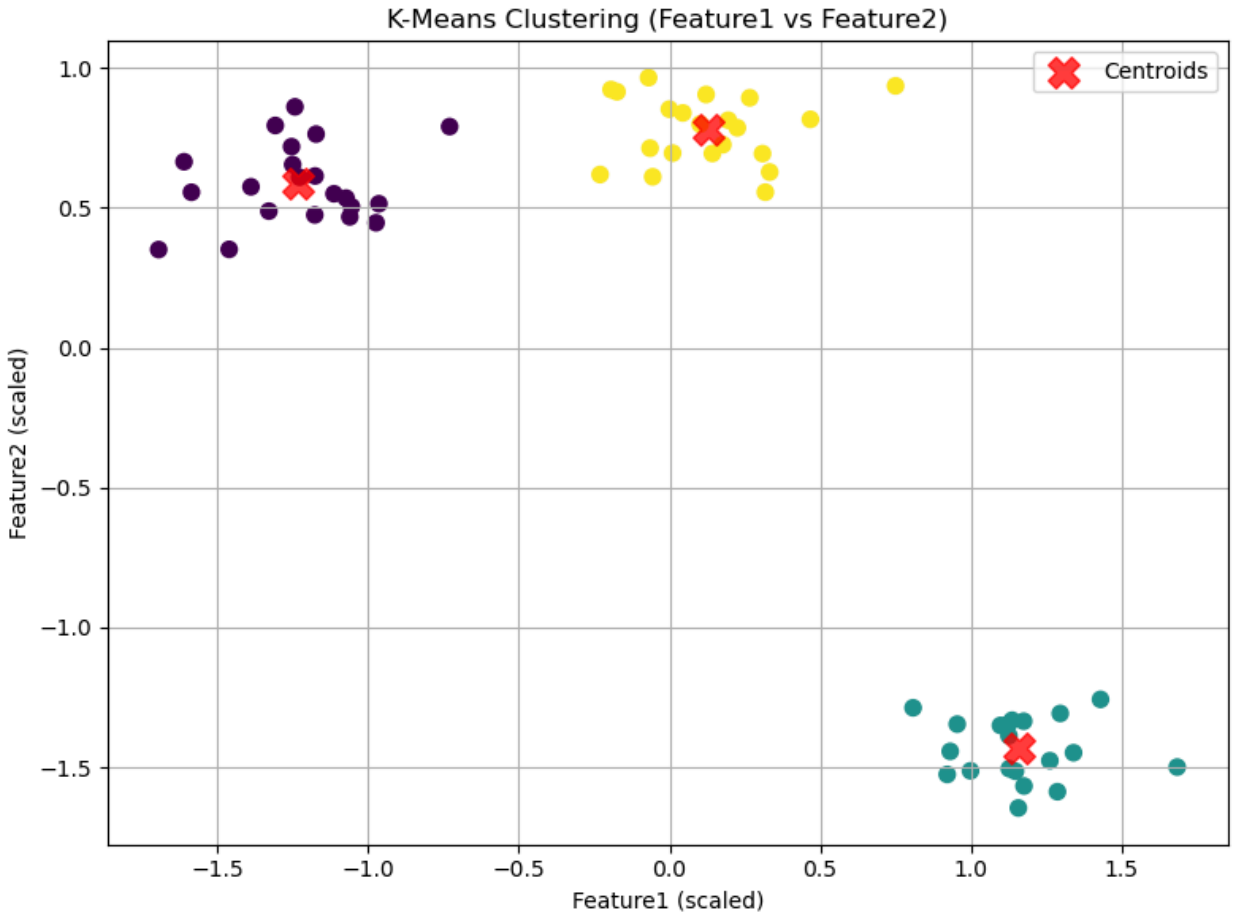
```
KMeans(n_clusters=3, random_state=42)
```

In [9]:
```
df['Cluster'] = kmeans.labels_
```

In [10]:
```
output_file = 'Kkmeans_output.xlsx'
df.to_excel(output_file, index=False)
print(f"Clustered data saved to {output_file}")
```

```
Clustered data saved to Kkmeans_output.xlsx
```

In [31]:
```
if scaled_data.shape[1] >= 2:
    plt.figure(figsize=(8, 6))
    plt.scatter(scaled_data[:, 0], scaled_data[:, 1], c=kmeans.labels_, cmap='
    plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1],
                c='red', s=200, alpha=0.75, marker='X', label='Centroids')
    plt.title('K-Means Clustering (Feature1 vs Feature2)')
    plt.xlabel('Feature1 (scaled)')
    plt.ylabel('Feature2 (scaled)')
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.show()
```


K-Means Clustering (Feature1 vs Feature2)
Centroids
Feature2 (scaled)
Feature1 (scaled)
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

In [ ]: