Clustering Data Obtained from STFs

The features we are using to perform clustering are Magnitude, Depth and Moment Release.

Additionally we are using opt files as they show the actual recording of an earthquake closed to mean and hence produces more accurate results.

Importing Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score
```

```
file_path = '/content/extracted_data-opt.csv'
df = pd.read_csv(file_path)
df.head()
```

	Filename	Depth_km	M0_Nm	Mw	lat	lon
0	fctoptsource_19920120_133703_BONIN_ISLANDS__JA...	522.0	8.882000e+18	6.566	27.98	139.40
1	fctoptsource_19920213_012913_VANUATU_ISLANDS	20.0	1.493000e+19	6.716	-15.89	166.32
2	fctoptsource_19920305_143910_OFF_EAST_COAST_OF...	53.0	3.607000e+18	6.305	52.90	159.62
3	fctoptsource_19920307_015337_COSTA_RICA	79.0	6.732000e+18	6.485	10.21	-84.32
4	fctoptsource_19920313_160104_ANDREANOF_ISLANDS	214.0	5.273000e+18	6.415	52.45	-178.95

Next steps: Generate code with df View recommended plots New interactive sheet

```
features = ['Mw', 'M0_Nm', 'Depth_km']
data = df[features]
```

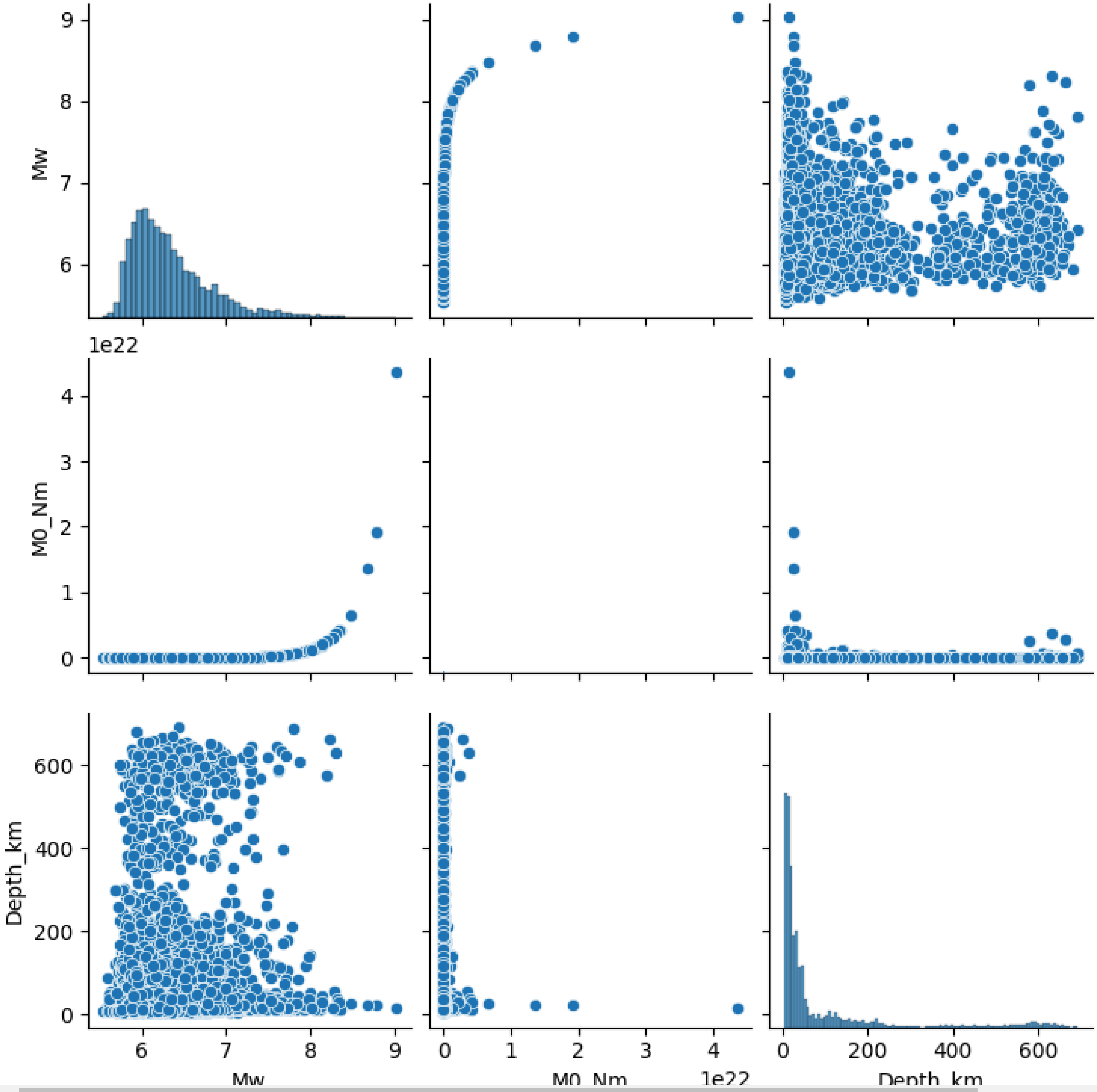
```
len(df)
```

```
4255
```

```
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)

sns.pairplot(df[features])
plt.show()

```

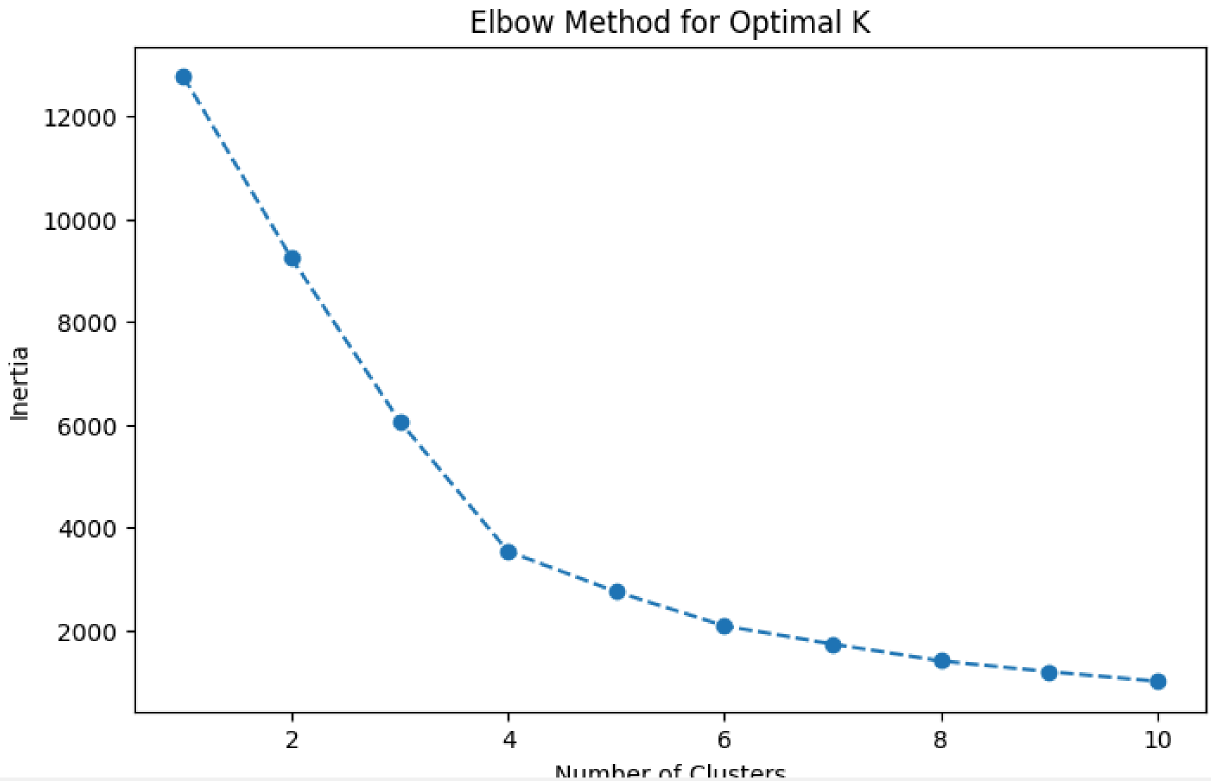


Using KMeans and Elbow Method

```python
inertia = []
K_range = range(1, 11)

for k in K_range:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(data_scaled)
    inertia.append(kmeans.inertia_)

plt.figure(figsize=(8, 5))
plt.plot(K_range, inertia, marker='o', linestyle='--')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.title('Elbow Method for Optimal K')
plt.show()
```



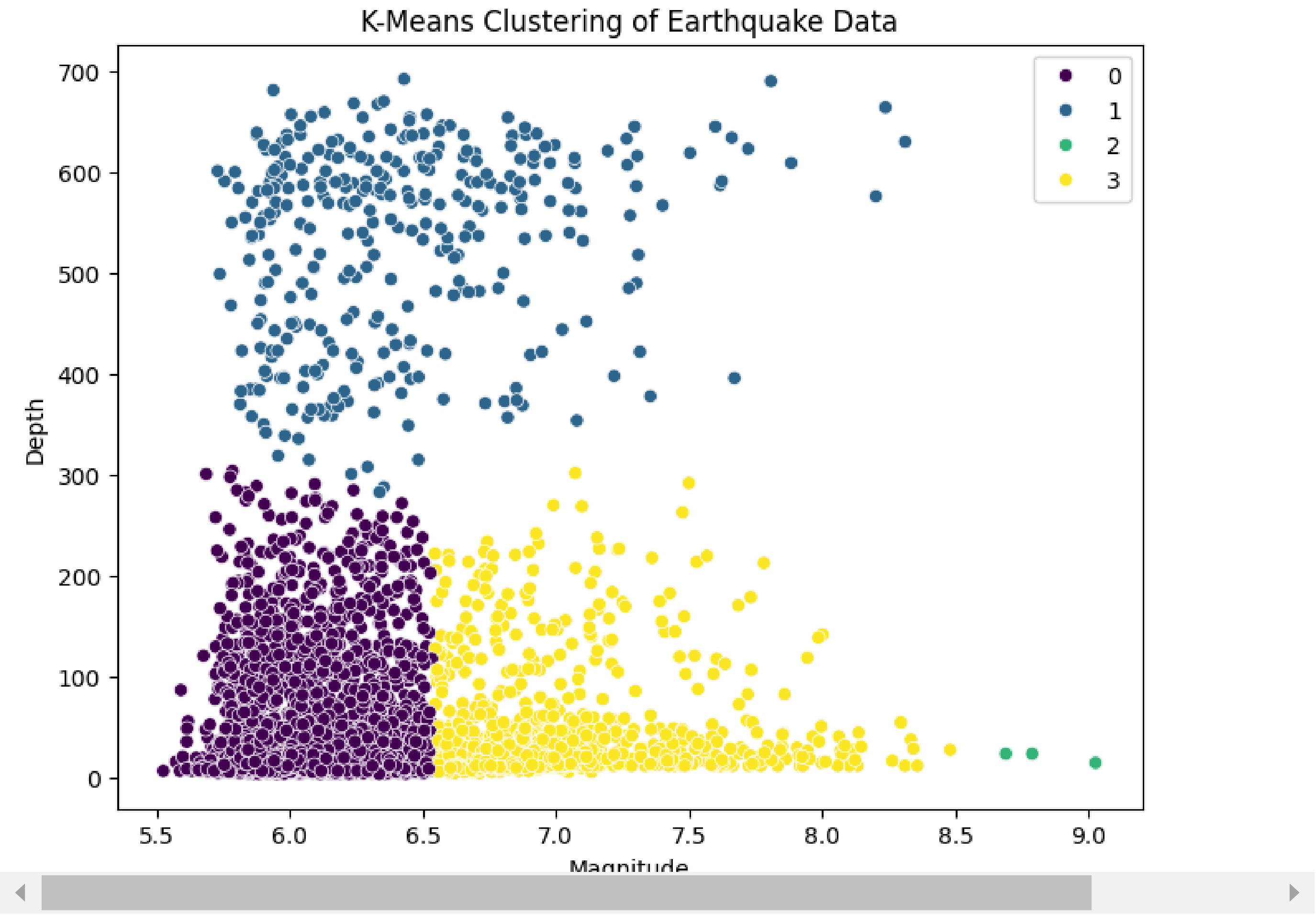
```
optimal_k = 4
kmeans = KMeans(n_clusters=optimal_k, random_state=42, n_init=10)
kmeans.fit(data_scaled)
df['Cluster'] = kmeans.labels_
```

```
silhouette_avg = silhouette_score(data_scaled, kmeans.labels_)
print(f"Silhouette Score: {silhouette_avg}")
```

```
Silhouette Score: 0.5528959012014433
```

Our score: Moderate clustering quality: Clusters are reasonably well-separated, but there could be some overlap or variation in density. Decent structure: Chosen number of clusters (K) is meaningful, but further fine-tuning (e.g., testing other K values) might improve results.

```
plt.figure(figsize=(8, 6))
sns.scatterplot(x=df['Mw'], y=df['Depth_km'], hue=df['Cluster'], palette='viridis')
plt.xlabel('Magnitude')
plt.ylabel('Depth')
plt.title('K-Means Clustering of Earthquake Data')
plt.legend()
plt.show()
```


K-Means Clustering of Earthquake Data
Depth
Magnitude
0
1
2
3
700
600
500
400
300
200
100
0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0