```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn
from sklearn import datasets
```

```
df=pd.read_csv('fitness_tracker.csv')
```

```
df.head()
```

	User_ID	Steps	Heart_Rate	Calories_Burned	BMI	Workout_Intensity	Active_Minutes
0	1	11895	131	333	22.8	High	119
1	2	10618	82	759	29.4	Moderate	106
2	3	12674	130	607	26.9	High	127
3	4	19579	84	505	25.4	High	196
4	5	5156	127	433	20.3	Moderate	52

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

```
df.isnull().sum()
```

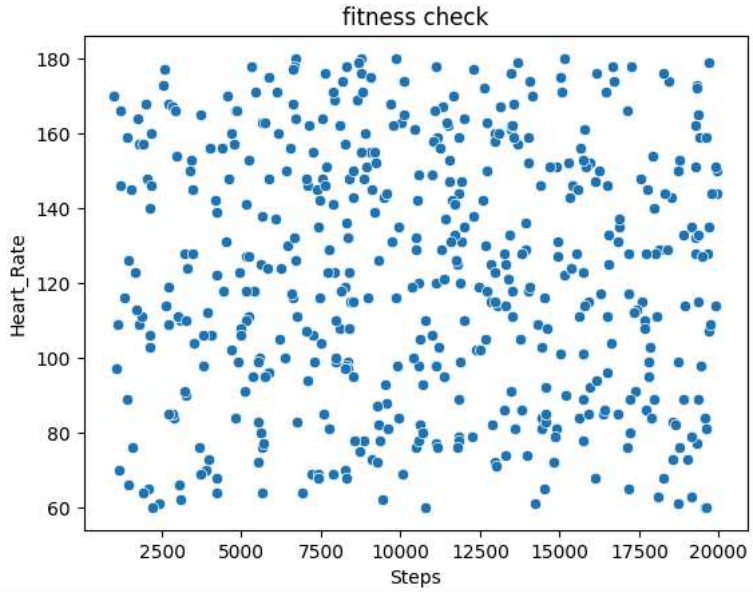
	0
User_ID	0
Steps	0
Heart_Rate	0
Calories_Burned	0
BMI	0
Workout_Intensity	0
Active_Minutes	0

dtype: int64

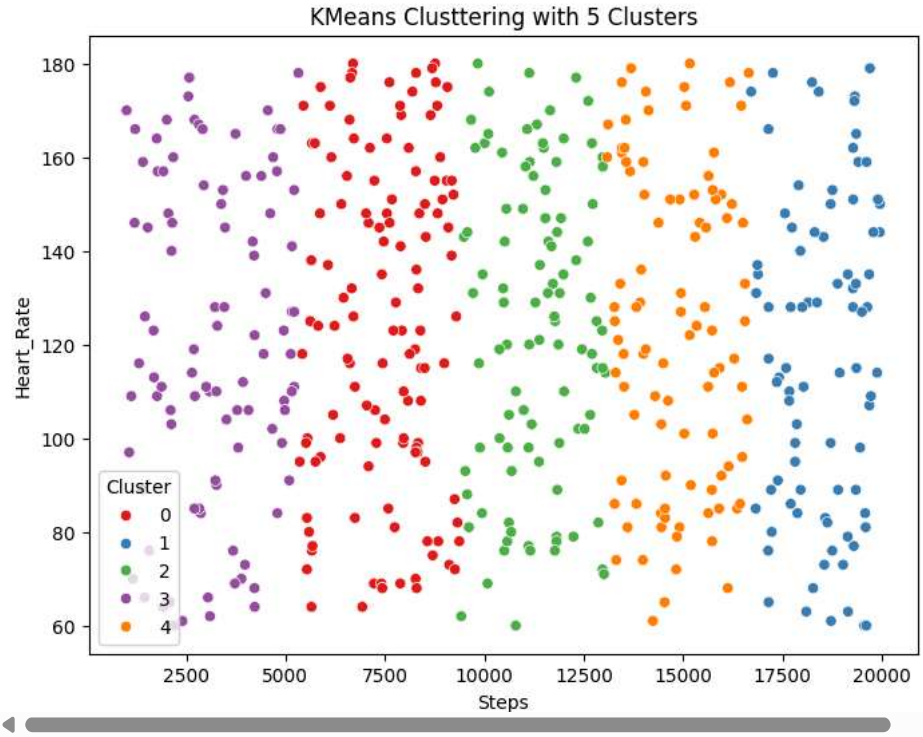
```
corr = df.corr(numeric_only=True)
```

```
from sklearn.cluster import KMeans
X = df[['Steps','Heart_Rate','Calories_Burned','BMI','Active_Minutes']]
```

```
import seaborn as sns
sns.scatterplot(data=df, x="Steps", y="Heart_Rate")
plt.title("fitness check")
plt.show()
```




```
k=5
kmeans = KMeans(n_clusters=k,random_state=60,n_init="auto")
df["Cluster"] = kmeans.fit_predict(X)

#plot clustered result
plt.figure(figsize=(8, 6))
sns.scatterplot(data=df,x="Steps", y="Heart_Rate", hue="Cluster",palette="Set1")
plt.title(f"KMeans Clusttering with {k} Clusters")
plt.show()
```



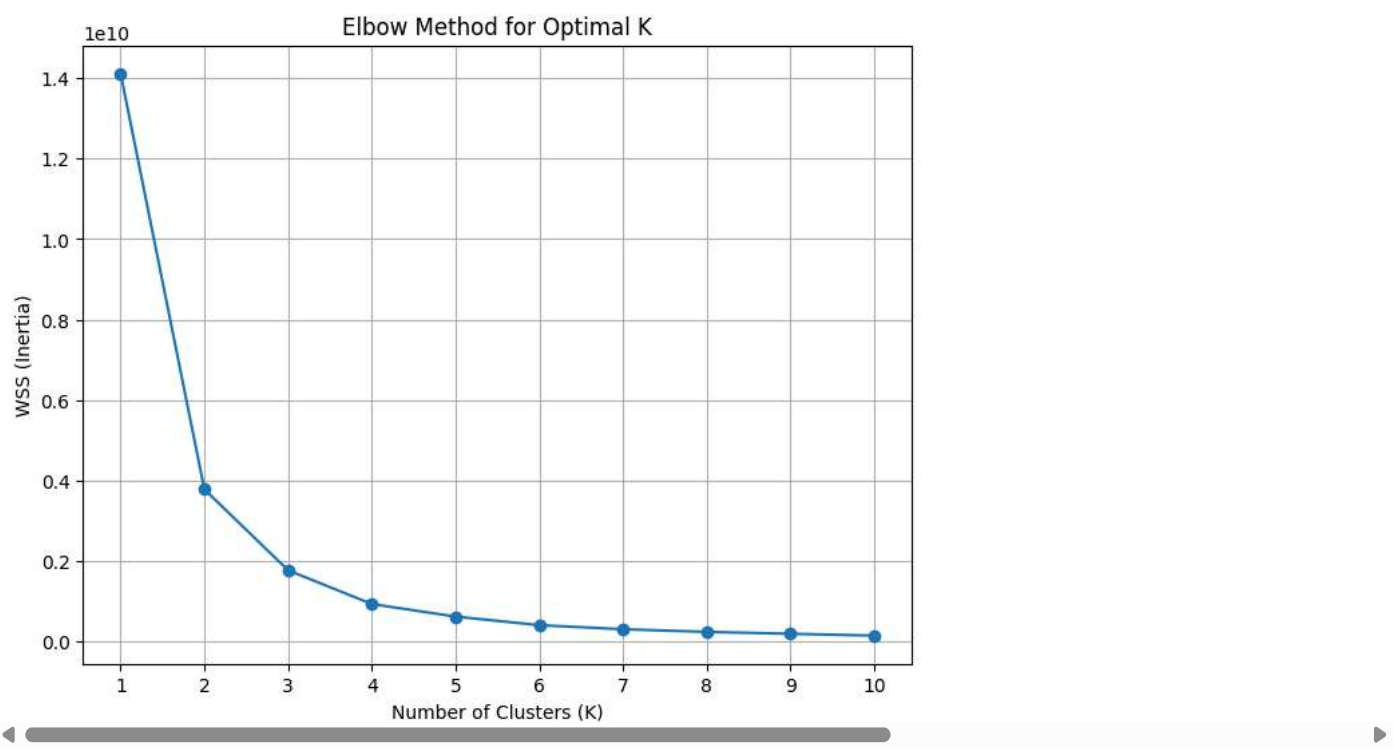

```
X = df[["Steps","Heart_Rate"]]
# feature for clustering
kmeans = KMeans(n_clusters=3,random_state=42,n_init="auto")
kmeans.fit_predict(X)
centroids = kmeans.cluster_centers_
print("Centroid for 5 clusters:\n",centroids)
```

```
Centroid for 5 clusters:
 [[17020.75         119.25641026]
 [ 4457.25477707   121.73248408]
 [10632.79144385   125.65775401]]
```

```
X = df[["Steps","Heart_Rate"]]

WSS = []  # List to store WSS for each K
k_range = range(1, 11)  # Try k from 1 to 10

for k in k_range:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init="auto")
    kmeans.fit(X)
    WSS.append(kmeans.inertia_)

#print WSS values
for k, wss in zip(k_range, WSS):
    print(f"K={k}: WSS={wss}")
```

```
K=1: WSS=14117709720.079998
K=2: WSS=3779286738.5061526
K=3: WSS=1765823707.8925357
K=4: WSS=925756034.8805773
K=5: WSS=614119262.3169428
K=6: WSS=400536644.445768
K=7: WSS=299188815.116137
K=8: WSS=233531551.05205774
K=9: WSS=185660493.5035561
K=10: WSS=141106552.80645436
```

```
plt.figure(figsize=(8, 6))
plt.plot(k_range, WSS, marker='o')
plt.title('Elbow Method for Optimal K')
plt.xlabel('Number of Clusters (K)')
plt.ylabel('WSS (Inertia)')
plt.xticks(k_range)
plt.grid(True)
plt.show()
```



From the Elnow Method the optimal value for k will be 3

```
Start coding or generate with AI.
```