```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns

df=pd.read_csv('/content/Employee.csv')
df.head()
```

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
0	Bachelors	2017	Bangalore	3	34	Male	No	0	0
1	Bachelors	2013	Pune	1	28	Female	No	3	1
2	Bachelors	2014	New Delhi	3	38	Female	No	2	0
3	Masters	2016	Bangalore	3	27	Male	No	5	1
4	Masters	2017	Pune	3	24	Male	Yes	2	1

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

```
df.tail()
```

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
4648	Bachelors	2013	Bangalore	3	26	Female	No	4	0
4649	Masters	2013	Pune	2	37	Male	No	2	1
4650	Masters	2018	New Delhi	3	27	Male	No	5	1
4651	Bachelors	2012	Bangalore	3	30	Male	Yes	2	0
4652	Bachelors	2015	Bangalore	3	33	Male	Yes	4	0

```
df.shape
```

```
(4653, 9)
```

```
df.describe()
```

	JoiningYear	PaymentTier	Age	ExperienceInCurrentDomain	LeaveOrNot
count	4653.000000	4653.000000	4653.000000	4653.000000	4653.000000
mean	2015.062970	2.698259	29.393295	2.905652	0.343864
std	1.863377	0.561435	4.826087	1.558240	0.475047
min	2012.000000	1.000000	22.000000	0.000000	0.000000
25%	2013.000000	3.000000	26.000000	2.000000	0.000000
50%	2015.000000	3.000000	28.000000	3.000000	0.000000
75%	2017.000000	3.000000	32.000000	4.000000	1.000000
max	2018.000000	3.000000	41.000000	7.000000	1.000000

```
df.replace({'Bachelors':1, 'Masters':2, 'PHD':3}, inplace=True)
df.head()
```

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
0	1	2017	Bangalore	3	34	Male	No	0	0
1	1	2013	Pune	1	28	Female	No	3	1
2	1	2014	New Delhi	3	38	Female	No	2	0
3	2	2016	Bangalore	3	27	Male	No	5	1
4	2	2017	Pune	3	24	Male	Yes	2	1

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

```
df.replace({'Bangalore':1, 'Pune':2, 'New Delhi':3}, inplace=True)
df.head()
```

```
<ipython-input-7-2c79d39905ab>:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future ver
  df.replace({'Bangalore':1, 'Pune':2, 'New Delhi':3}, inplace=True)
```

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
0	1	2017	1	3	34	Male	No	0	0
1	1	2013	2	1	28	Female	No	3	1
2	1	2014	3	3	38	Female	No	2	0
3	2	2016	1	3	27	Male	No	5	1
4	2	2017	2	3	24	Male	Yes	2	1

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

```
df.replace({'Male':1, 'Female':2}, inplace=True)
df.head()
```

```
<ipython-input-8-cedbb1bc21a8>:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future ver
  df.replace({'Male':1, 'Female':2}, inplace=True)
```

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
0	1	2017	1	3	34	1	No	0	0
1	1	2013	2	1	28	2	No	3	1
2	1	2014	3	3	38	2	No	2	0
3	2	2016	1	3	27	1	No	5	1
4	2	2017	2	3	24	1	Yes	2	1

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

```
df.replace({'Yes':1, 'No':2 }, inplace=True)
df.head()
```

```
<ipython-input-9-689c9e7f3934>:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future ver
  df.replace({'Yes':1, 'No':2 }, inplace=True)
```

	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
0	1	2017	1	3	34	1	2	0	0
1	1	2013	2	1	28	2	2	3	1
2	1	2014	3	3	38	2	2	2	0
3	2	2016	1	3	27	1	2	5	1
4	2	2017	2	3	24	1	1	2	1

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

```
df.to_csv("/content/Employee-new.csv",index=False)
new_df=pd.read_csv("/content/Employee-new.csv")
new_df.head()
```

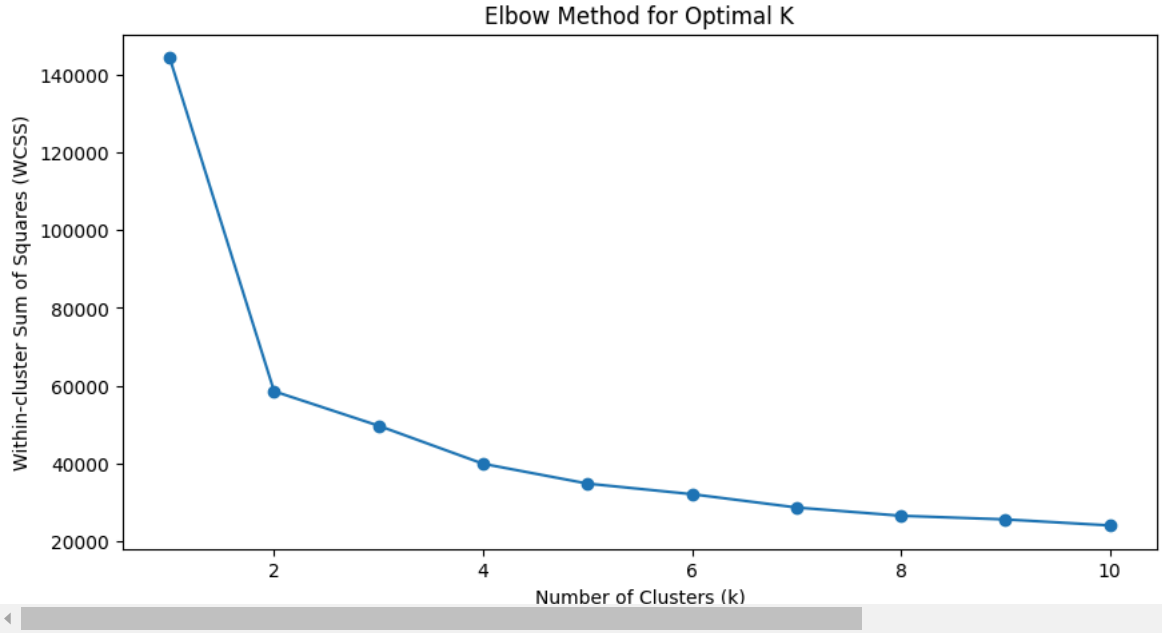
	Education	JoiningYear	City	PaymentTier	Age	Gender	EverBenched	ExperienceInCurrentDomain	LeaveOrNot
0	1	2017	1	3	34	1	2	0	0
1	1	2013	2	1	28	2	2	3	1
2	1	2014	3	3	38	2	2	2	0
3	2	2016	1	3	27	1	2	5	1
4	2	2017	2	3	24	1	1	2	1

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

Step 1: Choosing the Right Number of Clusters Using the Elbow Method

```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Sample DataFrame (replace `df` with your actual DataFrame)
```

```
df=pd.read_csv('/content/Employee-new.csv')

# Elbow Method function
def elbow_method(df, max_clusters=10):
    wcss = []  # List to store within-cluster sum of squares
    for i in range(1, max_clusters + 1):
        kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
        kmeans.fit(df)
        wcss.append(kmeans.inertia_)  # Append WCSS for each k

    # Plotting the Elbow Curve
    plt.figure(figsize=(10, 5))
    plt.plot(range(1, max_clusters + 1), wcss, marker='o')
    plt.title('Elbow Method for Optimal K')
    plt.xlabel('Number of Clusters (k)')
    plt.ylabel('Within-cluster Sum of Squares (WCSS)')
    plt.show()

# Call the elbow method function to find the optimal number of clusters
elbow_method(df)
```



Step 2: Apply K-means Clustering

```
from sklearn.cluster import KMeans

# Define the optimal number of clusters based on the Elbow Method plot
optimal_clusters = 4  # Replace this with the number you choose from the elbow plot
kmeans = KMeans(n_clusters=optimal_clusters, init='k-means++', random_state=42)
kmeans_labels = kmeans.fit_predict(df)

# kmeans_labels contains the cluster labels for each data point
```

Step 3: Apply K-medoids Clustering

```
!pip install scikit-learn-extra
```

```
Collecting scikit-learn-extra
  Downloading scikit_learn_extra-0.3.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.6 kB)
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn-extra) (1.26.4)
Requirement already satisfied: scipy>=0.19.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn-extra) (1.13.1)
Requirement already satisfied: scikit-learn>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn-extra) (1.5.2)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.23.0->scikit-learn-ext
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.23.0->scikit-le
Downloading scikit_learn_extra-0.3.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 21.0 MB/s eta 0:00:00
Installing collected packages: scikit-learn-extra
Successfully installed scikit-learn-extra-0.3.0
```

```
from sklearn_extra.cluster import KMedoids

# Apply K-medoids Clustering with the chosen number of clusters
kmedoids = KMedoids(n_clusters=optimal_clusters, random_state=42)
kmedoids_labels = kmedoids.fit_predict(df)

# kmedoids_labels contains the cluster labels for each data point
```

```
/usr/local/lib/python3.10/dist-packages/sklearn_extra/cluster/_k_medoids.py:329: UserWarning: Cluster 1 is empty! self.labels_[self
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn_extra/cluster/_k_medoids.py:329: UserWarning: Cluster 2 is empty! self.labels_[self
  warnings.warn(
/usr/local/lib/python3.10/dist-packages/sklearn_extra/cluster/_k_medoids.py:329: UserWarning: Cluster 3 is empty! self.labels_[self
  warnings.warn(
```

Step 4: Visualizing the Clusters for Both K-means and K-medoids

```
import matplotlib.pyplot as plt

# Plotting K-means Clusters
plt.figure(figsize=(12, 5))
plt.subplot(1, 2, 1)
plt.scatter(df.iloc[:, 0], df.iloc[:, 1], c=kmeans_labels, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='red', marker='X', label='Centroids')
plt.title('K-means Clustering')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()

# Plotting K-medoids Clusters
plt.subplot(1, 2, 2)
plt.scatter(df.iloc[:, 0], df.iloc[:, 1], c=kmedoids_labels, cmap='viridis')
plt.scatter(df.iloc[kmedoids.medoid_indices_, 0], df.iloc[kmedoids.medoid_indices_, 1], s=300, c='blue', marker='X', label='Medoids')
plt.title('K-medoids Clustering')
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.legend()

plt.show()
```