K_Medoids_Wajahat

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

```
df = pd.read_csv("/content/KC house complete data.csv")
df.head()
```

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqf
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650	1.0	0	0	...	7	
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242	2.0	0	0	...	7	
2	5631500400	20150225T000000	180000.0	2	1.00	770	10000	1.0	0	0	...	6	
3	2487200875	20141209T000000	604000.0	4	3.00	1960	5000	1.0	0	0	...	7	
4	1954400510	20150218T000000	510000.0	3	2.00	1680	8080	1.0	0	0	...	8	

5 rows × 21 columns

```
df.describe()
```

	id	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	c
count	2.161300e+04	2.161300e+04	21613.000000	21613.000000	21613.000000	2.161300e+04	21613.000000	21613.000000	21613.000000	21
mean	4.580302e+09	5.401822e+05	3.370842	2.114757	2079.899736	1.510697e+04	1.494309	0.007542	0.234303	
std	2.876566e+09	3.673622e+05	0.930062	0.770163	918.440897	4.142051e+04	0.539989	0.086517	0.766318	
min	1.000102e+06	7.500000e+04	0.000000	0.000000	290.000000	5.200000e+02	1.000000	0.000000	0.000000	
25%	2.123049e+09	3.219500e+05	3.000000	1.750000	1427.000000	5.040000e+03	1.000000	0.000000	0.000000	
50%	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e+03	1.500000	0.000000	0.000000	
75%	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000	1.068800e+04	2.000000	0.000000	0.000000	
max	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	3.500000	1.000000	4.000000	

```
df.drop(15870, axis = 0, inplace = True)
# reset index, because a row is droped.
df.reset_index(drop=True, inplace = True)
df.shape
```

```
(21612, 21)
```

```
df[df.columns[df.isnull().sum()>0]].isnull().sum()
```

```
Series([], dtype: float64)
```

```
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 (2.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 MB 27.8 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.10/dist-packages (from scikit-learn-extra) (1.25.
Requirement already satisfied: scipy>=0.19.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn-extra) (1.11.
Requirement already satisfied: scikit-learn>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn-extra)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.23.0->scik
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.23.
Installing collected packages: scikit-learn-extra
Successfully installed scikit-learn-extra-0.3.0
```

```
df.drop(['date', 'id'], axis = 1, inplace = True)
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
Clus_dataSet = scaler.fit_transform(df)
Clus_dataSet
```

```
array([[-0.8663876 , -0.40692359, -1.44745951, ..., -0.30611525,
        -0.94339773, -0.26072358],
       [-0.00592751, -0.40692359,  0.17558163, ..., -0.74637458,
        -0.43272969, -0.18787744],
       [-0.98044416, -1.50829275, -1.44745951, ..., -0.13569228,
         1.07009338, -0.17238527],
       ...,
       [-0.37586002, -1.50829275, -1.77206774, ..., -0.60435544,
        -1.41029422, -0.39414664],
       [-0.38157918, -0.40692359,  0.50018986, ...,  1.02886466,
        -0.84126412, -0.42051628],
       [-0.5857377 , -1.50829275, -1.77206774, ..., -0.60435544,
        -1.41029422, -0.41795257]])
```

```
from sklearn_extra.cluster import KMedoids
kmedoids = KMedoids(n_clusters=3).fit(Clus_dataSet)
```

```
df.insert(0, 'kmedoids Cluster Labels', kmedoids.labels_)
df.head()
```

	kmedoids Cluster Labels	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sq
0	1	221900.0	3	1.00	1180	5650	1.0	0	0	3	7	1180	
1	0	538000.0	3	2.25	2570	7242	2.0	0	0	3	7	2170	
2	2	180000.0	2	1.00	770	10000	1.0	0	0	3	6	770	
3	0	604000.0	4	3.00	1960	5000	1.0	0	0	5	7	1050	
4	2	510000.0	3	2.00	1680	8080	1.0	0	0	3	8	1680	

Next steps: View recommended plots

```
X = df.loc[:, df.columns != 'kmedoids Cluster Labels']
X.head()
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_baseme
0	221900.0	3	1.00	1180	5650	1.0	0	0	3	7	1180	
1	538000.0	3	2.25	2570	7242	2.0	0	0	3	7	2170	4
2	180000.0	2	1.00	770	10000	1.0	0	0	3	6	770	
3	604000.0	4	3.00	1960	5000	1.0	0	0	5	7	1050	9
4	510000.0	3	2.00	1680	8080	1.0	0	0	3	8	1680	

Next steps: View recommended plots

```
from sklearn import preprocessing
X= preprocessing.StandardScaler().fit(X).transform(X)
X[0:5]
```

```
array([[-0.8663876 , -0.40692359, -1.44745951, -0.97984121, -0.22832648,
        -0.91546593, -0.08717466, -0.3057672 , -0.62914619, -0.55885272,
        -0.73474634, -0.65864212, -0.5449314 , -0.21013346,  1.87013949,
        -0.35252787, -0.30611525, -0.94339773, -0.26072358],
       [-0.00592751, -0.40692359,  0.17558163,  0.53360192, -0.18989137,
         0.93645991, -0.08717466, -0.3057672 , -0.62914619, -0.55885272,
         0.46079706,  0.2451683 , -0.68111108,  4.74656291,  0.87957332,
         1.16160686, -0.74637458, -0.43272969, -0.18787744],
       [-0.98044416, -1.50829275, -1.44745951, -1.42625249, -0.12330593,
        -0.91546593, -0.08717466, -0.3057672 , -0.62914619, -1.40959054,
        -1.22987038, -0.65864212, -1.29391966, -0.21013346, -0.93334967,
         1.28357482, -0.13569228,  1.07009338, -0.17238527],
       [ 0.17373198,  0.69444556,  1.14940631, -0.13057096, -0.2440192 ,
        -0.91546593, -0.08717466, -0.3057672 ,  2.44468843, -0.55885272,
        -0.89173689,  1.39752658, -0.20448219, -0.21013346,  1.08516253,
        -0.28324429, -1.2718454 , -0.9142167 , -0.2845295 ],
       [-0.08214669, -0.40692359, -0.1490266 , -0.4354372 , -0.16965983,
        -0.91546593, -0.08717466, -0.3057672 , -0.62914619,  0.29188511,
        -0.13093654, -0.65864212,  0.54450607, -0.21013346, -0.07361299,
         0.40959143,  1.19928763, -0.27223402, -0.19285837]])
```

```
y = df["kmedoids Cluster Labels"]
y.head()
```

```
0    1
1    0
2    2
3    0
4    2
Name: kmedoids Cluster Labels, dtype: int64
```

```
pip install pywaffle
```

```
Collecting pywaffle
  Downloading pywaffle-1.1.0-py2.py3-none-any.whl (30 kB)
Collecting fontawesomefree (from pywaffle)
  Downloading fontawesomefree-6.5.1-py3-none-any.whl (25.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 25.6/25.6 MB 13.9 MB/s eta 0:00:00
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from pywaffle) (3.7.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (0.12
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle)
Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (1.25.
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (2
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (9.4
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffle) (
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib->pywaffl
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotli
Installing collected packages: fontawesomefree, pywaffle
Successfully installed fontawesomefree-6.5.1 pywaffle-1.1.0
```

```
Count =  df.groupby(["kmedoids Cluster Labels"], as_index=False).count()[["kmedoids Cluster Labels", "price"]]
Count.columns = ["kmedoids Cluster Labels", "Count"]
Count
```

	kmedoids Cluster Labels	Count
0	0	6776
1	1	7964
2	2	6872

Next steps: View recommended plots

```
from pywaffle import Waffle
fig = plt.figure(
    FigureClass=Waffle,
    figsize=(12, 8),
    rows=5,
    values=list(Count.Count/150),
    colors=("magenta", "yellow", "cyan"),
    legend={'loc': 'upper left', 'bbox_to_anchor': (1, 1)},
    icons='sticky-note', icon_size=18,
    icon_legend=True,
    title={'label': 'Number of Houses in each K-medoids Cluster', 'loc': 'center'},
    labels=list(Count['kmedoids Cluster Labels']))
```

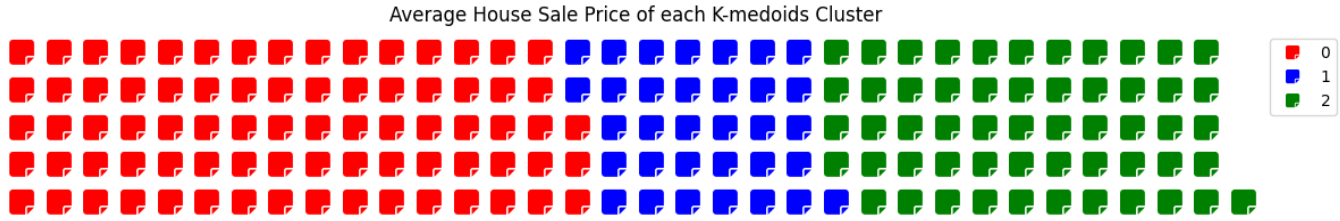
Average House Sale price of each cluster

```
labels =  df.groupby(["kmedoids Cluster Labels"], as_index=False).mean()[["kmedoids Cluster Labels", "price"]]
labels
```

	kmedoids Cluster Labels	price
0	0	783642.483471
1	1	327354.355977
2	2	546755.739086

Next steps: View recommended plots

```
from pywaffle import Waffle
fig = plt.figure(
    FigureClass=Waffle,
    figsize=(12, 8),
    rows=5,
    values=list(labels.price/10000),
    colors=("red", "blue", "green"),
    legend={'loc': 'upper left', 'bbox_to_anchor': (1, 1)},
    icons='sticky-note', icon_size=18,
    icon_legend=True,
    title={'label': 'Average House Sale Price of each K-medoids Cluster', 'loc': 'center'},
    labels=list(labels['kmedoids Cluster Labels']))
```



Double-click (or enter) to edit